Conception de learning games en réalité mixte

Cyril Benazeth

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE RHONE-ALPES

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : INFORMATIQUE

OPTION : SYSTEMES D'INFORMATION

par

Cyril BENAZETH

Conception de learning games en réalité mixte

Soutenu le 17 Janvier 2013

JURY

PRESIDENT :	**Monsieur Christophe PICOULEAU**	***Enseignant-chercheur CNAM Paris***
MEMBRES :	**Monsieur Bertrand DAVID**	***Enseignant-chercheur École Centrale de Lyon***
	Monsieur Claude GENIER	***Enseignant-chercheur CNAM Lyon***
	Monsieur René CHALON	***Enseignant-chercheur École Centrale de Lyon***
	Monsieur Florent DELOMIER	***Doctorant-chercheur École centrale de Lyon***

Remerciements

En préambule à ce mémoire, je souhaitais adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de ce stage de fin d'études.

Je tiens à remercier sincèrement Le Professeur Bertrand DAVID, Professeur à l'École Centrale de Lyon, qui, en tant que Directeur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi que pour l'inspiration, l'aide et le temps qu'il a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Mes remerciements s'adressent également à Monsieur René CHALON, Maître de conférences à l'École Centrale de Lyon pour sa générosité et la grande patience dont il a su faire preuve malgré ses charges académiques et professionnelles.

Je tiens à remercier Monsieur Florent DELOMIER, Doctorant-chercheur au laboratoire LIRIS, pour le partage de ses connaissances, sa confiance et son enthousiasme, cruciaux dans la réalisation de ce travail.

Je remercie également l'équipe SEGAREM du laboratoire LIRIS à l'INSA ainsi que l'équipe de la société SYMETRIX pour leur accueil et leur précieuse aide tout au long du projet.

J'exprime ma gratitude à tous les chercheurs doctorants du laboratoire LIRIS à l'ECL et à l'INSA qui m'ont chaleureusement accueilli et ont contribué de près ou de loin à ce travail.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, et en particulier à ma famille, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Table des matières

Introduction

Le jeu est une activité protéiforme dont aucune définition ne fait actuellement autorité : toute activité humaine peut faire l'objet d'un jeu, et réciproquement tout jeu peut cesser de le devenir. Le jeu est une institution proposant un espace de liberté au sein d'une légalité particulière définie par la règle du jeu. Il est également un moyen de représenter le monde, transposant ainsi dans un objet concret des systèmes de valeurs ou des systèmes formels abstraits, tout en étant une manière de substituer à l'ordre confus de la réalité des règles précises et arbitraires. Selon Roger Caillois, le jeu est donc une activité distincte des activités utiles, et est en ce sens « condamné à ne rien fonder ni produire, car il est dans son essence d'annuler ses résultats ». Pourtant, en tant que simulation de la vie réelle sans les conséquences tragiques de l'échec, il est aussi une stratégie d'apprentissage de la vie, donc qui produit de la connaissance.

Nous sommes aujourd'hui face à une tendance qui vise à introduire des mécanismes de jeu dans d'autres domaines : cette technique appelée ludification, ou gamification, s'appuie sur la prédisposition naturelle au jeu pour obtenir des personnes des comportements face à des situations que l'on pourrait considérer sans intérêt ou sans enjeu ou bien encore des actions que l'on ne souhaiterait ordinairement pas faire telles que remplir un questionnaire, acheter un produit, regarder une publicité ou enfin assimiler des informations. Dans ce dernier cas le jeu contribue aussi à l'acquisition de connaissances et de compétences : on parle alors de jeu pédagogique (ou Learning game, dans la mouvance Serious Game), où la motivation de gagner et la collaboration permettent entre autres de renforcer l'attention des apprenants. « Garde-toi de donner par force aux enfants l'aliment des études, mais que ce soit en le mêlant à leurs jeux, afin d'être encore plus capable d'apercevoir quelles sont les inclinaisons de chacun. » : ainsi Platon en théoricien de la connaissance et d'un point de vue de maître évoquait déjà la possibilité - voire la nécessité ? - d'utiliser le jeu dans une activité sérieuse comme l'apprentissage.

SErious GAme en REalité Mixte : SEGAREM est un projet de recherche commencé en 2010 qui vise à apporter des réponses sur le plan des méthodes, modèles et outils pour la production de dispositifs interactifs de type Serious Games utilisant des techniques d'interaction permises en réalité mixte. Les réflexions et outils définis au cours du projet SEGAREM permettront une meilleure compréhension des apports de la réalité mixte aux Serious Games. Ce projet est destiné à ouvrir la voie à de nombreux nouveaux usages dans l'apprentissage humain en environnement informatique. Dans le cadre de ce projet, le présent travail propose de concevoir un environnement comprenant une architecture et des outils permettant de créer et d'exécuter tout jeu collaboratif en réalité mixte destiné à l'apprentissage.

Après un premier chapitre exposant la structure du projet SEGAREM, nous en présentons les concepts clés que sont la réalité mixte et le Serious Game dans un deuxième chapitre, suivi d'un troisième proposant un rapide état de l'art en matière de conception de jeu. Le chapitre suivant est consacré à la description des méthodes, modèles et outils employés lors de la conception de l'architecture ainsi que les spécifications de cette dernière. La mise en œuvre de l'architecture est ensuite développée dans le cinquième chapitre. Enfin le sixième chapitre est consacré au cas Lea(r)nIt, expérimentation conduite dans le cadre du projet SEGAREM dans le but d'en évaluer les hypothèses et lors de laquelle notre architecture a été déployée, avant de conclure sur les perspectives offertes par la présente étude.

Chapitre 1

Le projet SEGAREM

1.1 Présentation

1.1.1 Objectifs et acteurs

Nous assistons à une tendance à la ludification (ou gamification) consistant à introduire des mécanismes du jeu dans des activités initialement dépourvues d'aspects ludiques dans le but d'améliorer la motivation de la personne vis-à-vis de l'activité. Introduire le jeu permet également une simulation dans laquelle l'échec n'a pas d'impact sur la vie réelle. Alors très présent aujourd'hui dans le domaine de la communication, ce phénomène a en fait émergé du domaine militaire, en tant qu'outil de publicité et de recrutement (America's Army, [U.S. Army, 2002]), en tant qu'outil de formation (Virtual Battlespace 2, [Bohemia Interactive Studio, 2007]) et enfin de briefing, dans une tentative de remplacer l'activité de présentation basée sur Powerpoint dont la complexité des diapositives ne fait que croître [Banks, 2010].

Suivant cette mouvance, les partenaires du projet SEGAREM émettent l'hypothèse que l'apprentissage peut-être plus efficace en utilisant des mécanismes hérités de jeux et dans un environnement mixte employant alors des supports nouveaux tels que des interfaces tangibles. En utilisant les ressorts de la résolution de cas, de la simulation et du jeu vidéo, les Serious Games reposent sur la mise en situation et l'immersion des apprenants. Malheureusement, la construction de compétences opérationnelles se heurte souvent à l'artificialité ou la décontextualisation des situations d'apprentissage et à la barrière que constitue parfois l'usage de l'ordinateur traditionnel. Pour tenter de dépasser ces limites, SEGAREM s'appuie sur une double approche :

- une approche interactionnelle en expérimentant l'utilisation de techniques d'interaction issues des travaux sur la Réalité Mixte (RM) comprenant la Réalité Augmentée (RA) et les interfaces tangibles (IT), pour augmenter l'environnement (par des émetteurs/capteurs sur les objets et outils), et augmenter l'acteur (à l'aide de périphériques portés), en vue de lui créer une perception en RA (en y ajoutant des éléments non perceptibles naturellement) et une interaction augmentée (capacité d'agir sur des objets virtuels);
- une approche orientée modèles destinée à la méta-modélisation des Serious Games Mixtes pour aider à leur conception et leur production. Cette approche suppose la définition de règles d'exécution, de scénarios types, de patterns d'interaction homme-machine, etc.

Dans le cadre de ce projet dirigé par Sébastien George, Maître de conférences à l'INSA, plusieurs équipes dont deux entreprises partenaires travaillent en collaboration, je rejoins l'équipe de Florent Delomier, doctorant au laboratoire du LIRIS au sein de l'équipe SILEX (Supporting Interaction and Learning EXperience) sous la supervision de Bertrand David et de René Chalon, enseignants chercheurs à l'Ecole Centrale de Lyon.

Le projet SEGAREM est financé dans le cadre du plan de relance économique lancé par la Direction Générale de la Compétitivité, de l'Industrie et des Services (DGCIS), projet labellisé par le pôle de compétitivité IMAGINOVE.

1.1.2 Responsabilités et livrables attendus

Ma première responsabilité dans ce projet a été de formaliser et concevoir une architecture générique envisagée du point de vue de l'ingénierie logicielle, afin de pouvoir créer plus facilement et d'exécuter un plus grand nombre de jeux aux mécaniques diverses mais de nature généralement collaborative. Enfin, j'ai été chargé de concevoir l'application Lea(r)nIT, « version mixte » du jeu LEAN (chapitre 6) ainsi que d'en réaliser une partie en vue de son évaluation dans le cadre du projet SEGAREM.

Notons qu'une difficulté du projet est le gouffre séparant les différentes approches impliquées, à la croisée entre sciences cognitives et sciences de l'information, entre ergonomie des IHM (Interfaces Homme-Machine) et ingénierie des IHM. Par exemple, il n'y a pas de lexique uniformisé, certains formalismes empruntant la métaphore du théâtre pour décrire le jeu (rôle, acte, etc…) alors que d'autres utilisent les même mots pour décrire d'autres entités (acte pour décrire un geste, etc.).

Enfin le développement de jeux en réalité mixte est encore à l'état expérimental. Bien que des avancées certaines aient été faites en termes de formalisation et de modèles et que des librairies spécialisées existent déjà, de telles applications dans le but spécifique des Serious Games est un nouveau pas en avant pour le domaine pédagogique. Certaines techniques pourront être empruntées au domaine du développement de jeux vidéo dont l'industrie n'est que récemment devenue mature mais dont l'aspect générique (moteur de jeux) est un sujet très complexe encore au début de son exploration.

1.2 Partenaires

Le LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information) est né début 2003 à la suite du regroupement de plusieurs laboratoires de recherche lyonnais travaillant dans le domaine des Sciences et Techniques de l'Information et de la Communication. Il est associé au CNRS. Regroupant environ 300 personnes, dont près de 110 chercheurs et enseignants-chercheurs, le LIRIS a cinq tutelles : le CNRS, l'INSA de Lyon, l'Université Claude Bernard Lyon 1, l'Ecole Centrale de Lyon et l'Université Lumière Lyon 2, et des sites à La Doua, Ecully et Bron. Au sein du LIRIS, l'équipe SILEX (Supporting Interaction and Learning EXperience) est spécialisée dans la dynamique de la connaissance, le traçage de l'expérience, dans l'adaptabilité des interfaces homme-machine. Les travaux du SILEX sur les EIAH (Environnements Informatiques pour l'Apprentissage Humain) concernent la co-conception d'EIAH situés. Enfin, d'autres membres de l'équipe SILEX, doctorants et maîtres de conférences ainsi qu'un expert pédagogique, ont travaillé sur les aspects pédagogiques du jeu à créer, liés aux mécaniques du jeu original ainsi que sur la méthode d'évaluation des expérimentations.

La technologie de la société Total Immersion est spécialisée dans les applications exploitant des principes de réalité augmentée ainsi que le tracking d'interfaces tangibles sans marqueur, par le biais de leur logiciel D'Fusion.

Enfin, nous avons collaboré plus particulièrement avec la société Symetrix, groupe SBT, basée à Grenoble et spécialisée dans le développement de Serious Games. Suivant les principes définis au laboratoire, Symetrix a réalisé de nombreux éléments dans le processus d'implémentation du jeu, notamment la partie serveur (5.4.1) ainsi que les éléments graphiques de l'application. Symetrix a également fourni des éléments matériels innovants tels que la table Microsoft PixelSense, sans compter leur grande expertise en gamification et dans le monde du jeu en général, qui nous fut précieuse et grâce à laquelle nous avons pu améliorer la ludicité du dispositif.

1.3 Participation à SysCo

Les 28-29-30 septembre 2012, nous avons participé à la conférence Sysco à Sousse en Tunisie, première conférence francophone sur les systèmes collaboratifs. Le papier rédigé à cette occasion présente entre autres l'état des recherches relatives à la conception de l'architecture et a été sélectionné pour une publication dans la revue « Ingénierie des Systèmes d'Information » éditée par Hermès – Lavoisier.

Chapitre 2

Concepts clés du projet SEGAREM

Ce chapitre a pour vocation de définir les concepts clés en jeu dans le projet SEGAREM, tels que le « Serious Game », la réalité mixte et les interfaces tangibles. Il se base sur les différents travaux effectués dans le cadre du projet.

2.1 De l'apprentissage co-situé

Du point de vue théorique, l'apprentissage collaboratif est justifié par les théories socioconstructivistes [Vygotsky, 1978] plaçant l'individu au sein d'un réseau social et dans un environnement pour construire ses connaissances afin de résoudre un problème en particulier. Les deux principales méthodes pour renforcer l'apprentissage en groupe se basent sur l'apprentissage coopératif et collaboratif (leur différence porte sur la distribution ou non des tâches). Ainsi, la réalisation d'activités collaboratives synchronisées et co-spatialisées permet la co-construction et le maintien de connaissance/compétences [Roshell et al., 1994]. Dans la lignée de la pensée vygotskienne, des études ont montré une meilleure performance des apprenants en situations collectives et médiatisées [Johnson, 1996]. Les apports sont entre autres associés à une motivation [Smith, 1981], un bien-être et une satisfaction accrus des apprenants [Jehng, 1997]. Afin de permettre cet apprentissage, nous étudions la conception et la réalisation de Learning Games collaboratifs comme outil pédagogique.

2.1.1 Learning Games Collaboratifs

Un *Learning game* est un environnement informatisé utilisant des ressorts ludiques pour favoriser l'apprentissage. Ce type d'outil pédagogique fait partie de la mouvance des *Serious Games*. L'objectif principal est d'inclure du contenu pédagogique dans des mécanismes de jeu et de créer ainsi un contexte ludique d'apprentissage. Avant d'être informatisés, ces outils pédagogiques étaient utilisés entre autres sous l'appellation « jeu d'entreprise » pour l'apprentissage de situations où la démarche analytique et individuelle ne donne pas les résultats escomptés. L'objectif est de faire émerger une situation proche de la réalité à partir de situations collaboratives en utilisant le rôle des acteurs et un processus temporel. Le comportement collaboratif, coopératif ou compétitif des acteurs et la dynamicité de l'environnement sont directement intégrés dans les mécanismes de jeu.

2.1.2 Learning Game Collaboratif Contextualisé (LGCC)

Nous définissons un LGCC comme un outil pédagogique mettant en œuvre des ressorts de jeu pour créer une expérience collective d'apprentissage prenant en compte les caractéristiques de l'environnement d'utilisation dans le but améliorer l'acquisition ou la formalisation de connaissances ou compétences. La contextualisation du LG se base sur des besoins de co-construction de connaissances/compétences et des besoins de configuration environnementale. De plus, [Sommer, 1969] précise que certaines configurations spatiales des utilisateurs d'un système sont plus intéressantes que d'autres dans la réalisation d'une activité. Enfin, l'impact de

l'utilisation simultanée de représentations physiques et numériques des entités présentes sur l'interface a déjà été évalué pour améliorer l'apprentissage: [Marshall, 2007], [Liu, W. et al., 2007] [Cook et al., 2008]. Chacun d'eux a montré différents avantages de ces représentations pour améliorer la compréhension et la mémorisation.

2.2 Les serious games

Bien qu'il n'y ait pas de définition universellement reconnue des serious games, la majorité des acteurs impliqués s'accordent pour dire que ce sont des jeux ayant un but qui dépasse le simple amusement [Susi et al., 2007]. Le terme de serious game regroupe de nombreux types de jeux qui se distinguent par leurs objectifs. On y trouve :

- Les advergames, qui sont des jeux publicitaires à buts de faire connaître une marque ou un produit
- Les learning games, qui sont des jeux à buts d'apprentissage de contenus pédagogiques
- Les serious games qui sont des jeux à buts de sensibilisation et/ou d'information

Cette liste ne présente que les catégories de jeux les plus représentées. Les types de Serious Games évoluent constamment avec la création de nouveaux jeux, ce qui fait évoluer les classifications actuellement existantes.

L'utilisation des Serious Games en entreprise est de plus en plus courante. Ce sont des jeux qui visent l'apprentissage de connaissances ou compétences professionnelles. Ils peuvent être des jeux de rôle ou des jeux de gestion, qui ne sont pas forcément des jeux vidéo. Il s'ensuit qu'en fonction des définitions de Serious Games, les jeux d'entreprise peuvent être rangés dans les Serious Games ou en être exclus. Dans le cadre du projet SEGAREM, puisque les Serious Games auxquels nous nous intéressons ont essentiellement pour objet l'apprentissage de connaissances et de compétences, nous nous limitons à la notion de learning game. Le learning game se définit dans ce contexte comme un « environnement informatisé utilisant des ressorts ludiques pour favoriser des apprentissages ».

2.2.1 Exemples de Serious Games

Un des premiers Serious Games à se faire connaître par le grand public est America's Army. Il a été créé pour l'armée des Etats-Unis afin d'inciter les joueurs à s'enrôler et est disponible gratuitement sur internet depuis le 4 juillet 2002. L'application, basée sur le moteur du jeu Unreal Tournament, propose de simuler des exercices d'entraînement militaire et des missions de combat. Depuis, les Serious Games se sont diversifiés. Ils sont aujourd'hui particulièrement présents dans les domaines de la défense, de l'enseignement et de la formation, de la publicité, de l'information et de la communication, de la santé, de la culture et dans le secteur militant [Michaud, 2008]. Dans le domaine de la formation et de l'éducation, nous pouvons citer Simples Machines, disponible gratuitement sur le site du muséum des sciences et de l'industrie de Chicago, qui met en scène un robot qui doit collecter des éléments. Pour cela, le joueur doit construire des machines simples jouant avec les lois de la physique. Ces lois sont ensuite rapidement expliquées au joueur [Simple Machines, 2009]. Dans le même domaine et pourtant très différent, nous pouvons aussi citer Conspiracy code, qui est un jeu conçu pour les étudiants de l'Université de Floride. Dans ce jeu d'espionnage, ils doivent construire et utiliser leurs connaissances en histoire américaine pour arrêter un complot, le tout en communiquant via une interface web [Florida Virtual School, 2009].

Depuis une douzaine d'années, le LIRIS, en collaboration avec le département Génie Industriel de l'INSA de Lyon, a conçu et développé des jeux pédagogiques pour enrichir la formation initiale et continue des Ingénieurs, Masters et Mastères Spécialisés. Ces Serious Games ont tous pour objectif l'apprentissage de compétences métier : Puissance 7, REACTIK, KITSMED, NEGOCOM, CIRES, TRANSFER, FACTORY 21, PROHA, FLAM, BIVOUAC, CHALOG, PERFINDUS, MANVAL, CASSIOPEE, 6-SIGMA [LIESP, 2010]. Dans la grande majorité des cas, les apprenants sont par équipe de 3 et sont accompagnés par un tuteur, qui les encadre, replace en perspective les acquis du jeu et peut éventuellement en modifier le déroulement. Par exemple, à travers le jeu puissance 7, les élèves doivent assimiler des méthodologies de résolution de problème et savoir utiliser les outils de la qualité. Pour cela, ils se retrouvent par équipes de 3 face à une entreprise de distribution de produits électroménagers ayant des problèmes de livraison. Il faut enquêter pour identifier le problème et imaginer des solutions pour le résoudre. Ce jeu est utilisé dans différents départements d'enseignement de l'INSA de Lyon (génie industriel, informatique, télécom), où il est intégré à la formation des ingénieurs. Les apprenants sont guidés dans le jeu par le tuteur, qui anime aussi plusieurs phases de débriefing au cours des douze heures de jeu.

Chacun de ces jeux pourrait bénéficier des apports d'un environnement mixte. L'objectif du présent travail est de proposer un environnement de création et d'exécution facilitant la production de leurs jeux, afin de fournir un meilleur apprentissage.

2.3 Réalité Mixte

Le terme de Réalité Mixte regroupe schématiquement deux composantes : la Réalité Augmentée, où l'on part du réel et où le virtuel vient ponctuellement enrichir le réel, et la Virtualité Augmentée, où, à l'opposé, on part du virtuel et on enrichit celui-ci par le réel.

2.3.1 Réalité augmentée

Le terme Réalité Augmentée (RA) a été défini pour la première fois au début des années 1990 [Caudell & Mizell, 1992] afin de désigner des systèmes interactifs caractérisés par la présentation en temps réel d'entités virtuelles. Elle a ensuite été définie par [Mackay, 1993] comme une manière de réintégrer « l'information électronique dans le monde physique». Il s'agit de «permettre aux gens de tirer parti de leurs compétences dans l'interaction avec le monde de tous les jours, tout en profitant de la puissance des réseaux informatiques ». La réalité augmentée implique une relation sémantique et contextualisée entre les scènes réelles et les entités virtuelles [Fuchs, Moreau & Papin, 2001]. La notion d'augmentation fait référence à l'enrichissement des informations perçues par l'utilisateur. Nous pouvons caractériser la réalité augmentée selon 3 propriétés :

- Incorporation perceptible par l'utilisateur d'entités virtuelles dans un espace réel
- Augmentation de la réalité réalisée en temps réel
- Cohérence de la disposition des objets réels et virtuels

3 cibles de l'augmentation sont possibles :

- L'augmentation de l'utilisateur par l'utilisation de dispositifs (sur la tête ou les mains) permettant l'acquisition de nouvelles informations sur les objets physiques.
- L'augmentation des objets physiques, de manière à ce que des objets soit modifiés en leur ajoutant un périphérique d'entrée et/ou de sortie ou alors des capacités informatiques sur ou en lui.

- L'augmentation de l'environnement, où des périphériques indépendants fournissent et collectent de l'information de/sur l'environnement.

2.3.2 Réalité Mixte

La « réalité mixte » désigne le continuum et l'hybridation qui relient les mondes physique et numérique, les mondes « réel » et « virtuel ». En pratique, cette hybridation se traduit au niveau de la perception (quel que soit le ou les sens utilisés pour cette perception) par un mélange entre les informations données par l'environnement physique de l'utilisateur et les informations calculées et produites par un système informatique. De manière synthétique, le concept de réalité mixte peut être utilisé pour désigner une combinaison d'environnements virtuels et réels [Milgram & Kishino, 1994].

Figure 1. Le continuum de la réalité mixte selon [Milgram, 1994]

Dans le cas de la réalité augmentée, l'objet du domaine (ou objets de la tâche) est réel, alors que dans le cas de la virtualité augmentée, l'objet du domaine est virtuel. Les différents outils utilisés n'ont aucune influence sur le positionnement du dispositif sur le continuum.

La réalité mixte désigne donc des systèmes interactifs associant objets réels et données informatiques de manière cohérente [Chalon 04]. Deux aspects sont étudiés : l'augmentation des acteurs (par des dispositifs portés) et l'augmentation de l'environnement (par des capteurs/émetteurs sur les objets ou les outils). La perception et la manipulation d'objet réel sont augmentées par l'association d'information issue du monde numérique (ajout d'éléments non perceptibles naturellement).

2.3.3 Virtualité augmentée

D'un côté du continuum, la réalité augmentée utilise les avantages des deux mondes pour réaliser des actions dans le monde réel. De l'autre côté du continuum est présent un environnement virtuel augmenté par des objets réels pour faciliter la perception et la manipulation d'objets virtuels. Ainsi, pour améliorer la manière dont l'homme interagit avec son environnement informatique de production, une tendance vise à introduire des objets physiques dans les interfaces utilisateurs conventionnels pour permettre une interaction plus intuitive ou efficiente. [Fitzmaurice and Buxton, 1997, Ishii and Ullmer, 1997].

2.3.4 Interfaces tangibles

La notion d'interface tangible (TUI : Tangible User Interface) a été introduite par [Ishii et Ullmer, 1997]. Leur approche se base sur l'utilisation des "Graspable User Interface" de Fitzmaurice, Ishii et Buxton. La proposition est d'utiliser des objets physiques pour interagir avec l'ordinateur et sortir des paradigmes d'interfaces graphiques habituels (appelés WIMP Windows / Icon / Mouse / Pointer - ou GUI - Graphical User interface). Voici les 4 propriétés clés des interfaces tangibles proposés par [Ullmer et Ishii, 2000] traduites par [Couture, 2010] :

Propriété 1 : Couplage informatique des représentations tangibles à l'information numérique sous-jacente : la principale caractéristique des TUIs est que les représentations tangibles sont reliées à l'information numérique et aux modèles informatiques sous-jacents.

Propriété 2 : Incarnation par les représentations tangibles des mécanismes pour le contrôle interactif : les représentations tangibles servent aussi de contrôle physique de l'information numérique. Les interacteurs tangibles peuvent être inertes, se déplaçant seulement lorsqu'ils sont manipulés directement par les mains de l'utilisateur. Les interacteurs tangibles peuvent aussi être animés par le système informatique (moteurs, retour de force, force magnétique).

Propriété 3 : Couplage des représentations tangibles, de manière perceptuelle, aux représentations intangibles dynamiques : les TUIs reposent sur un équilibre entre représentations tangibles et intangibles. Bien que l'incarnation tangible des éléments joue un rôle central, et principal dans la représentation et le contrôle d'une TUI, les représentations intangibles jouent aussi un rôle au sein des TUIs. Une représentation intangible au sein d'une TUI médiatise souvent la majeure partie de l'information dynamique fournie par le système informatique sous-jacent. Avoir une réaction immédiate, de la représentation intangible, correspondant à la manipulation de la représentation tangible, est alors un point critique pour assurer la perception du couplage. La coïncidence des espaces d'action et de perception (continuité spatiale des représentations tangibles et intangibles) est aussi une condition essentielle afin d'améliorer la perception du couplage.

Propriété 4 : L'état physique des artéfacts de la TUI incarne partiellement l'état numérique du système. L'état de la représentation physique doit représenter et correspondre à l'état du modèle numérique. Si l'état du modèle numérique change, l'état de la représentation physique doit changer aussi.

L'interface tangible est donc une alternative aux interfaces graphiques et textuelles, permettant de coupler des objets physiques à des données informatiques. Aussi, l'affordance [Norman 90] (capacité d'un objet à suggérer sa propre utilisation) de l'objet physique peut être associée à celle de l'objet virtuel. Le parti pris par le projet SEGAREM est d'utiliser les interfaces tangibles qui offrent une nouvelle approche de l'interaction homme-machine et entrent dans le cadre de la réalité mixte. Elles proposent que l'utilisateur manipule directement les données au travers d'objets réels présents dans son environnement.

2.4 Exemples de jeux en réalité mixte

2.4.1 ARQuake

ARQuake [Piersky et Thomas, 2002] est un jeu basé sur le jeu vidéo Quake [Id Software, 1996], devenu open source et modifié pour intégrer de nouvelles interactions. Il se déroule à la première personne, en extérieur avec un équipement mobile adapté : HMD (Head-Mounted-Display), arme, ordinateur. Le joueur perçoit des monstres superposés au monde réel par le biais de son HMD et tente de les abattre à l'aide de son pistolet.

Figure 2. Un joueur d'ARQuake

2.4.2 Incretable

IncreTable [Leitner, 2008] est une table de jeu en réalité mixte inspirée de "L'incroyable machine" (jeu qui consiste à créer une machine en utilisant des éléments divers: des engrenages, des fusées, des souris en cage, des chats, des bascules, des dynamites, des boules de bowling, des ciseaux...). Ce système mélange un grand nombre de technologies dans le but de permettre une action du réel sur le virtuel mais aussi du virtuel sur le réel. Pour cela, ils exploitent des capteurs à base de « Time of Flight » camera (qui utilise la vitesse de la lumière pour déterminer les distances et les profondeurs), de tracking et de marqueurs. Mais l'ajout majeur est la création d'un objet réel appelé Portal servant d'actionneur bidirectionnel entre le réel et le virtuel. Ici il s'agit d'un petit cylindre pouvant recevoir un signal optique et permettant de déclencher une action mécanique (faisant tomber les dominos réels). A l'inverse ce cylindre peut aussi détecter qu'un domino réel vient de tomber sous la tour (grâce à une rupture de faisceau optique) et d'envoyer un signal lumineux à une caméra placée sous la table, déclenchant alors la chute des dominos virtuels.

Figure 3. IncreTable

2.5 Exemples de Learning Games en Réalité Mixte

Les Serious Games mixtes (dans ce cas destinés à l'apprentissage) sont à ce jour encore peu nombreux. Nous pouvons citer le jeu « AR Kanji » [D. Wagner et I. Barakonyi, 2003], qui utilise des PDA pour enseigner la signification de kanji (symboles japonais). Les joueurs étendent sur la table 10 cartes imprimées des deux côtés. Le PDA énonce un mot puis demande au joueur de retourner la carte associée au vocabulaire demandé. Lorsque le joueur retourne la carte, le PDA (filmant la table et les cartes via une caméra) superpose un affichage en 3D de l'objet associé au mot inscrit sur la carte. Si l'image correspond au mot demandé, le joueur continue. Sinon, c'est au tour du 2e joueur.

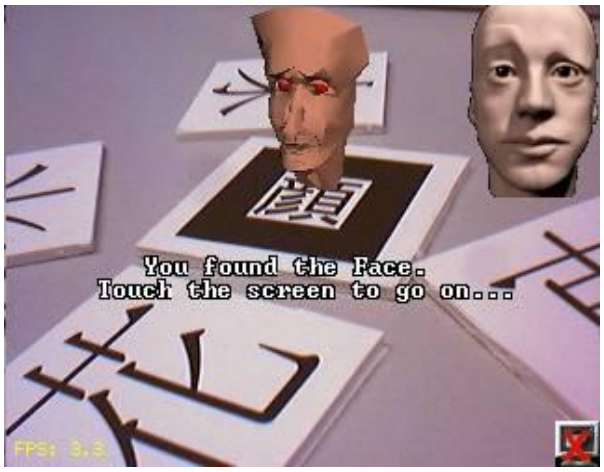


Figure 4. AR Kanji

Plusieurs Serious Games mixtes utilisent des dispositifs mobiles dans un contexte réel : « Zoo scene investigators » [Perry et al., 2008], « Reliving the revolution » [Schrier, K., 2006], et « Environmental detectives » [Squire et Jenkins., 2003]. Ces jeux sont basés sur une enquête à résoudre : *Environmental detectives* par exemple est un jeu basé sur un PDA avec GPS. Les joueurs doivent, en se déplaçant à travers le campus, préparer un rapport avec tous les problèmes rencontrés (sur l'environnement). Les joueurs doivent trier les informations, les organiser, et les interpréter.

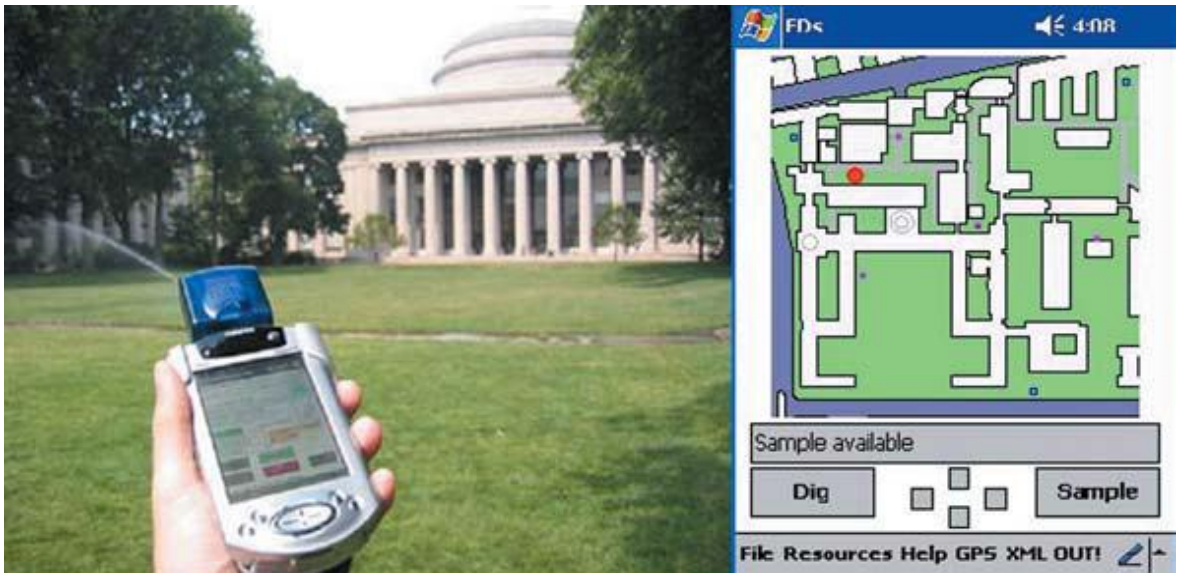


Figure 5. Environmental détectives - A gauche, un joueur calibre son GPS. A droite, la position du joueur est indiquée sur la carte du MIT

D'autres environnements utilisant des interfaces tangibles et destinée à l'apprentissage pour des enfants sont à la limite du jeu. « System blocks » [Zuckerman et al., 2005] est basé sur la manipulation de blocs, tandis que Liu et al. [Liu et al., 2007] privilégient la manipulation d'une tasse pour comprendre le système solaire ou le cycle de vie d'une plante.

Ces quelques rapides exemples illustrent la diversité pouvant émerger des jeux utilisant les interactions permises en réalité mixtes. Voyons plus en détail ce qui fait un jeu et les moyens et méthodes existants pour les créer.

Chapitre 3

De la conception de jeux sérieux mixtes

La conception d'un jeu (le « Game Design ») est un sujet si vaste qu'il est impossible d'en exposer tous les aspects ici. [Djaouti 2011] a traité dans sa thèse l'essentiel des aspects relatifs au Game Design de Serious Games et nous en reprenons ici quelques éléments qui nous semble importants d'être rappelés. S'agissant de concevoir un moteur capable d'exécuter un jeu collaboratif dans un contexte de réalité mixte, nous avons commencé par explorer les différentes solutions issues du domaine vidéo-ludique, l'industrie quoique relativement jeune offrant déjà de nombreuses possibilités. Loin de se vouloir exhaustif en la matière, ce chapitre propose un tour d'ensemble de nos recherches préliminaires.

3.1 Définir un jeu

De nombreuses définitions existent, mais aucune ne semble faire de consensus. [Juul 2003], reprenant les travaux de [Zimmerman & Salen 2003], considère qu'une bonne définition du jeu doit décrire 3 aspects :

- Le jeu en tant qu'objet créé par un concepteur *(« the game »)*.
- La relation entre le joueur et le jeu *(« the player »)*.
- Le rapport du jeu au reste du monde *(« the world »)*.

Les travaux de Juul le conduisent à proposer 6 points cruciaux devant définir un jeu au regard desquels les 3 aspects précédemment cités doivent être mis (figure 6):

- Les règles
- Le résultat quantifiable variable
- La valorisation du résultat
- L'effort du joueur
- L'attachement du joueur au résultat
- Les conséquences négociables

	« The Game » *Le jeu en tant qu'objet créé par un concepteur*	« The Player » *La relation entre le joueur et le jeu*	« The World » *Le rapport du jeu au reste du monde*
1. Règles	X		
2. Résultat quantifiable variable	X		
3. Valorisation du résultat		X	
4. Effort du joueur	X	X	
5. Attachement du joueur au résultat		X	
6. Conséquences négociables			X

Figure 6. Tableau définitoire du modèle classique du jeu [Juul, 2005]

Enfin, il propose la définition du « jeu » suivante: *« Un jeu est un système formel basé sur des règles donnant lieu à un résultat quantifiable variable. Ces variations de résultat sont associées à des valeurs différentes. Le joueur est donc émotionnellement attaché au résultat du jeu, et s'implique de manière à influencer le résultat produit. Cependant, les conséquences réelles d'une telle activité restent facultatives et négociables. »*

Dans le cas du Learning Game, les conséquences négociables en question sont de nature pédagogique : au-delà du but du jeu interne (l'état vers lequel le joueur doit tendre pour « gagner »), le vrai but est que le joueur ait pu construire ou maintenir ses connaissances. Enfin, notons que Juul considère qu'un même jeu peut utiliser différents supports tant que des derniers permettent à la fois une puissance de calcul suffisante pour appliquer les règles du jeu en réponse aux actions du joueur, ainsi qu'une capacité à mémoriser l'état actuel du jeu et de ses composants.

3.2 Conception de jeu

Le terme « game design » peut se traduire par « conception de jeu », tout comme un « game designer » est donc un concepteur de jeu. Cette traduction renvoie à la définition donnée par [Salen & Zimmerman, 2003]. Les deux chercheurs définissent le Game Design comme *« Le processus par lequel un concepteur crée un jeu, destiné à être utilisé par un joueur, afin que naisse une expérience de jeu »*.

Cette définition reste vague quant à la nature exacte de ce « processus » de game design. Djaouti semble penser que cela est dû à l'apparente diversité des pratiques se rapportant à la création de jeux et que s'il n'existe pas de méthode absolue pour créer un jeu, il existe une grande diversité d'approches dans la création d'un jeu. Il distingue trois « cultures » en particulier, selon le milieu dans lequel évolue le game designer, le niveau de formation théorique et technique et les outils utilisés:

- Les game designers professionnels venant de l'industrie vidéo-ludique, produisant de nombreux titres qui seront commercialisés dans les rayons « jeux vidéo » des magasins. Disposant de moyens toujours plus importants et issus de formations spécialisées, ils peuvent utiliser des kits de développements spécifiques et commercialisés (3.4.2) pour être plus productifs.

- Les game designers professionnels indépendants, évoluant généralement dans des équipes plus petites (1 à 5 personnes). Ils partagent parfois le niveau de formation des game designers venant de l'industrie, mais ont des moyens plus modestes et sont souvent impliqués dans de plus nombreuses phases de création d'un jeu (conception, développement, graphismes, etc.). Bien qu'ils utilisent des outils (moins complexes, et moins chers), les game designers indépendants expérimentent souvent des pratiques qui seront ensuite récupérées par l'industrie, comme par exemple le fait de fournir à la communauté des joueurs les outils permettant de modifier le jeu selon leurs goûts.
- Les créateurs amateurs, caractérisés non pas par leur cadre de travail mais plutôt par les outils qu'ils utilisent, tels que les outils de « modding » (modification) permettant de modifier ou d'étendre un jeu existant, ou bien les « usines à jeux » facilitant le travail technique requis pour créer un jeu.

Ces trois cultures ont chacune contribué à ce qu'est devenu aujourd'hui le game design en créant des méthodes et outils dont nous avons utilisé certains dans le cadre du projet.

3.3 Gameplay

Lors de la conception d'un jeu, plusieurs éléments doivent être pris en compte afin d'en définir le « Gameplay », anglicisme dont le français n'a pas d'équivalent mais qui peut être défini comme « caractérisant les éléments d'une expérience ludique, c'est-à-dire le ressenti du joueur quand il utilise le jeu. Un 'bon Gameplay' tiendra le joueur en éveil alors qu'un 'mauvais Gameplay' le frustrera ou l'ennuiera. » [Wikipédia]. A cette définition du Gameplay nous pouvons ajouter le fait que la notion de logique de jeu (les règles et mécaniques) y est également présente, ou du moins contribue au ressenti du jeu. De manière générale, un bon Gameplay est la caractéristique déterminante d'un jeu classique dans lequel le joueur vient y chercher du divertissement brut, mais dans le cas du Learning Game il n'est qu'un des éléments permettant de motiver l'apprenant, l'apprentissage devant rester au cœur du processus.

Les éléments typiquement constitutifs du Gameplay sont dépendants des genres de jeux, la classification des genres eux-mêmes souffrant d'un manque de taxonomie globale. Différentes taxonomies sont proposées mais aucune ne recouvre tous les aspects des Gameplays présents dans la culture ludique. L'approche de [Mariais et al. 2011] établit une classification des types de jeux par ressorts ludiques et par mécaniques de jeu.

3.3.1 Ressorts ludiques

Selon Mariais, les ressorts ludiques sont les éléments pouvant amener une motivation pour le joueur. Mariais a distingué les ressorts suivants, certains pouvant être combinés:

- La compétition : le sentiment de confrontation ou d'affrontement avec d'autres joueurs
- La transfiguration : le sentiment de devoir se mettre à la place d'un personnage réel ou fictif
- L'aléa : le sentiment de devoir s'adapter à une situation non anticipée ni anticipable
- La reconnaissance : le sentiment d'être valorisé par et dans ses propres actions
- La collaboration : le sentiment de partager et échanger des ressources avec d'autres joueurs-apprenants dans un objectif commun
- Le défi : le sentiment de devoir dépasser ses propres capacités

3.3.2 Mécaniques de jeu

Pour chacun de ces ressorts, [François-Gilles Ricard, *in* Mariais C., Dupin C., 2010] a identifié les mécaniques de jeu suivantes permettant de les mettre en œuvre:

- L'évolution (du personnage, de l'environnement avec passage de niveaux, récompenses, etc…)
- Le principe d'élimination d'adversaires
- L'alignement, le positionnement
- Le hasard
- Le blocage, l'obstruction
- Le bluff
- La déduction
- La diplomatie
- L'échange, le négoce
- Les enchères
- Le principe de majorité
- La programmation/planification
- Le parcours (ex. Monopoly, jeu de l'oie)
- La course
- La mémoire
- La vitesse, les réflexes
- L'agilité, l'adresse
- Les questions-réponses – devinettes (Pictionary, Trivial Pursuit, etc.)
- Les jeux de lettres, les jeux de chiffres
- L'optimisation, la maîtrise des règles
- La reproduction, l'imitation (Tangram, puzzle…)

Ces mécaniques de jeu et les ressorts qu'ils permettent de mettre en œuvre font l'objet d'une étude par Symetrix qui sera complétée et finalisée dans le cadre du projet SEGAREM et qui s'attachera notamment à définir et spécifier les liens entre les besoins pédagogiques d'une part, et d'autre part les ressorts de jeu, les mécaniques de jeu et les interactions de jeu qui permettent de les mettre en œuvre. Les mécaniques de jeu décident généralement du cadre des règles de jeu qui seront structurées. Ces mécaniques sont également une propriété attractive du jeu, en ce qu'elles définissent un cadre objectif et non ambigu que les joueurs vont pouvoir exploiter pour progresser dans le jeu. Un moteur dit générique devrait dans l'idéal supporter l'ensemble des mécaniques et ressorts, et rendre possible des combinaisons, par exemple « planification et optimisation ».

3.3.3 L'univers fictionnel du jeu

La plupart des jeux ont une forme d'univers fictionnel supportant l'action, existant sous forme de graphismes, de sons, de textes, de règles de jeux. Ce monde fictionnel, ou « background » est aussi une propriété attractive forte d'un jeu [Malone, 1980], pour des raisons opposées à celles motivées par les mécaniques : l'univers fictionnel y est au contraire subjectif, ambigu et prête à de nombreuses interprétations et discussions.

3.3.4 La temporalité dans le jeu

Un jeu peut être vu comme une simulation dans un univers fictionnel plus ou moins abstrait d'une situation confrontant les joueurs à un (ou plusieurs) but(s). L'univers fictionnel a sa propre temporalité, distincte de celle appartenant au temps de jeu réel. Le temps a donc une place

importante dans le Gameplay, selon le type de jeu. On distingue les types de temporalités suivants:

- Tour par tour : chacun attend que le joueur précédent ait fini ses actions avant de pouvoir jouer
- Parallèle sans contrainte : chacun peut jouer indépendamment des actions des autres joueurs
- Parallèle contraint par les données : tout le monde joue en même temps, mais les actions possibles d'un joueur dépendent du résultat des autres joueurs
- Parallèle contraint par le temps : ici, nous sommes dans une simulation ou les résultats sont consolidés à des jalons de temps défini. Lorsque le grain est très fin, la simulation peut y être vue comme en « temps réel ».

Le choix d'une temporalité fait partie intégrante de la conception d'un jeu au plus bas niveau (5.2.1) en y imposant son propre rythme de manière plus ou moins perceptible.

3.4 Outils pour la conception de jeux pédagogiques

3.4.1 Formalisme de conception

En entrée du processus de conception d'un jeu pédagogique se trouvent les spécifications, ici présentées sous forme de scénarii. Ce processus de conception comprend une étape de créativité pour laquelle il n'existe pas de méthode universelle. Il n'existe pas non plus de telle méthode permettant d'écrire des scénarii incluant des éléments de jeu, des spécifications d'ordre pédagogique, ainsi que des éléments de réalité mixte. [Orliac, 2012] propose dans ses travaux sur les jeux pédagogiques en réalité mixte (JPRM) de faciliter le processus de spécification de Learning Games, notamment à l'aide d'une extension de spécification pédagogique, d'un modèle des tâches détaillé ainsi que d'une spécification pédagogique prenant en compte la réalité mixte. Orliac a identifié 7 étapes dans la conception d'un jeu :

1. Définition du projet : le concepteur pose les bases de la motivation, des besoins et des objectifs
2. Préparation : le concepteur explore les données existantes en rapport avec son besoin initial, sélectionne le type de jeu le plus adapté.
3. Incubation : période pendant laquelle le concepteur détourne son attention du problème donné. La réflexion sur ce problème continue mais se fait inconsciemment.
4. Illumination : L'illumination est une prise de conscience des idées muries durant la phase d'incubation. Le nombre d'idées grandit durant les phases 2, 3 et 4, chaque idée pouvant amener à une autre. La figure 7 illustre ce nombre croissant d'idées selon les 3 axes composant un JPRM : la pédagogie, le jeu, et la RM. A la fin, l'ensemble des idées qui ont émergé constitue un ensemble de solutions potentielles.
5. Vérification : Chaque solution envisageable (prise dans l'ensemble de solutions précédent) est évaluée par rapport aux idées d'origine, aux contraintes et aux autres solutions déjà choisies. Chaque décision impacte les autres choix à faire, puisque il faut assurer une cohérence globale. Le nombre d'idées se restreint au fur et à mesure que le concept du JPRM s'affine. Si une solution potentielle ne convient finalement pas, une

autre idée est sélectionnée et vérifiée. Il se peut aussi qu'aucune des solutions de cet ensemble ne convienne, ce qui amène à recommencer une partie du cycle en recherchant de nouvelles informations, et en faisant émerger de nouvelles idées (conception itérative)

6. Formalisation : Le concepteur décrit ensuite les idées ou solutions retenues. L'objectif final de la formalisation est d'obtenir des spécifications rédigées de manière claire pour le développeur. Mais elle constitue aussi un moyen pour le concepteur d'exprimer une idée pour la vérifier ou l'affiner. De plus, elle permet de présenter ses idées à d'autres personnes, et d'en discuter plus facilement. En ce sens, les étapes de créativité et de formalisation ne sont pas entièrement dissociées dans un processus de conception.
7. Spécifications finales : L'affinage et la formalisation des idées aboutissent à des spécifications rédigées, qui vont servir de base au développement d'un prototype.

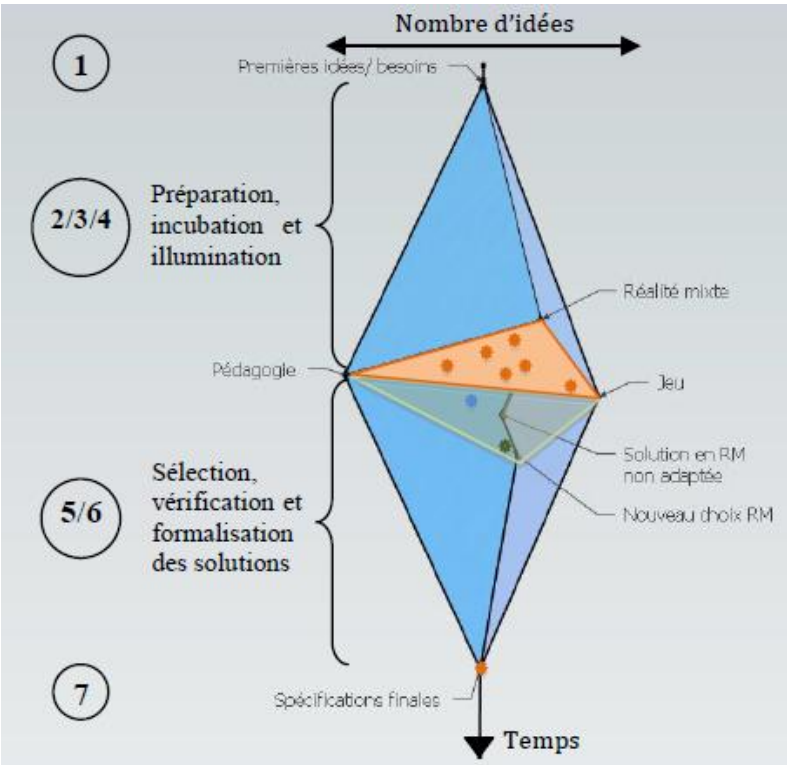


Figure 7. Illustration du processus de conception de JPRM

Inspiré de spécifications existantes et utilisant la métaphore du théâtre, le formalisme proposé F-MRLG prend en compte les éléments d'un Learning Game à plusieurs niveaux :

- Eléments globaux : audience visée, objectifs pédagogiques, mécaniques et ressorts de jeux, thème, contexte physique et périphériques
- Eléments généraux de flux de travaux (Workflow) : rôles, missions et activités
- Flux de travaux détaillé : arbre des tâches, arbres des concepts, objets interactifs et gestes

Cette étape de créativité est supportée par un outil d'aide à la créativité et de transformation de ces spécifications en langage XML, MIRLEGADEE. L'outil prend en compte les éléments exprimés dans la figure 8.

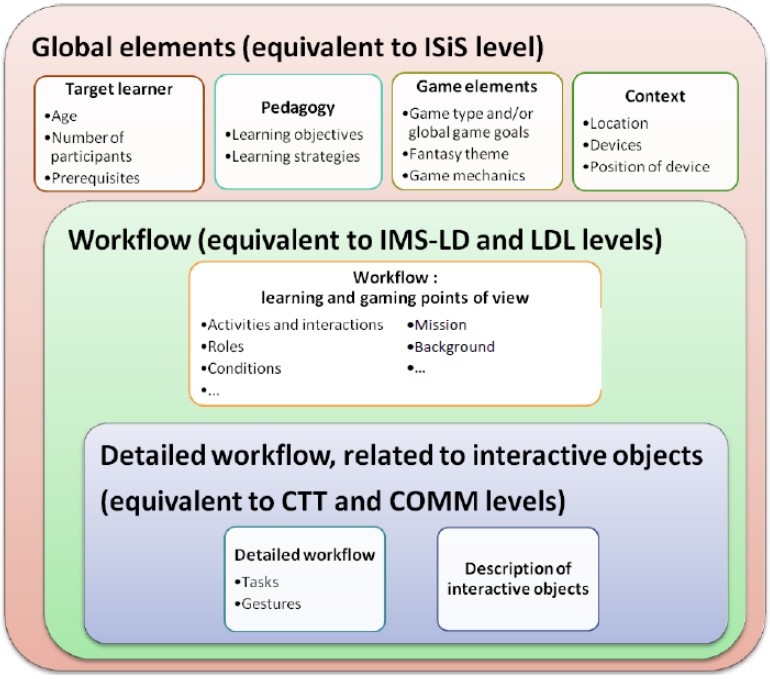


Figure 8. Schéma F-MRLG

3.4.2 Outils pour le développement de jeux

Half-Life, [Valve Corporation 1997], Quake, [Id software, 1996], StarCraft [Blizzard, 1998] et Unreal [Epic Games & Digital Extreme, 1998] sont entre autres des références en matière de jeu vidéo. Un de leurs points communs est d'être chacun basé sur un moteur ayant facilité le développement d'autres jeux de même calibre, voire ultérieurement de meilleurs. Cependant il s'agît en l'occurrence de moteurs commercialisés souvent à un prix élevé, de plus ils sont clairement orientés vers un type de jeu 'First Person Shooter', dont la vue subjective d'un environnement en « 3D » est caractéristique, et souvent associée à des mécaniques de jeu classique. Ces moteurs ont beaucoup évolué depuis, mais la tendance a longtemps été à la course à la performance et au réalisme plus que vers la généricité des mécaniques de jeux à exécuter, permettant l'affichage de plus en plus de polygones, et de plus en plus de fonctionnalités telles que la gestion des textures, des éclairages et ombrages, des animations de squelettes, de la simulation physique des éléments, etc. De plus, malgré d'évidents avantages d'immersion lorsqu'il s'agît d'un jeu où le joueur est seul face à l'écran, l'approche de la vue tridimensionnelle « aplatie sur un écran » est, au sens IHM, finalement assez éloignée des concepts prônés par la réalité mixte.

Les usines à jeux sont des outils de création de nouveaux jeux autonomes. Ils fournissent généralement divers outils tels que des éditeurs de contenu (graphismes, niveaux, avatars, etc…), de règles, permettant de créer rapidement des jeux vidéo. L'offre est pléthorique, [Djaouti, 2011] ayant répertorié 363 logiciels de ce type dans son corpus, et admettant que sa liste est loin d'être exhaustive. Néanmoins l'on distingue deux grandes catégories d'usines à jeux : les usines à jeux spécialisées dans la création d'un jeu de genre spécifique (course, plateforme, jeu de rôle, etc…) et les usines à jeux dites « généralistes ». La première catégorie est très riche et, souvent gratuite ou libre, elle est plus liée à une audience de game designers amateurs désireux de se lancer rapidement dans la création d'un jeu d'un genre donné, dont les mécaniques sont communes à

d'autres jeux du même genre. Cherchant à permettre le maximum de créativité ainsi que le niveau de généricité le plus élevé possible, nous nous intéressons plus particulièrement à la deuxième catégorie, les usines à jeux « généralistes » permettant une plus grand souplesse dans la création de jeux, dont voici quelques exemples significatifs extraits des travaux de Djaouti.

3.4.2.1 Gamemaker

Pionnier en la matière, Gamemaker [Kitchen, 1985] permet déjà, malgré des limitations (pas de scrolling, nombre total d'instructions et sprites limités), de créer une grande variété de jeux (de l'époque !), notamment grâce à son éditeur de règles sous forme de langage de programmation simple ainsi qu'un éditeur de niveaux sous forme d'interface textuelle.

3.4.2.2 La famille « Klik »

Les éditeurs issus de la société Clickteam proposent un environnement tout-en-un ainsi qu'une interface de programmation sous forme de tableur proposant toutes les règles et permettant de les éditer facilement sans forcément connaître la programmation. Ce concept a ensuite été repris de nombreuses fois, et désigné sous l'appellation de « la famille Klik ». La version la plus récente, The Games Factory 2, est parue en 2006. En 2011 le site The Daily Click répertoriait plus de 400 créations basées sur ce principe. Des alternatives gratuites ou libres existent telles que GameDevelop ou Construct.

3.4.2.3 Game Maker

Game Maker a été créé par un professeur d'informatique, Mark Overmars, dans le but d'apprendre à ses élèves la programmation informatique en s'appuyant sur la conception de jeux vidéo. Il reprend les concepts de la famille Klik mais permet d'insérer des méthodes dans les objets du jeu, là où la famille Klik permettait de n'utiliser que des propriétés. Ainsi, chaque objet manipulable par le joueur ou le concepteur illustre le paradigme de programmation orienté objet, permettant entre autres l'utilisation de concepts fondamentaux de la POO tels que l'héritage ou l'encapsulation. Le succès de Game Maker est analogue à celui de la famille Klik, le site Game Maker Games recensant en 2011 près de 4000 titres.

3.4.2.4 Scratch

Poussant le concept de programmation « sans code » encore plus loin, Scratch [MIT Media Lab 2007] permet d'utiliser des briques visuelles pour programmer le comportement des objets. L'utilisateur crée des blocs de comportements (ou en utilise à partir d'une bibliothèque) et les connecte aux objets graphiques.

3.4.2.5 Virtools

Développé par [Dassault Systems, 2009] et orienté vers le design d'objets tridimensionnels et la simulation tels que proposés par CATIA, 3DVia Virtools 5.0 peut être utilisé comme une usine à jeux et est intéressante dans son approche de simplification de la programmation, utilisant à l'instar de Scratch des briques visuelles pour introduire de la logique fonctionnelle dans les objets.

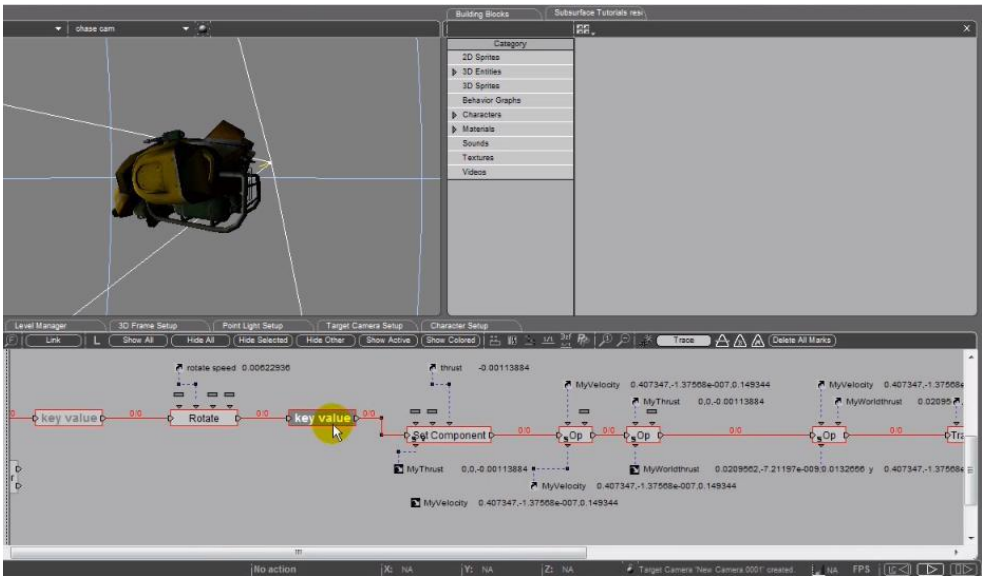


Figure 9. Virtools, éditeur de niveau (en haut) et de règles (en bas)

3.4.2.6 SGTools

Développé par Onlineformapro (2010), SGTools a pour vocation de simplifier la création de Serious Games, et a un fonctionnement similaire à celui de Virtools (éditeur de niveaux en 3D, et éditeur graphique de règles). Ciblant l'industrie du Serious Game, SGTools est produit en utilisant la technologie Flash (3.4.2.8), ce qui lui permet d'être largement diffusable par Internet, et intègre notamment des fonctionnalités permettant le suivi du joueur respectant la spécification SCORM, dans le but d'être intégré à des plateformes LMS.

3.4.2.7 Kits de développements

Les usines à jeux vues précédemment restent dans la catégoriques des usines à jeux généralistes situés dans une gamme de prix accessible aux amateurs, voire à des professionnels de la scène indépendantes. En conséquence, la technologie proposée (moteur graphique, physique, IA, etc…) n'est pas comparable à celle utilisée dans l'industrie du jeu vidéo regroupant des jeux à gros et très gros budgets et réalisés par des grandes équipes de développement. Ces jeux « AAA », équivalents vidéo-ludiques des « blockbusters hollywoodiens », ont souvent l'avantage d'une campagne de promotion importante et ont un taux de succès commercial relativement prévisibles. Les kits de développements proposent une technologie de pointe, qui nécessite une certaine maîtrise et s'adressent donc plus aux professionnels ayant une solide formation technique qu'aux amateurs, et plutôt que d'accroître l'accessibilité de leurs fonctions vont privilégier l'industrialisation des tâches répétitives du développement d'un jeu. Ainsi, bien que certains d'entre eux soient disponibles gratuitement pour toute création non-commerciale, tel que l'Unreal Development Kit, le savoir technique requis pour utiliser cette usine à jeux met cette solution hors de portée de la plupart des game designers amateurs et indépendants.

3.4.2.8 Flash : un logiciel de création professionnel accessible aux amateurs

Développé par Macromédia (racheté en 2005 par Adobe Systems), Adobe Flash est une suite de logiciels permettant la manipulation de graphiques vectoriels, de bitmaps et de scripts ActionScript, qui sont majoritairement utilisés pour les applications web, les jeux et les vidéos. Malgré un support vacillant (les lecteurs flash n'étant pas supportés sur toutes les plateformes) il jouit d'une grande popularité auprès des développeurs de jeux vidéo, amateurs ou non, ainsi qu'auprès des écoles multimédia. Adobe Flash comporte notamment un environnement de développement complet mêlant éditeur vectoriel et d'animation et utilisation d'ActionScript (ainsi que des assistants de scripts), ce qui le rend relativement simple d'usage pour un large public (développeurs, graphistes), et malgré tout très puissant : il propose un grand nombre de librairies ActionScript. Le compilateur est gratuit, et des alternatives gratuites à l'éditeur Adobe Flash existent, telles que FlashDevelop (6.5.5). Enfin Adobe fournit également une plateforme d'exécution et de déploiement : AIR (Adobe Integrated Runtime), fonctionnant sur les mêmes principes que la machine virtuelle de Java, et supporté par la plupart des plateformes visées par le projet SEGAREM (6.5).

3.5 Concevoir une architecture de jeu générique

Notre but est de concevoir un environnement capable d'exécuter une grande variété de mécaniques de jeu telles que définies par Mariais (3.3.2) tout en étant suffisamment flexible pour être étendu à d'autres fonctionnalités. De plus l'architecture doit:

- permettre une expérience de jeu à plusieurs, donc collaborative, sur des dispositifs physiques hétérogènes.
- permettre à plusieurs personnes de jouer simultanément sur un seul dispositif
- prendre en compte les styles d'interaction issus de la réalité mixte (interfaces tangibles, multi-touch, etc.)
- pouvoir mémoriser un état du jeu, le recharger
- permettre une traçabilité des tâches et des actions utilisateurs
- permettre de déployer rapidement des zones numériques
- permettre de déployer des modifications à distance des règles et d'interfaces utilisateur
- supporter les 5 types de dynamicité (3.3.4)

Les architectures de moteurs de jeux sont des systèmes, voire des systèmes de systèmes très complexes. [Gregory, 2009], dans son livre de référence « Game Engine Architecture », insiste sur le fait que la limite entre ce qui est le jeu et son moteur est souvent floue et dépend très fortement des jeux en question. Certains en font une claire distinction, proposant des cadres d'exécution dans lesquels il est par exemple possible de définir un « monstre » entièrement par les données alors que d'autres n'en font aucune, codant « en dur » le fameux monstre. A partir du moment où de tels éléments deviennent codés « en dur », ou bien utilisent du code spécifique pour certains objets de jeu ou types de jeux, il devient très difficile voire impossible de réutiliser le code pour un autre jeu. Le terme « moteur de jeu » est alors réservé pour les logiciels qui sont extensibles et peuvent être utilisés comme fondation pour de nombreux autres jeux sans modification majeure.

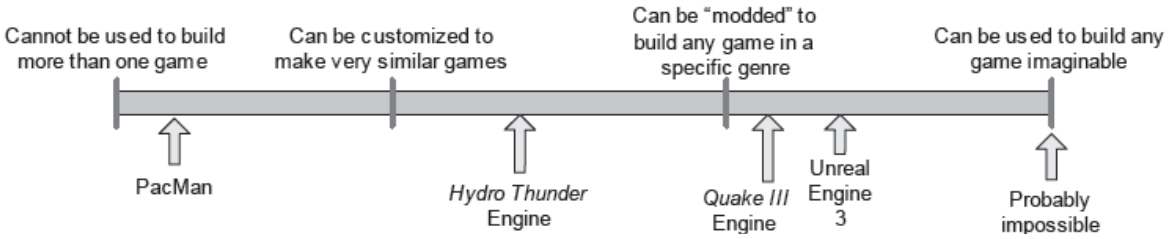


Figure 10. Gamme de réutilisabilité des moteurs de jeu selon [Gregory, 2009]

Dans la figure 10 exprimée par Jason Gregory, on voit clairement que la limite entre jeu et moteur de jeu n'est pas évidente. D'ailleurs selon lui un moteur capable d'exécuter n'importe quel jeu imaginable n'existe pas à priori, ce qui ne veut pas dire que l'on ne puisse pas tendre vers une telle généricité.

3.5.1 Une architecture centrée sur les données

L'aspect paramétrable et l'indépendance forte des composants d'un moteur générique sont essentiels. Dans cet esprit et sous l'angle de l'ingénierie des logiciels, [Plummer, 2004] a analysé deux grands jeux-vidéos, StarCraft [Blizzard, 1998] et Unreal Tournament [Epic Entertainment, 1999] et leur prérequis fonctionnels avant de proposer une architecture de jeu vidéo en tant qu'un « système de systèmes » duquel émerge le jeu vidéo lui-même. En effet, un jeu vidéo est un système complexe devant intégrer des systèmes provenant de domaines eux-mêmes complexes, et très différents les uns des autres tels que les graphismes, l'intelligence artificielle, le son, les contrôleurs, le réseau etc. Dans une telle architecture exprimée en couche il faut, comme dans tout logiciel, éviter les dépendances circulaires, ces dernières menant à un très fort couplage entre les couches, rendant alors le moteur très difficile à tester, instable et inhibe la réutilisation de code.

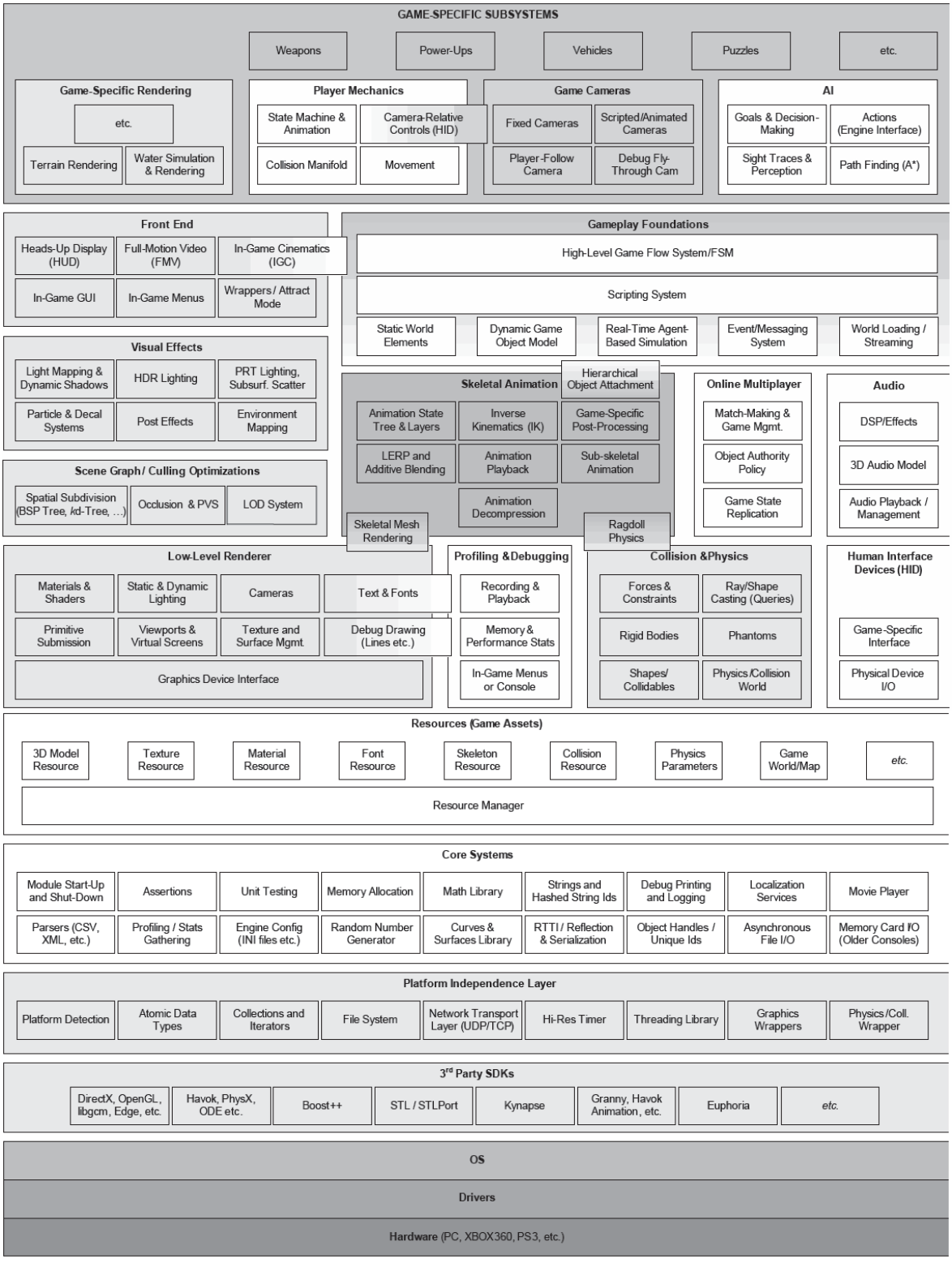


Figure 11. Une architecture d'exécution d'un moteur de jeu selon [Gregory, 2009]

Dans cette figure 11 l'on peut distinguer plusieurs macro-niveaux :

- Le premier niveau concerne les « ressources », fournissant l'exécution bas niveau ainsi que les images, sons, etc. Sa structure en couche est très clairement définie.
- Au-dessus de ce niveau on trouve d'autres composants tels que le rendu bas-niveau, l'animation, la gestion de la physique, et la fondation du Gameplay. On remarque néanmoins que la structuration en couche y est moins stricte, par exemple la gestion de l'animation des squelettes semble « déborder » sur celles du rendu bas niveau, des physiques ainsi que celle des systèmes de script.
- Enfin, tout en haut on trouve tous les sous-systèmes spécifiques au jeu tels que l'intelligence artificielle, les mécaniques, les objets etc.

Dans un souci constant de performances, nombre de moteurs ont créé des dépendances fortes entre ces différents domaines, par exemple le système de collision (« physique » du jeu) est souvent intégré au système de rendu graphique, évitant ainsi de surcharger les communications entre ces module, néanmoins [Plummer, 2004] pense que la performance n'est pas un critère de qualité essentiel dans la conception d'un moteur, les opérations les plus complexes (et « gourmandes ») devant être effectuées par les sous-systèmes spécialisés. Par exemple, la boucle graphique dessinant 10 millions de polygones d'un objet est bien plus significative que la communication demandant au système graphique de le dessiner. Il ne s'agît pas de considérer que la performance n'est pas importante, mais que dans la conception d'un « système de systèmes » elle est de priorité moindre que sa flexibilité et son extensibilité.

Plummer, constatant l'utilisation de plus en plus courante de composants « sur l'étagère » (COTS : Components On-The-Shelf) permettant de forts gains de productivité, insiste encore sur le très faible couplage qu'il doit exister entre ces composants, mais aussi entre les « composants internes ». Après avoir étudié les différentes approches architecturales, Plummer conclut que seul un modèle dirigé par les données (Data-Driven Model) peut répondre aux exigences de flexibilité, de maintenabilité et d'extensibilité d'un moteur de jeu dont la généricité doit être une des principales qualités. La philosophie d'un « système de systèmes » permet notamment de réduire les dépendances fortes qu'il peut exister entre les différents domaines.

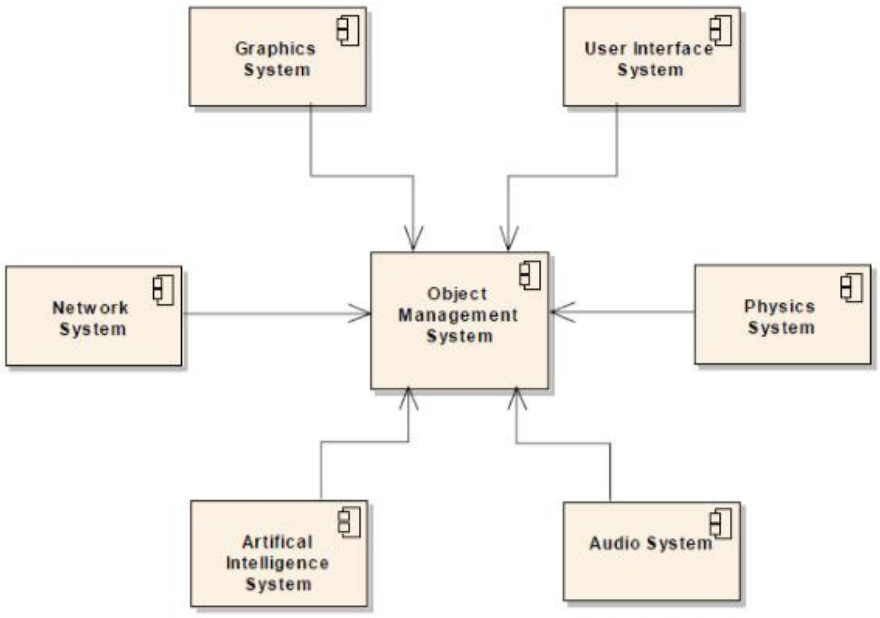


Figure 12. Architecture centrée sur les données selon [Plummer, 2004]

Alors qu'une telle architecture offre la souplesse visée ainsi que d'autres qualités telles que la réutilisation de composants, elle ne donne aucune indication quant à la façon dont le fonctionnement du jeu doit être implémenté. De plus, la communication entre les modules n'y est pas spécifiée : doit-elle être implémentée sous forme d'interfaces, basée sur des évènements? Alors que l'interface utilisateur devrait être basée sur des évènements, car les interactions utilisateurs sont par nature asynchrones, la communication entre les graphismes et la simulation physique elle devrait *à priori* être de nature synchrone.

Alors qu'il nous semble ici incomplet, le modèle centré sur les données demeure intéressant pour notre étude en ce qu'il propose une paramétrisation poussée des différents modules. Il nous faut maintenant étudier la façon dont nous voulons aborder la modélisation des mécaniques du jeu, qui vont déterminer sa nature ainsi que la façon dont les joueurs vont interagir avec le jeu.

3.5.2 Un moteur de mécaniques de jeu

La logique de jeu constituée par ses règles existe pour tout jeu mais elles sont créées différemment pour chaque moteur de jeu. Cette dépendance existe à cause de la manière dont le modèle objet interne est implémenté dans le moteur, spécifiquement à l'endroit où la logique est intrinsèquement liée aux données résidant dans les sous-systèmes de plus bas niveau. [Magnusson, 2011] propose un modèle objet tendant vers l' indépendance la plus grande possible entre ces composants et les sous-systèmes dans le but d'offrir un cadre de travail propice à toute logique de jeu : un moteur de jeu indépendant du genre, appelé alors moteur de mécaniques de jeu (Game Mechanics Engine).

Les règles du jeu sont une part essentielle de toute mécanique de jeu. Elles définissent à la fois les possibilités et les limitations du jeu, et donnent du sens à l'action du joueur ainsi qu'une structure au jeu. Les règles sont généralement définies, non ambigües et faciles à comprendre, tout en fournissant un challenge suffisamment intéressant pour que l'acte de jouer se transforme en processus d'apprentissage.

[Juul 2005] voit les règles de jeux comme finies et définies, telles des algorithmes, ayant un niveau de spécification en entrée suffisamment détaillé pour amener une machine à état fini dans un autre état. En tant que telles, elles sont alors exprimables par un langage informatique. Il constate également qu'il y a deux importantes façons de structurer des règles dans un jeu : par *émergence* et par *progression*. La structuration par *progression* mène à une structure de jeu linéaire, dans lequel le joueur progresse dans un enchaînement de défis, alors que dans la structuration par *émergence* les défis prennent place indirectement via les règles du jeu, ce dernier n'ayant alors aucune séquentialité définie. Selon Juul la plupart des jeux tombent dans une catégorie hybride quelque part entre ces deux extrêmes. Si l'on reprend les mécaniques définies par Mariais (3.3.2), au vu des combinaisons possibles on peut considérer qu'il existe une très grande variété d'ensemble de règles. Heureusement on trouve généralement des règles communes dans les jeux de genre communs, ce qui permet de faire émerger des patrons de mécaniques (« Game mechanics pattern »). Par exemple, dans les jeux de parcours tels que le jeu de l'oie ou les petits chevaux, le gagnant du jeu est le premier arrivé. Dans un jeu de style FPS (First-Person-Shooter), les joueurs ne peuvent généralement pas traverser les murs, ce qui nous donne une règle commune de collision joueur/mur, etc.

Après avoir exprimé une définition d'un jeu et les problématiques liées à sa conception, nous avons exploré quelques possibilités techniques accompagnant cette activité créative. Définissons à présent les concepts, méthodes et modèles dont nous allons avoir besoin pour créer notre architecture.

Chapitre 4

Méthodologie et modèles pour la conception d'environnement interactif

Ce chapitre expose en premier lieu la méthodologie utilisée pour la recherche de solution, les différentes approches et concepts utilisés dans la construction de l'architecture, ainsi que l'expression des besoins que cette dernière doit satisfaire.

4.1 Cadre de conception et de développement logiciel

4.1.1 Principes MDA

Nous utilisons des techniques empruntées à l'ingénierie dirigée par les modèles (IDM, ou MDA pour Model-Driven-Architecture). L'approche MDA est constituée de trois éléments : les processus, les modèles et les outils.

La puissance de l'approche MDA est de produire, à partir de spécifications et scénarii, une architecture indépendante de l'implémentation cible, que ce soit au niveau matériel ou logiciel, tout en assurant une grande réutilisation des modèles en jeu.

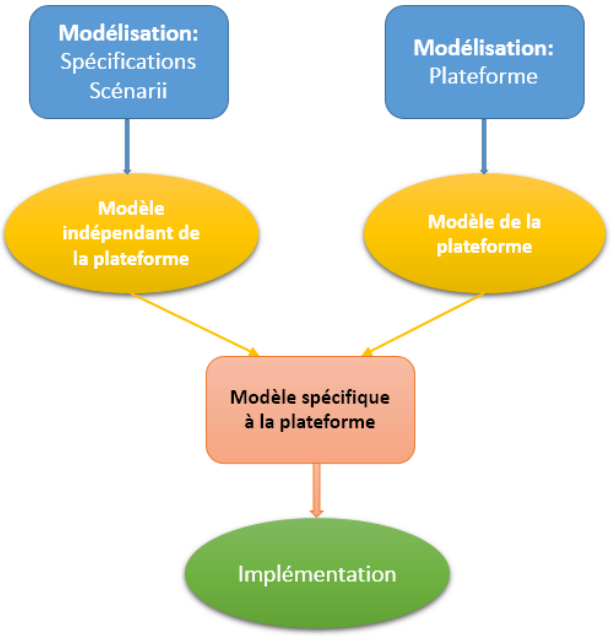


Figure 13. Principes généraux MDA exprimés dans un processus de type 2TUP

Ainsi basés sur un processus 2TUP (2-Track Unified Process, dont la forme en Y est caractéristique), une implémentation de Unified Process, nous utilisons à la fois des modèles existants, que nous pouvons étendre et transformer, et de nouveaux modèles, dans le but d'atteindre une architecture finale supportant les fonctionnalités souhaitées. Les processus, à l'image de 2-TUP, sont guidés par les besoins utilisateurs, génériques, itératifs et incrémentaux et centrés sur l'architecture.

Enfin, nous proposons des outils facilitant le processus de transformation des modèles jusqu'à obtenir un code interprétable.

4.1.2 CoCSyS comme exemple de démarche MDA

Depuis de nombreuses années, l'équipe LIRIS de l'ECL (anciennement LIESP) travaille sur la problématique d'utilisation des technologies de réalité mixte dans différents domaines. Nous cherchons ici à présenter succinctement certains travaux de l'équipe, en se concentrant principalement sur la méthodologie de conception d'un système de réalité mixte CoCSyS.

4.1.2.1 Introduction

La construction du système collaboratif mobile tenant compte des contextes est un travail complexe. Pour l'organiser et pour rapprocher les développeurs et les utilisateurs, une démarche appelée CoCSyS (COoperative Capillary SYStem) pour la construction et l'évolution des systèmes collaboratifs mobiles a été présentée par [Delotte, 2006]. L'approche par les modèles, et leur transformation permettant la contextualisation et l'adaptation sont des caractéristiques majeures de CoCSyS.

CoCSyS est un processus qui peut être considéré comme un cycle de vie d'applications collaboratives mobiles centrées utilisateurs. Sa finalité est la création d'applications collaboratives ayant comme point de départ l'identification des besoins à l'aide de scénarii appropriés (scénarii contextualisés). A partir des besoins exprimés par les utilisateurs, la conception de l'application collaborative est formalisée et assistée.

En général, CoCSys est un processus :

- Centré utilisateur (participation des utilisateurs dans les quatre phases)
- Basé sur les scénarios contextualisés
- Basé sur les modèles (modèles de scénarios, modèle comportemental, modèle d'architecture)

4.1.2.2 Principes

Figure 14. Processus CoCSys (d'après [Delotte, 2006])

La figure 14 ci-dessus illustre une vue globale du processus CoCSys. Ce processus se décompose en 4 phases principales, dont la quatrième phase et l'évolution du processus et du système, qui permet à réintégrer les changements des contextes des utilisateurs ou les scénarios du travail collaboratif. Les trois phases de développement sont :

Collecte des scénarios : Les scénarios sont collectés durant les discussions avec les utilisateurs potentiels. Ces scénarios sont reliés à des tâches spécifiques ou activités d'utilisateurs particuliers (décrit comme dans la figure 15).

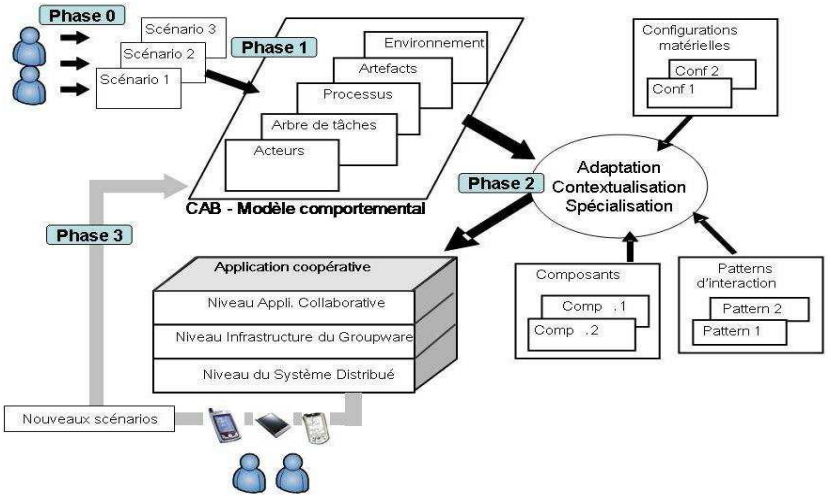


Figure 15. Démarche CoCSyS (d'après [Delotte, 2006])

Elaboration du modèle comportement (CBM - Cooperative Behaviour Model) : Ce modèle de comportement coopératif caractérise l'application collaborative concernée, contient des acteurs concrets, des artefacts, des tâches et contextes que cette application coopérative prendra en compte (Figure 16). Les scénarii sont étudiés afin d'en extraire les tâches des différents acteurs et de déterminer les contraintes fonctionnelles et temporelles permettant de définir l'organisation de ces activités, en vue de bâtir le modèle comportemental de référence.

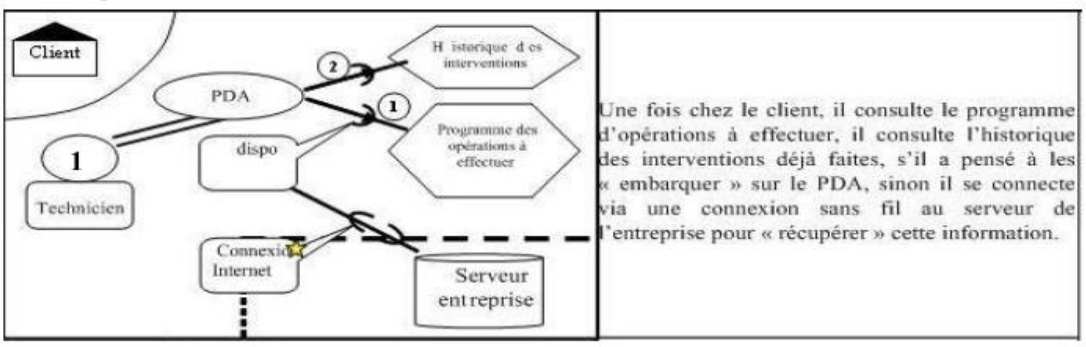


Figure 16. Informations synthétisées dans le modèle comportemental (d'après [Delotte, 2006])

Projection du modèle comportemental sur l'architecture collaborative. A partir du modèle comportemental, une transformation (au sens Model Driven Architecture) basée sur des modèles de plateforme matérielle, des patterns d'interactions et des composants permettent de projeter une architecture adaptée à la plateforme d'application coopérative. Cette architecture est structurée en 3 niveaux :

- **Le niveau de système distribué** (modèle DSI : Distributed System Infrastructure) est en charge essentiellement de la distribution des messages et de la gestion du contrôle du contenu. Ce niveau est orienté système qui fournit les mécanismes pour la communication et la synchronisation des composants distribués qui ne sont pas adaptés au travail collaboratif.
- **Le niveau intermédiaire** (modèle CSA : Collaborative System Architecture) est un niveau générique entre l'application et le système distribué. Il contient les éléments communs de l'activité du groupe et peut être vu comme un « système d'exploitation » dédié aux collecticiels. Il supporte le travail collaboratif en contrôlant les sessions, les utilisateurs et les groupes, fournit des outils coopératifs génériques et est responsable de la gestion de la concurrence.
- **Le niveau de l'application collaborative** (modèle CUO : Collaboration User-Oriented) utilise les services du niveau intermédiaire. Ce niveau regroupe les éléments spécifiques à l'application développée. Elle englobe notamment ce qui a trait à l'interface utilisateur.

4.1.3 Différents modèles utilisés

4.1.3.1 Orchestra pour modéliser la collaboration

Dans notre approche basée sur des modèles, il s'agit de décrire les spécifications du système dans un formalisme de niveau élevé. [David et al. 2005, 2006] propose un cadre pour la conception, l'exécution et l'évolution de SCC (systèmes coopératifs capillaires). Ce formalisme appelé ORCHESTRA a pour objectif de fournir une expression visuelle du modèle de comportement coopératif d'un SCC. Il exprime le modèle coopératif de comportement (CBM), élaboré à partir d'une collection de scénarios, montrant de façon macroscopique l'orchestration des activités collaboratives. Ce formalisme, facilement compréhensible permet d'associer différents acteurs à ce processus constructif. Il peut être employé pendant des discussions initiales aussi bien que pendant le processus de mise en œuvre et d'adaptation.

Comme des scénarios sont des histoires limitées, exprimées principalement par différents acteurs, l'objectif du modèle coopératif de comportement (CBM) est de formaliser l'organisation globale du système coopératif. Ce modèle complet vise à contrôler le comportement du système coopératif et sera employé pendant le processus de mise en œuvre c.-à-d. la projection de ce modèle sur des architectures matérielles, de réseau et logicielles particulières. Les éléments principaux du modèle CBM sont :

- Les **acteurs ;** un acteur est une instanciation d'un ou plusieurs rôles, un rôle est un élément de base du comportement humain dans le système, qui peut être qualifié en tant que (A) agissant, observant (O) ou éditant (E).
- Les **activités** ; une activité décrit un travail identifié qu'un rôle peut effectuer, cette activité peut être pour un acteur également A, O ou E, c.-à-d. agissante, observante ou éditante.
- Les **processus** ; un processus est modélisé par un graphe états-transitions, dont chaque élément peut être qualifié pour un acteur par A, O ou E.
- Les **artefacts** : l'artefact peut être un outil ou un objet. L'outil est un instrument utilisé dans la tâche ; l'objet peut intervenir comme entrant dans la tâche, supportant la tâche

ou sortant de la tâche (comme résultat), il peut être qualifié pour un acteur par A, O ou E, i.e. l'acteur pouvant manipuler (agir), observer ou éditer l'artefact.

- Les **contextes** ; un contexte est une collection de trois aspects : la ou les plateformes, les lieux (logique, physique ou géographique) et les préférences des utilisateurs.

L'objectif d'ORCHESTRA est de proposer un formalisme qui peut exprimer ensemble les aspects principaux du CBM. ORCHESTRA adapte la notation des **partitions musicales** (Stewart, 1999) à notre problème de description de CBM. Pour nous, les 5 lignes d'une **portée** expriment les 5 aspects principaux du CBM (figure 15), qui sont : le rôle de l'utilisateur, l'activité, l'état ou la transition de processus où l'activité se passe, les artefacts impliqués dans l'activité et le contexte. Ces aspects sont exprimés sur chacune des lignes en y situant une ou plusieurs « **notes** » contenant un identifiant (nom). Chaque note peut recevoir une « hampe » qui indique sa participation (agissant, observant ou éditant). Nous distinguons l'acteur principal (double flèche) de l'acteur secondaire (flèche simple) ainsi que le rôle actif ✲du rôle passif ◎.

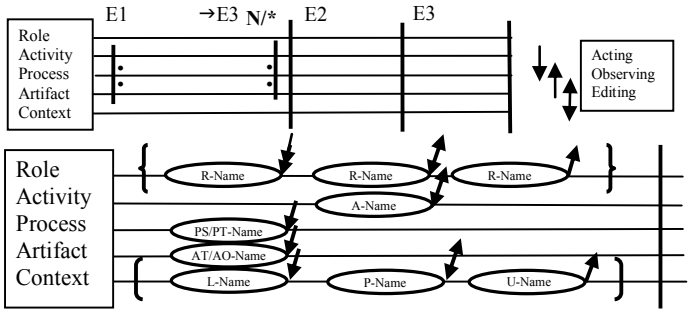


Figure 17. Principaux concepts d'ORCHESTRA

Différents signes placés au-dessus de la partition expriment des propriétés de collaboration comme des collaborations synchrones et asynchrones, des modes de collaboration et des modèles de coordination :

- @ : **asynchrone** avec temps de réponse infini ;
- @@ : asynchrone avec temps de réponse limité correspondant à l'**astreinte** ;
- & : coopération - collaboration synchrone **en réunion** ;
- && : collaboration synchrone **étroite**.

Orchestra propose également des patterns de collaboration dont l'objectif est d'exprimer de façon réutilisable des configurations / situations principales de coopération / collaboration.

- **Rendez-vous d'intervention** : Coopération synchrone ou asynchrone, sur appel ou en réunion avec une coordination implicite informatique et aucune conscience de groupe.
- **Consultation–vote** : Coopération synchrone, en réunion avec une coordination informatique et implicite et une conscience de groupe globale ou aucune conscience.
- **Présentation** : Coopération synchrone et en réunion avec une coordination sociale et explicite et une conscience de groupe globale.
- **Travail étroit** : Collaboration synchrone et détaillée avec une coordination explicite informatique et une conscience de groupe partielle.
- **Questions/réponses** : Activité synchrone, en réunion avec une coordination explicite sociale ou informatique.
- **Validation** : Coopération asynchrone et sur appel avec une coordination implicite et aucune conscience de groupe.

4.1.3.2 PAC (Présentation-Abstraction-Contrôle)

Similaire à MVC (Modèle-Vue-Contrôleur), PAC est un patron de développement introduit par [Coutaz, 1987], il stipule qu'un logiciel interactif peut être organisé comme une hiérarchie de composants constitué chacun de trois facettes :

- **La Présentation** prend en charge l'interaction avec l'utilisateur. L'ensemble des facettes de Présentation constitue la partie du programme purement dédiée à l'IHM.
- **l'Abstraction** gère les données à représenter. L'ensemble des Abstractions constitue le noyau fonctionnel.
- **Le Contrôle** gère la correspondance entre les deux autres facettes : cohérence des représentations avec les données internes, conversion des actions de l'utilisateur en opérations du noyau fonctionnel. Les facettes de contrôle servent aussi à créer une hiérarchie de composants logiciels pour organiser le programme : la facette de contrôle du composant parent, communique avec celle du composant fils.

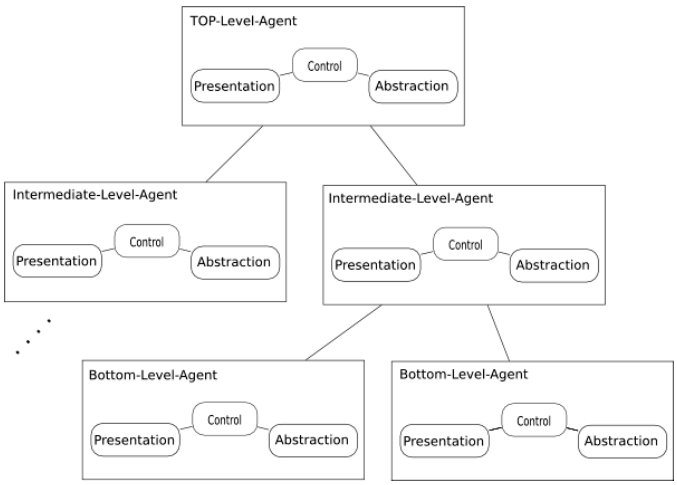


Figure 18.Structure des agents

PAC ne décrit pas sous quelle forme doivent être réalisées et connectées les différentes facettes.

4.1.3.3 AMF-C

AMF-C [Tarpin-Bernard, 1999], basé sur AMF [Ouadou, 1994] a introduit une généralisation des facettes et l'explicitation de mécanismes liés au travail coopératif.

Le modèle AMF part sur deux critiques du modèle PAC :

- La décomposition Abstraction/Présentation est généralement insuffisante pour des applications complexes. Des fonctionnalités se retrouvent mélangées dans des composants trop macroscopiques alors qu'elles relèvent de thématiques différentes.
- La structure de la facette Contrôle est peu formalisée. Or cette facette est la clef de voûte des modèles d'architecture.

Pour répondre à la première critique, AMF reprend les même facettes de base P, A, C mais permet d'en ajouter de nouvelles pour des besoins plus particuliers ce qui permet un

découpage plus fin des agents interactifs. Pour répondre à la deuxième critique, AMF propose un formalisme complet pour modéliser la facette Contrôle et représenter les échanges entre les différentes facettes de l'agent et entre les différents agents. La modélisation du contrôle est basée sur 2 concepts :

- **Les ports de communication**, au niveau des facettes, représentent les services qui sont offerts par la facette (ports en entrée) et ceux qui sont nécessaires (ports en sortie). Un troisième type de port est proposé (port en entrée/sortie) qui correspond à un service offert dont l'activation se conclut par l'invocation d'un autre service distant).

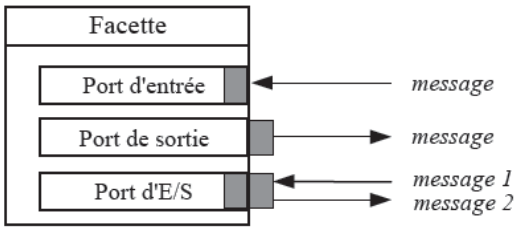


Figure 19. Représentation des ports de communication (d'après [Tarpin-Bernard, 1997])

- **Les administrateurs de contrôle**, au niveau de l'agent, connectent les ports de communication entre eux. Un administrateur de contrôle joue trois rôles :
  - Un rôle de connexion qui consiste à gérer les relations logiques pouvant exister entre les ports de communication qui lui sont attachés.
  - Un rôle comportemental qui exprime les règles d'activation de l'administrateur, c'est-à-dire sous quelles conditions et à quel moment les messages émis par les ports sources seront transmis aux ports cibles.
  - Un rôle de traduction qui consiste à transformer les messages émis par les ports sources en messages compréhensibles par les ports cible.

Dans AMF-C, les utilisateurs interagissent avec les agents selon 2 approches différentes :

- Soit ils interagissent tous avec un seul agent. Celui-ci doit être fragmenté en plusieurs morceaux répartis sur les différents postes de travail et éventuellement sur un ou plusieurs serveurs.
- Soit ils interagissent chacun avec leur propre copie de l'agent qui est donc répliqué sur chaque poste de travail.

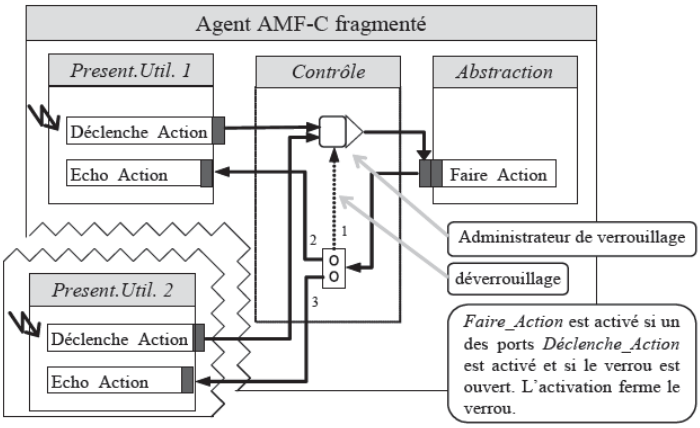


Figure 20. Exemple d'AMF-C avec administrateur de verrouillage selon [Tarpin-Bernard et al., 1998]

4.1.4 Le modèle objet

Le modèle objet, ou « game object model » est au cœur du modèle collaboratif et comportemental du jeu et est la manifestation abstraite des éléments du jeu pendant l'exécution. Tout élément interactif est un « game object ». Cela comprend : les objets et outils du domaine de la tâche (ce que les joueurs manipulent), les zones dans lesquels coexistent ces derniers, les informations relatives au jeu (une barre de point de vie, le temps restant, etc.), les éventuels ennemis, etc.

La façon dont le modèle objet est structuré a un impact fort sur :

- La flexibilité du modèle : peut-on le modifier facilement pour ajouter/enlever des choses ?
- La maintenabilité du modèle : peut-il supporter facilement des évolutions futures ?
- Les performances

Il existe de nombreuses possibilités de structurer ce modèle objet, mais la plupart des moteurs choisissent généralement entre deux styles architecturaux basiques : le modèle centré sur l'objet et le modèle centrée sur les propriétés [Gregory, 2009].

4.1.4.1 Modèle centré sur les objets

Dans un modèle centré sur les objets, chaque objet est représenté à l'exécution par une instance de classe ou une petite collection d'instances de classes. Chaque objet a ainsi une collection d'attributs et de comportement encapsulés dans l'objet. Le monde est alors une collection de « game objects » où l'héritage est la façon de propager des attributs et comportements communs.

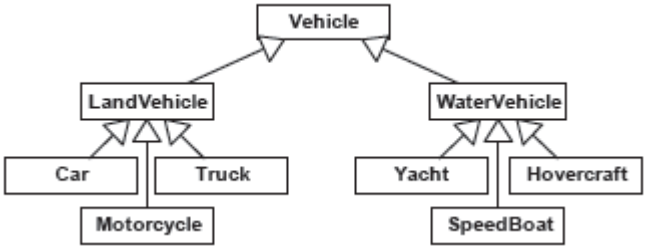


Figure 21. Une hiérarchie hypothétique de classes représentant des véhicules

Cette approche est puissante et intuitive, et incite les développeurs à naturellement créer une hiérarchie de classe, effectuant ainsi une classification taxonomique des objets du jeu. La plupart des jeux commerciaux utilisent un modèle centré sur l'objet. Alors que l'on commence généralement avec une hiérarchie petite et simple, il faut ensuite étendre le modèle pour ajouter de plus en plus de fonctionnalités spécifiques.

Le problème de cette approche est qu'elle est incapable de gérer des taxonomies multidimensionnelles, toute hiérarchie ne pouvant classer les objets que selon un seul « axe » : l'on se dirige alors fatalement vers une hiérarchie monolithique de plus en plus profonde et large, donc complexe et peu flexible et surtout difficile à maintenir.

Alors que l'héritage multiple semblerait être une solution pouvant pallier à cette impasse, elle est également connue pour conduire à d'autres problèmes pratiques typiques et compliqués à résoudre tels que le « Deadly diamond » : de quelle classe hériterait une classe « véhicule amphibien » ?

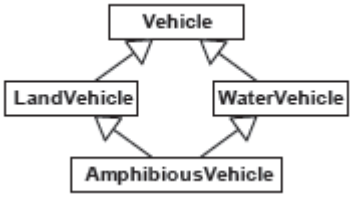


Figure 22. Créer une classe « véhicule amphibien » nous mène au fameux « Deadly diamond » : l'instance d'un véhicule amphibien contiendrait alors deux copies internes, l'une héritant de la classe du véhicule terrestre, l'autre du véhicule aquatique

Des méthodes existent pour pallier à ce problème, telles que l'utilisation d'interfaces (ainsi qu'utilisées en Java ou ActionScript), néanmoins elles ne mènent qu'à une profondeur et une complexité du modèle objet plus grandes encore.

4.1.4.2 Modèle centré sur les propriétés

Dans une autre optique, le modèle centré sur les propriétés permet de réduire considérablement la profondeur et la complexité du modèle objet centré sur les objets. Les game objects deviennent alors non pas des fils d'objets héritant des mêmes propriétés mais des objets *composés* de ces propriétés. Les propriétés deviennent alors des composants d'un objet, et peuvent inclure à la fois des attributs de l'objet (des points de vie, des roues) que des comportements tels que « peut mourir », ou « peut évoluer sur l'eau ». Alors ces derniers exemples sont d'ordre fonctionnel, on peut également utiliser cette approche pour des problèmes d'ordre plus technique, en utilisant

des propriétés du type « peut être animé » ou bien « possèdent une texture » etc. Pour y parvenir il faut convertir les relations entre classes de « *est un* » en « *a un* » :

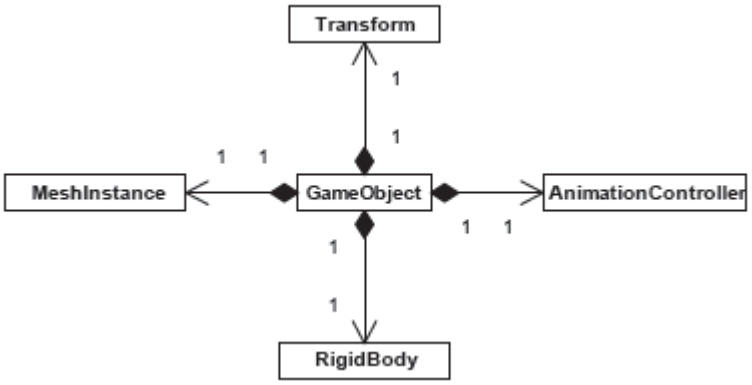


Figure 23. Notre game object devient un « hub » de composants

Un autre avantage est que ces propriétés ou composants peuvent alors être facilement décrits par les données. Le Game Object n'est alors plus que le « Hub » de ses propriétés : un identifiant unique centralisant le point de départ de la liste de propriétés affectées. Cette approche, dont la flexibilité est une qualité essentielle pour notre moteur, est celle qui a été retenue pour notre modèle fonctionnel (5.2.2).

4.1.5 IRVO : modélisation des environnements mixtes

Afin de prendre en compte la réalité mixte dans notre modèle, nous avons utilisé le modèle IRVO (Interacting with Real and Virtual Objects) [Chalon, 2004], couplé au modèle proposé par le projet Caméléon [Calvary et al. 2003]. IRVO a pour but de modéliser les interactions entre un ou plusieurs utilisateurs et le système de réalité mixte en représentant explicitement les objets et outils impliqués ainsi que leurs relations. La prise en compte de ces modèles nous mène à introduire deux nouveaux éléments mixtes dans la conception de notre moteur : la zone numérique et l'outil mixte.

Interacting with Real and Virtual Objects, IRVO, est un modèle conçu par [Chalon, 2004] qui permet de représenter 3 catégories principales d'entités :

- L'utilisateur (U), ou plus généralement plusieurs utilisateurs dans un système collaboratif
- Les objets qui peuvent être perçus et manipulés par les utilisateurs. Ils peuvent être des objets du domaine de la tâche (O) ou les outils (T), objets intermédiaires permettant aux utilisateurs d'interagir avec les objets du domaine de la tâche
- Le modèle interne (M) de l'application, représentant l'application sans la couche de présentation concrète.

IRVO introduit également le concept de « limite », permettant de représenter certaines propriétés des entités. Il existe deux types de limite :

- Entre le monde réel et le monde virtuel, qui peut être traversé par le biais de « transducteurs ». Les transducteurs permettent de capter les actions utilisateurs (senseurs) au monde virtuel, ou de transmettre une action provenant du monde virtuel

aux utilisateurs (effecteurs). Cette limite est représentée par une ligne horizontale pointillée

- Entre différents lieux du monde réel, représentée par une ligne pleine verticale

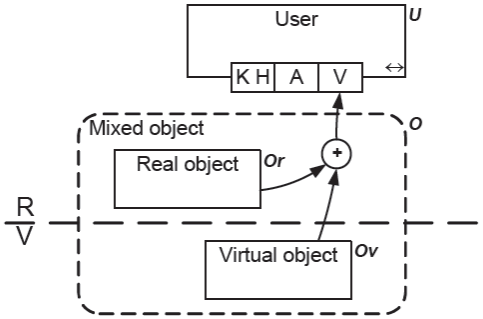


Figure 25. Exemple de représentation IRVO d'un objet mixte. En haut on trouve le monde physique, les objets physiques et l'utilisateur. En bas, les objets virtuels. Les pointillés représentant la limite sur laquelle devraient être placés les transducteurs

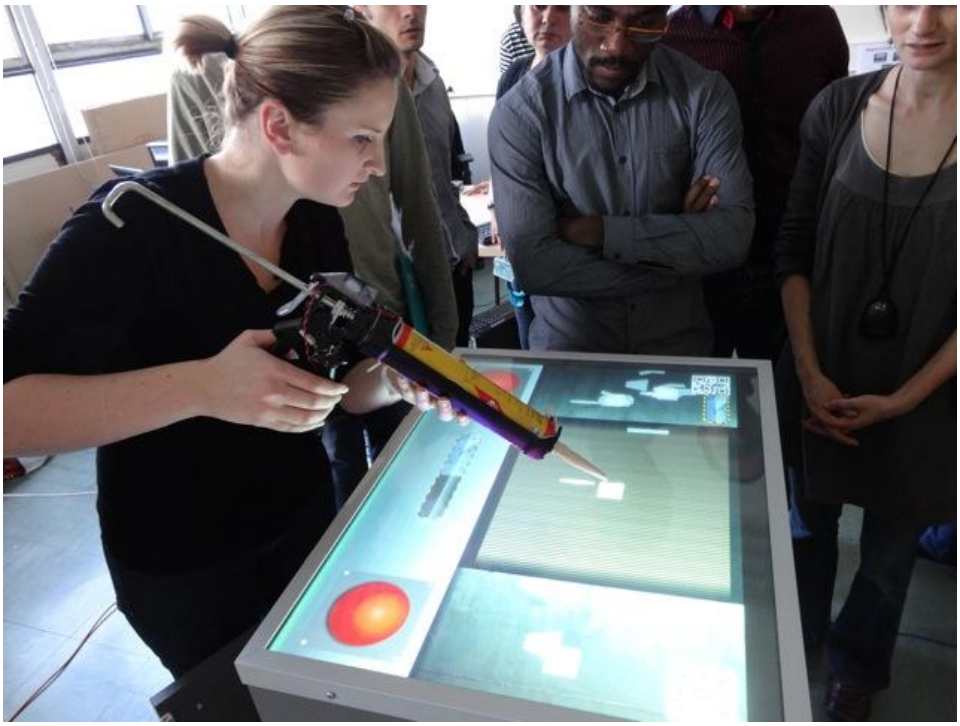


Figure 24. Tâche utilisant le pistolet à colle mixte, pendant les expérimentations de Lea(r)nIt. Une led placée au bout du pistolet et actionnée par un interrupteur permet de « déposer de la colle » sur des éléments avant de les assembler ensemble pour former un squelette de réplicants.

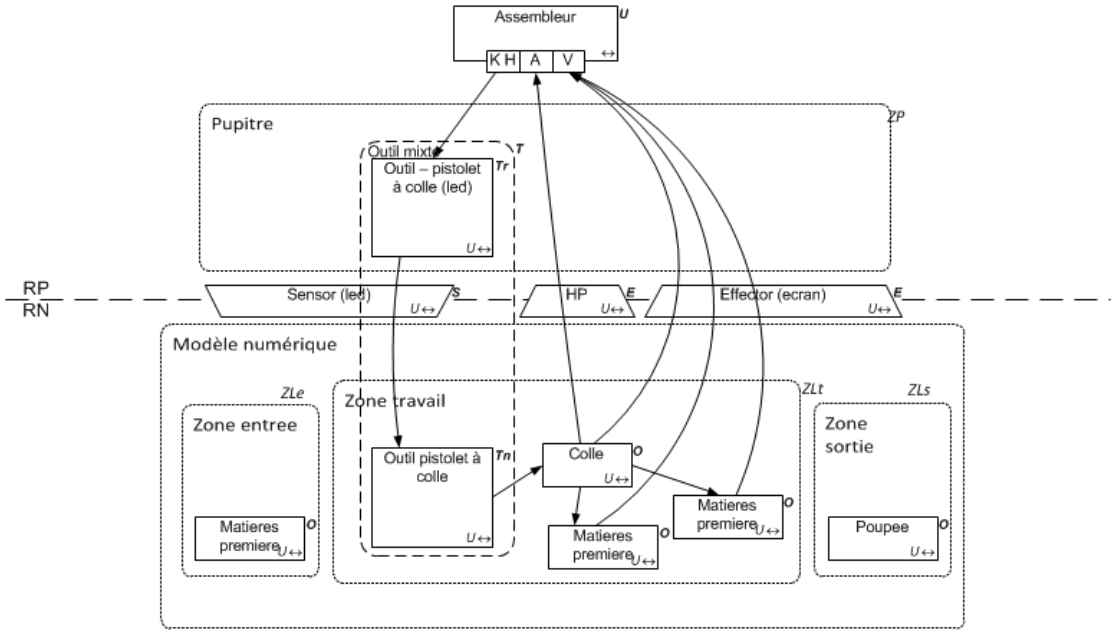


Figure 26. Représentation IRVO de la tâche présentée figure 24.

4.1.5.1 Intégrer IRVO à AMF

[Chalon, 2004] propose d'intégrer les spécificités du monde mixte en ajoutant aux agents AMF une facette « Réel » associée à la représentation physique des entités. IRVO modélise la présentation réelle et numérique des différents outils/objets présents dans l'interface et peut être couplé à l'agent AMF associé. IRVO couvre l'interface concrète et l'interaction alors qu'AMF est quant-à-lui utilisé à la fois pour le noyau fonctionnel, le dialogue et la présentation abstraite (figure 27).

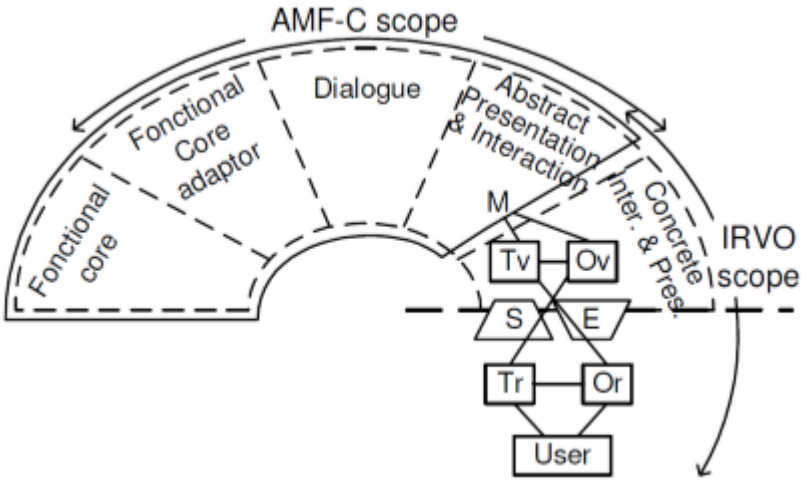


Figure 27. Lien conceptuel en AMF et IRVO

Du côté du modèle numérique, un outil mixte est un agent spécifique auquel on a ajouté une facette « réel » sous la forme d'une FUI (final user interface, 5.2.3) spécifique et écoutant les évènements produits par l'outil tangible. Du côté du monde physique, il est un objet tangible pouvant être capté par un dispositif physique et dont les actions sont transmises à la FUI de l'outil logique.

4.2 Spécifications d'un Learning Game mixte

La figure 28 montre une vue haut niveau de l'application SEGAREM, de la définition du jeu pédagogique à son exécution.

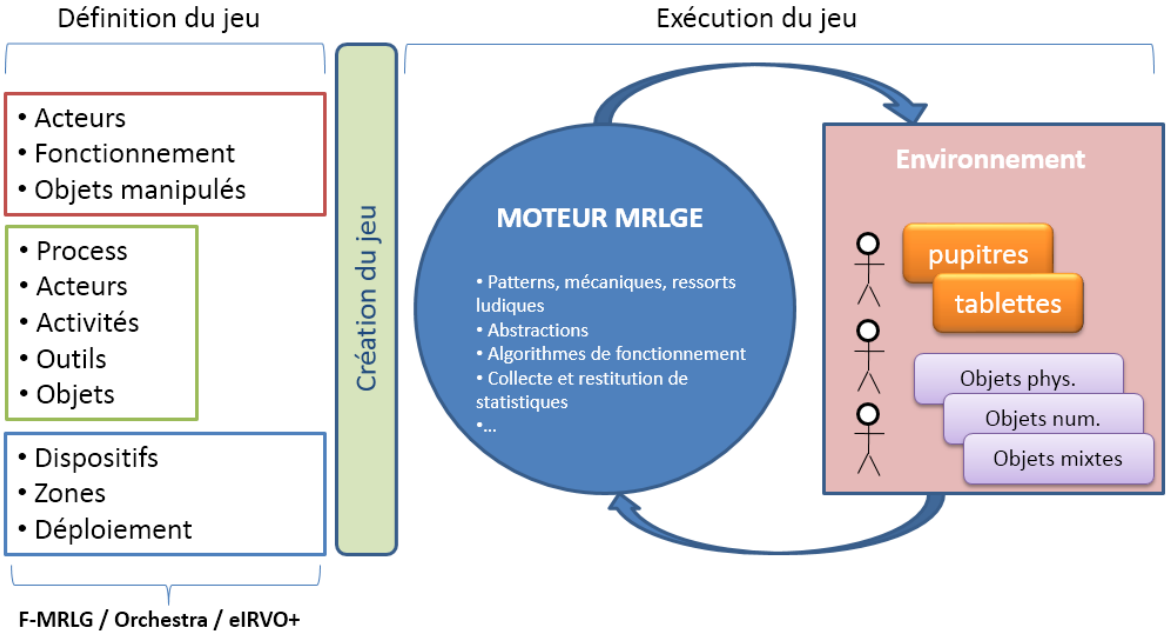


Figure 28. Vue haut niveau de l'application SEGAREM

Nous définissons en premier lieu les éléments physiques et logiques principaux, ainsi que leurs relations, dont est constitué notre monde numérique (appelé simplement « monde » dans la suite du document sauf explicitation). Viennent ensuite les éléments physiques, qui sont des éléments intervenant dans le monde physique : les personnes, les dispositifs et leur disposition physique, les objets tangibles. Les éléments logiques sont les éléments existant dans le modèle numérique. Enfin, les éléments mixtes sont ceux devant exister simultanément dans les deux mondes physique et numérique.

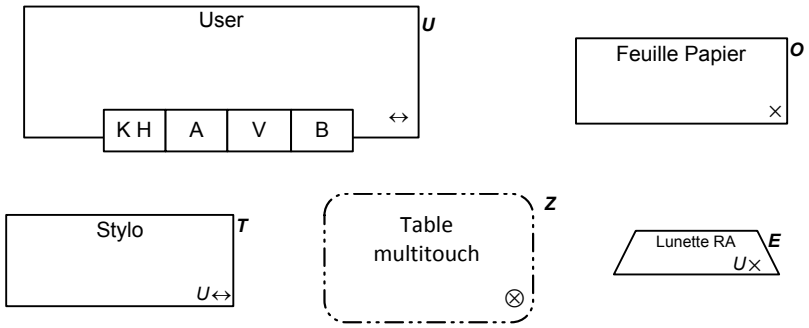


Figure 29. Différentes entités modélisées avec IRVO+

4.2.1 Eléments physiques

4.2.1.1 Acteur

Un acteur est une personne réelle interagissant avec le jeu. Un acteur peut avoir un ou plusieurs rôles. Dans le cas de Lea(r)nIT, il s'agît des participants au jeu ainsi que le tuteur.

4.2.1.2 Dispositif

Un dispositif est un périphérique physique servant de support à l'interaction entre les acteurs et le monde numérique. Un dispositif peut permettre la perception et la manipulation (directe ou non) d'objet du monde, via des interfaces (tangibles ou non), il peut aussi uniquement permettre la restitution (un écran de projection) et non la manipulation. Dans le cas de Lea(r)nIT nous avons les dispositifs suivants : pupitres, tablettes, table surface, smartphones, écrans de projection. Ils sont exposés en détail en 6.5.

4.2.1.3 Zone Physique

Liée à un dispositif, une zone physique représente un espace physique dans lequel peuvent être représentés des zones numériques (4.2.3), elles-mêmes constituées de zones logiques. Une zone physique peut aussi être représentée par un QR code, pour contextualiser l'action (5.2.2.5). Alors qu'un dispositif (un pupitre) héberge généralement une seule zone physique, il est possible d'en subdiviser l'espace en plusieurs sous-zones physiques.

4.2.1.4 Objet Physique

Ce sont des objets tangibles, manipulable voire préhensibles par l'acteur, et peuvent dans notre cas avoir une existence au niveau du modèle numérique : l'objet devient alors mixte.

4.2.1.5 Environnement physique

L'environnement physique est le contexte physique localisé dans lequel le jeu se déroule. En l'occurrence il s'agît d'une salle unique suffisamment spacieuse pour accueillir les acteurs et dispositifs physiques du jeu. La disposition spatiale de ces derniers influence de manière significative les performances des joueurs face au jeu, et peut évoluer en cours de jeu. Le fait de pouvoir modifier cette topologie physique des postes est un des éléments déterminant dans le jeu LEAN (6.1.2).

4.2.1.6 Action, tâche et activité physiques

Une action physique est effectuée par l'acteur pour accomplir une tâche. Elle consiste donc pour un acteur à interagir avec le(s) dispositif(s) dans une ou plusieurs zone(s) physique(s) et dans un environnement. Une tâche physique est un travail attendu d'être effectué par un acteur. Enfin une activité physique est un ensemble de tâches physiques réalisées dans un but, prescrit ou non.

4.2.2 Eléments logiques

Le monde (numérique) est un ensemble de zones logiques connectées entre elles et d'objets et d'outils contenus dans ces zones. Les acteurs peuvent interagir avec les objets et zones du monde par le biais d'outils et selon leur rôle.

4.2.2.1 Zone logique

La zone logique est l'unité d'espace logique à travers laquelle les acteurs peuvent interagir avec le monde numérique, notamment avec les objets et outils logiques. Une zone est une forme géométrique finie (un carré, un rectangle, un cube…) s'exprimant ou non via une zone numérique, et à l'intérieur de laquelle peuvent coexister objets et outils. Une zone logique est toujours identifiée par un nom.

Dans le cas Lea(r)nIT chaque dispositif contient une ou plusieurs zones, par exemple le dispositif dédié à l'opération de collage possède une zone de contrôle (pour actionner un tapis roulant, recharger le pistolet, etc.), une zone d'entrée (par où arrivent les objets à assembler), une zone de travail (pour l'opération d'assemblage à proprement parler, à l'aide de l'outil mixte pistolet à colle par exemple) et une zone de sortie (pour y déposer les objets assemblés).

4.2.2.2 Objet logique

Un objet logique (sous-entendu du domaine de la tâche) est la plus petite unité logique avec laquelle un acteur peut interagir. Dans le monde du cas Lea(r)nIT, les objets logiques sont par exemple les matières premières, les réplicants, etc. Chaque objet se comporte comme un agent (4.1.3.3), a un identifiant unique et peut avoir un ensemble de propriétés ou comportements tels que :

- Manipulable
- « Collidable » (sujet à collision)
- « Gluable » (pouvant être assemblés)
- « Damageable » (sujet au dommage et donc pouvant par exemple posséder des attributs du type 'points de vie')
- Glued/Broken/etc.
- Etc., etc…

➔ Un objet logique est toujours contenu dans une zone logique.

4.2.2.3 Rôle

Le rôle est une catégorie d'acteurs, symbolisant leur rôle fictif dans le jeu. Le rôle permet de déterminer les zones ainsi que les outils et objets auxquels l'acteur peut avoir accès. La présence du concept de rôle est également justifiée par le fait qu'il est possible d'avoir plusieurs acteurs pour un même rôle. Dans le cas de Lea(r)nIT, nous pouvons distinguer les rôles suivants : manutentionnaire, assembleur, peintre, client, agent de qualité, tuteur. On peut imaginer dans un scénario amélioré de Lea(r)nIT d'avoir plusieurs acteurs jouant le rôle de manutentionnaire.

4.2.2.4 Action, tâche et activité logiques

Elles sont la représentation abstraite de leurs homologues physiques, permettant leur expression dans un modèle de type « arbre des tâches » ou « workflow ».

4.2.2.5 Outil logique

Un outil logique est un objet logique (4.2.2.2) spécifique qui permet à un rôle d'interagir avec le monde numérique, notamment d'effectuer des transformations sur les objets (de position, d'état, de propriété). Un outil peut également être mixte par l'utilisation d'interfaces tangibles (4.2.1.4).

Un outil logique est caractérisé par 3 éléments :

- La classe d'outil : le fonctionnement au plus bas niveau de l'outil, ce qu'il fait (création, transformation, etc.)
- Le type d'outil : un nom qui spécifie la définition de l'outil à un plus haut niveau (un créateur de type d'objet spécifique, un outil décrémentant une propriété particulière de ces cibles, etc.).
- L'identifiant de l'outil : hérité de l'objet logique, permettant par exemple d'avoir plusieurs instances d'un même type d'outil dans des zones différentes

Nous distinguons les outils logiques génériques suivants, chacun correspondant à une fonction élémentaire et qui définit la 'classe' d'un outil:

- Création d'objet
- Suppression d'objet
- Transformation d'objet (modification de son état, de ses propriétés, de sa position relative interne à une zone)
- Déplacement d'objet d'une zone à une autre
- Fusion de plusieurs objets en un seul objet composite
- Eclatement d'un objet composite en plusieurs
- Contextualisation physique

4.2.3 Zone numérique

Il nous faut distribuer la structure du monde établie sur les dispositifs. Pour assurer la flexibilité de la répartition, pouvant à terme être dynamique, nous introduisons la notion de zone numérique effectuant le lien entre zones physiques et zones logiques. Une zone numérique représente à un instant ou contexte donné une instance d'une ou plusieurs zones logique sur un ou plusieurs dispositif(s). L'acteur en utilisant un outil mixte peut alors interagir avec le monde numérique à travers une zone numérique.

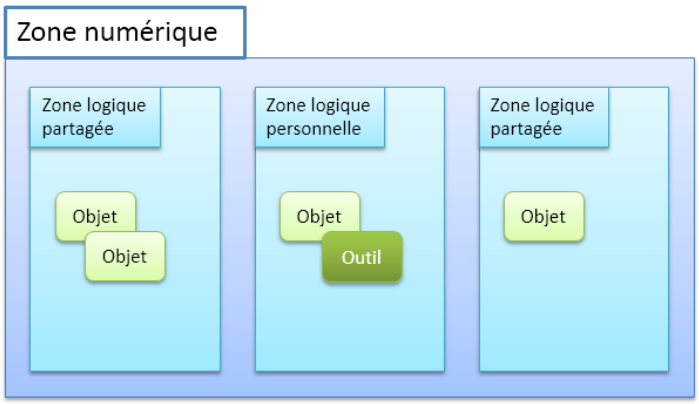


Figure 30. Une zone numérique contenant des zones logiques contenant elles-mêmes des objets et outils.

Côté logique, une zone numérique est structurée comme un agent à l'instar des autres composants du monde numérique.

Une zone numérique contient non seulement le lien vers une ou plusieurs zone(s) logique(s), mais également leur disposition finale, c'est-à-dire les informations relatives à l'interface utilisateur finale (FUI) des zones logiques concernées. Selon ces informations, une même zone logique peut avoir par exemple une position, des dimensions ou bien des façons d'afficher ou d'ordonner les objets différentes selon qu'elle soit déployée par une zone numérique ou par une autre. La distribution des zones numériques est décrite en 5.3.

Chapitre 5

Démarche, choix et outils de mise en œuvre

5.1 Méthodes agiles

Applicables dans d'autres domaines mais se limitant actuellement aux projets de développement informatique (conception de logiciels), les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles telles que le cycle de développement en V. Les méthodes agiles reposent sur une structure de cycle commune (itérative, incrémentale et adaptative), quatre valeurs communes déclinées en douze principes communs desquels découlent une base de pratiques, soit communes, soit complémentaires.

Manifeste pour le développement Agile de logiciels :

« Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire. Ces expériences nous ont amenés à valoriser :

- **Les individus et leurs interactions** plus que les processus et les outils
- **Des logiciels opérationnels** plus qu'une documentation exhaustive
- **La collaboration avec les clients** plus que la négociation contractuelle
- **L'adaptation au changement** plus que le suivi d'un plan

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers. »

Kent Beck	James	Robert C.	Jeff Sutherland	Alistair
Mike Beedle	Grenning	Martin	Dave Thomas	Cockburn
Arie van	Jim Highsmith	Steve Mellor	Ward Cunningham	Martin
Bennekum	Andrew Hunt	Ken Schwaber	Brian Marick	Fowler
	Ron Jeffries	Jon Kern		

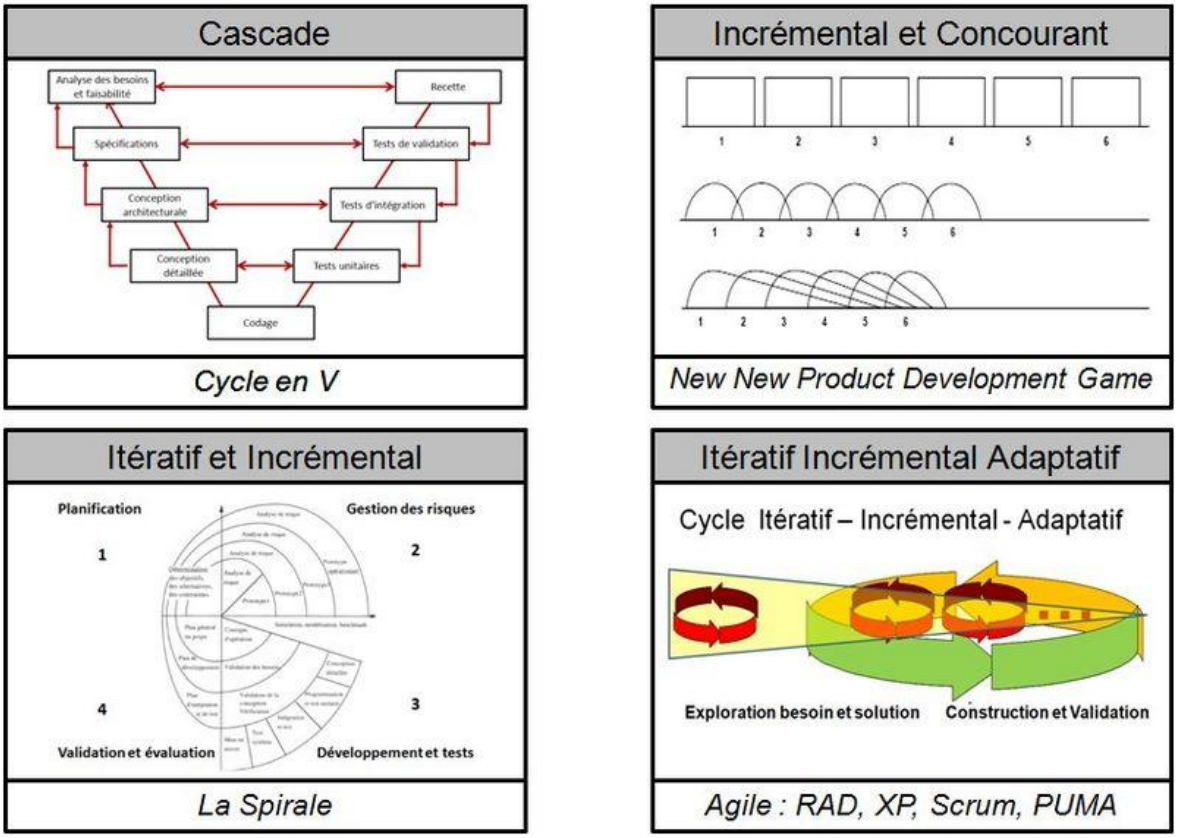


Figure 31. Cycles basiques de développement (Wikipédia)

Le choix d'une méthodologie Agile fut particulièrement adapté à la phase de développement du prototype SEGAREM, d'abord du fait que son adaptabilité facilite son insertion dans une méthodologie de plus haut niveau (telle que CoCSyS), et au vu de la taille de l'équipe impliquée au niveau du prototype (~8 personnes). Cette méthodologie, d'inspiration Kanban, était déjà adoptée en interne chez Symetrix. De plus c'est un outil structurant et synthétique: un tableau de tâches et de leur statut suffisant à transcrire autant les besoins en développement que la charge de travail en cours (figure 32). Dans un projet de recherche tel que SEGAREM, où les besoins évoluent rapidement tant en qualité qu'en quantité, et ce tout au long de la phase de développement, la simplicité d'utilisation et la nature peu chronophage des concepts Agile ont été déterminantes.

wip : work in progress. Nombre max de tâches en cours

"Stories" techniques par ordre décroissant de priorité	Étude/Modélisation		Réalisation		Validation		Documentation		Commentaires sur reste à faire
total wip	wip:	18	wip:	12	wip:	0	wip:	4	
34	in progress	done	in progress	done	in progress	done	in progress	done	
Modèles de dynamicité		x	0,4				0,75		- modes "parallèle-centré-sur-les-données" et synchro par temps court sont implémentés
Client : Approche PAC/AMF-C (Agents)		x	0,9				0,75		- ajout administrateurs de controle au niveau de l'agent - réactivité inter-agents non testée
Modèle du monde		x	0,9				0,5		
Modèle de deploiement mixte		x		x			0,75		
Client UI générique	0,9		0,5						- integration transition d'objets - 3 modes de dynamicité restant - zones logiques/numériques finales - intégration des améliorations - tests multi-utilisateurs - optimisation
Plugin Serveur	0,9		0,75						- ajout des traces - intégration des améliorations - tests multi-utilisateurs - optimisation
Gestion des améliorations	0,75		0,5						- dépendant du poste animateur
Integration TI issue des expérimentations UI									- attente des expérimentations
Poste presse		x	0,75						- intégration design
Design poste presse (Symetrix)		x	0,9						- animations presse - pièces détachées - séparer les éléments graphiques
Poste assemblage/collage		x	0,5						- pistolet à colle mixte - intégration design
Design poste assemblage/collage (Symetrix)		x	0,7						- pièces détachées - séparer les éléments graphiques
Poste manutentionnaire		x	0,9						- intégration design
Design poste manutentionnaire (Symetrix)		x							- pièces détachées
Poste "impregnation/réplication"	0,75		standby						
Design Poste "impregnation/réplication"		x							- pièces détachées et sprites de produit fini
Poste magasin surface (Symetrix)		x							
Poste contrôle qualité smartphone (Symetrix)	0,75								
Poste animateur et traces (Symetrix)	0,5								
Pré-tests grandeur nature									Probablement début Octobre
Expérimentation Learnit									24/25 Octobre 2012 :)
elRVO+		x	0,5						

Figure 32. Kanban technique au 10/09/12

La figure 32 montre l'état d'un tableau Agile décrivant l'avancement des développements. Généralement, la colonne réservée à la validation est remplie suite aux différents dialogues avec le client. Dans notre cas elle est restée vide car la plupart des « stories » n'était jamais vraiment conclue à cette période du développement les champs couverts par les stories étant peut-être un peu trop vastes (les stories auraient ainsi pu bénéficier d'un niveau de granularité plus fin); la taille de colonne « reste à faire » servait alors de guide.

5.2 Architecture fonctionnelle

Dans l'exécution du jeu, nous devons simuler un monde numérique avec lequel les acteurs du jeu puissent interagir. La simulation d'un tel monde en implique la modélisation la plus flexible possible pour être exécutable par un moteur générique exécutant les différentes mécaniques et ressorts (3.3.1) ainsi que les 5 cas de dynamicité (3.3.4) tout en supportant les modalités d'interactions multiples que permettent les dispositifs physiques dans un environnement de réalité mixte, par le biais d'outils. Enfin nous présentons le mécanisme de couches permettant de distribuer la présentation du monde numérique sur un ensemble de dispositifs physiques, lequel sert de socle à l'éditeur de distribution des zones exposé plus en détail en 5.3.

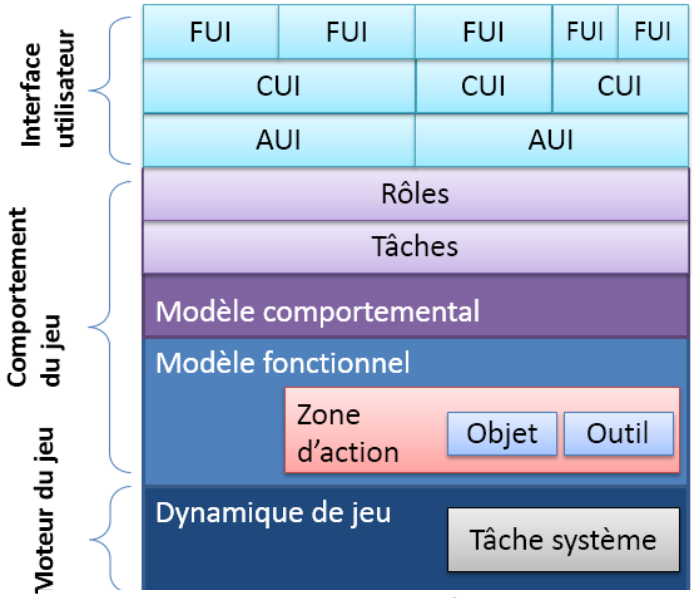


Figure 33. Architecture fonctionnelle

Nous proposons un modèle d'architecture en couches qui se compose de trois grandes parties: le moteur de jeu, le comportement du jeu et les interfaces utilisateurs du jeu (figure 33).
Le **moteur de jeu** est le cœur du modèle comportemental et fonctionnel : celui-ci connaît tous les éléments actifs du jeu et ses points d'entrée. Le comportement du moteur est décrit par un automate qui garantit la cohérence temporelle et les accès visant à la modification de l'état du modèle comportemental et fonctionnel. Le **modèle comportemental et fonctionnel**, sur lequel est projeté le modèle coopératif de comportement (CBM, 4.1.2.1) définit les tâches du jeu et la manière dont ceux-ci sont atteints, avec pour chaque tâche une décomposition en sous-tâches. Ces dernières, associées à des groupes de joueurs, permettent la prise en compte du contexte social. La troisième grande partie concerne la prise en compte des interventions des différents joueurs qui au travers des **interfaces utilisateurs** modifient le modèle fonctionnel. Les interfaces utilisateurs sont décomposées en couches selon le modèle proposé par Caméléon [Calvary, Coutaz & Thevenin, 2003] (5.2.3).

5.2.1 Moteur du jeu

Nous nous intéressons ici à la couche la plus basse, le moteur du jeu lui-même, garant de l'intégrité du modèle (ou noyau) fonctionnel du jeu selon l'axe temporel. Il coordonne les interventions des joueurs et sollicite les opérations associées dans le contexte du modèle comportemental et fonctionnel. Il a pour rôle d'assurer l'ordonnancement des activités des acteurs en conformité du style de jeu choisi. Nous proposons ici 5 styles d'ordonnancement de jeu que nous illustrons par des types de jeux :

- **Séquentialité imposée** : l'ordre d'intervention à tour de rôle des joueurs
- **Parallèle sans contrainte** : quand chaque joueur peut jouer à tout moment
- **Parallèle avec synchronisation par les données** : lorsque le comportement des joueurs est conditionné par des données partagées/actions des autres
- **Parallèle avec synchronisation par temps long** : lorsque le comportement des joueurs est conditionné par des jalons définis temporellement.
- **Parallèle avec synchronisation par temps court** : avec une synchronisation intégrale du monde à chaque unité temporelle élémentaire.

Ces modes de fonctionnement doivent être implémentés au niveau de la « boucle du jeu », gérant l'orchestration des différentes opérations.

5.2.1.1 Séquentiel

Il s'agît du cas le plus simple. Dans ce mode, chaque joueur peut effectuer des actions uniquement lorsque c'est à son tour de jouer, sinon il attend son tour. Un processus mono-tâche assure la séquence de jeu, met le noyau fonctionnel du jeu à jour et notifie les différents acteurs suite à chaque action.

Le fonctionnement de cette tâche unique peut être exprimé par la boucle suivante :

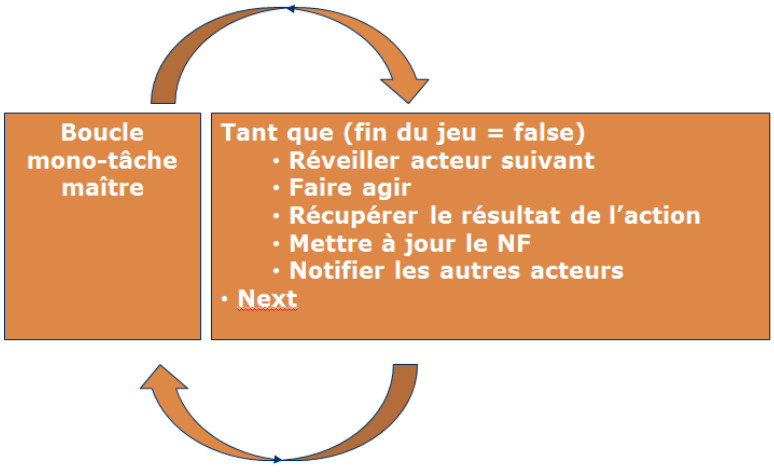


Figure 34. Modèle séquentiel

Exemple typique de jeu en séquentiel : jeu de l'oie, petits chevaux, échecs, …

5.2.1.2 Parallèle sans contrainte

Dans ce mode, les acteurs effectuent des travaux indépendamment (T) des autres acteurs et possèdent une vue (V) spécifique également indépendante des autres acteurs. Ils peuvent néanmoins mettre à jour un noyau fonctionnel commun. Un mécanisme de file d'accès au noyau fonctionnel est alors nécessaire pour en garantir l'intégrité.

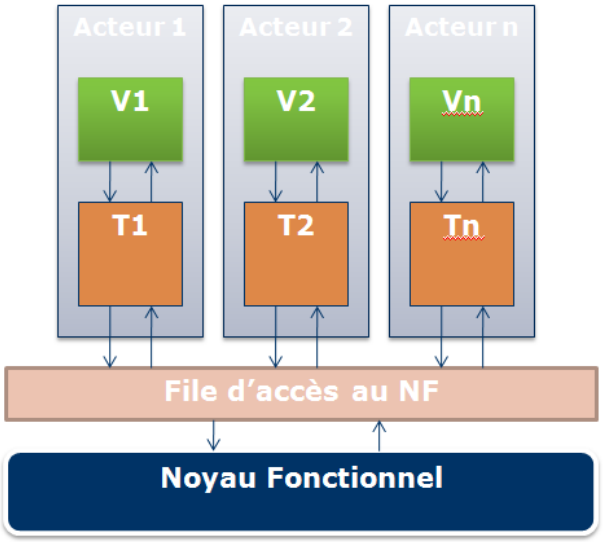


Figure 35. Travaux parallèles sans contrainte : seul le noyau fonctionnel est protégé par une file d'accès

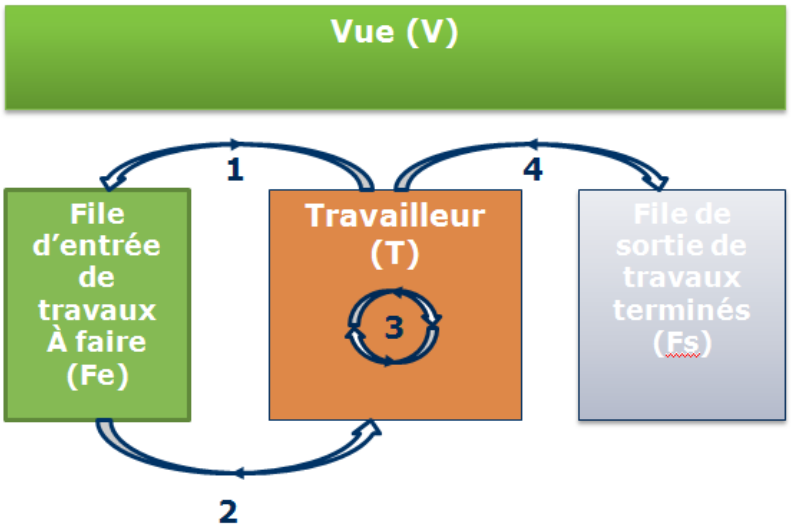
Ce cas est peu intéressant car il n'y a pas de contrainte collaborative entre acteurs au moment du travail. Néanmoins, une fois les travaux terminés, on peut comparer les résultats des différents acteurs dans le noyau fonctionnel et tirer des conclusions. Une certaine compétition implicite peut alors exister pendant l'exécution, mais n'impacte pas les travaux en parallèle.

Exemple de jeu : concours de résolution de problème mathématique.

5.2.1.3 Parallèle avec synchronisation par les données

Dans ce mode, les acteurs travaillent de façon parallèle. Chaque acteur possède une file d'entrée de travail où ils peuvent prendre un travail, ainsi qu'une file de sortie où ils peuvent déposer le travail terminé. Ils ne peuvent effectuer leur travail que lorsque leur file d'entrée leur en donne les moyens. De la même façon ils ne peuvent rendre leur travail que si leur file de sortie n'est pas pleine.

Voici le mode de fonctionnement d'une unité de travail :



1. **Si T non occupé alors T consulte Fe**
2. **Si Fe non vide alors un travail est retiré de Fe (sinon il n'y a rien à faire)**
3. **T effectue le travail**
4. **Si Fs non pleine alors T dépose le travail terminé dans Fs. Sinon T reste occupé**

Figure 36. Fonctionnement d'une unité de travail

Les travaux à effectuer sont les données (peuvent être des objets) dont la présence (l'état des files) détermine la synchronisation du système. L'activité de plusieurs unités de travail peut ainsi être exprimée de la façon suivante:

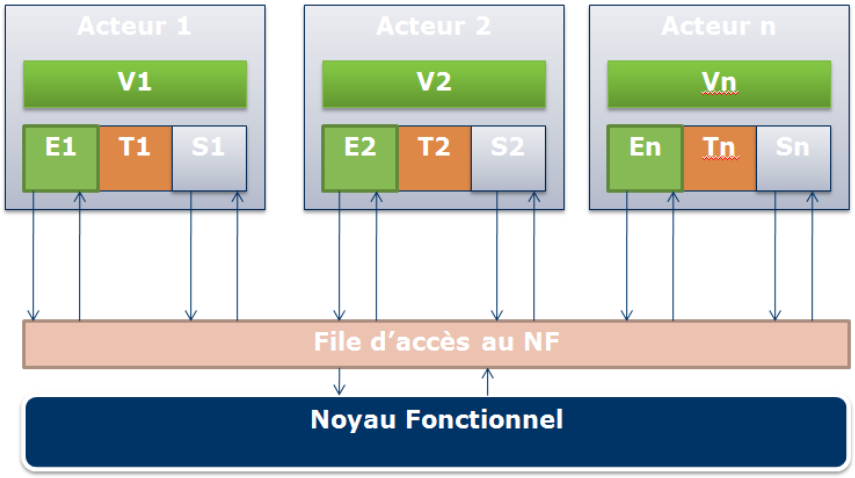


Figure 37. Travaux parallèles synchronisés par les données.

Chaque unité de travail pourrait avoir plusieurs files d'entrée (ainsi que plusieurs files de sortie), chaque file pouvant supporter/contenir un type spécifique de travail/donnée. La file d'entrée peut être alimentée soit par la file de sortie d'une autre unité de travail, soit directement par le noyau fonctionnel. La gestion de la vue n'est pas détaillée ici.

Il s'agît du mode de fonctionnement utilisé dans Lea(r)nIT (the Buckingham Lean Game).

5.2.1.4 Parallèle avec synchronisation par temps long

Dans ce mode, le comportement des joueurs est conditionné par des jalons temporels ponctuels. C'est-à-dire que les actions parallèles sont autonomes pendant une période donnée, à l'issue de laquelle les résultats sont réconciliés par le noyau fonctionnel puis redistribués aux acteurs. Une horloge est responsable de la gestion des intervalles. Elle permet de déclencher les phases de jeu, et au temps défini, de demander à un répartiteur de mettre à jour les travaux et vues des autres acteurs.

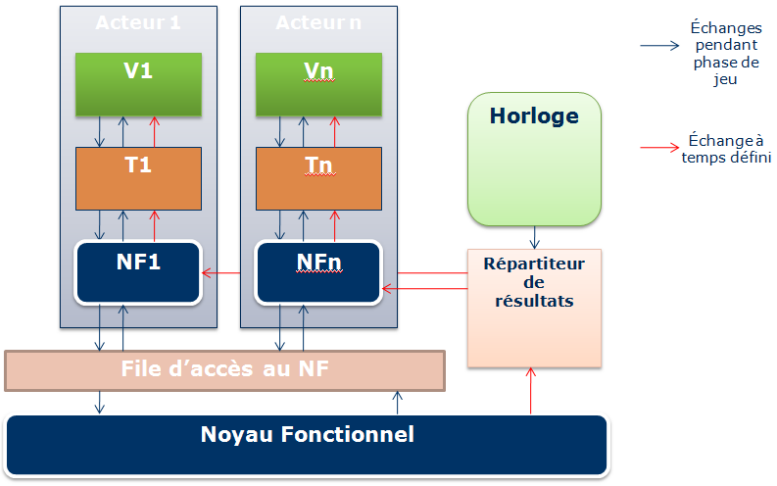


Figure 38. Fonctionnement parallèle avec synchronisation par temps long

Exemple de jeu : Reactik, une simulation de développement économique dont l'année est l'unité de temps de jeu (en temps réel, une année = 10 mn). Pendant la phase de jeu, les joueurs font des choix stratégiques (achat de matières, vente de produits, conquêtes de marchés, etc…). Tous les 'ans', le monde est mis à jour en fonction de ces actions, les joueurs entament alors une nouvelle 'année'.

5.2.1.5 Parallèle avec synchronisation par temps court

Le cas est similaire au cas « parallèle avec synchronisation par temps long », hormis le fait que les noyaux fonctionnels des acteurs, puis leur vue, doivent être mis à jour dès qu'une modification du noyau fonctionnel survient. Les mises à jour doivent donc être optimisées en étant fortement typées. Un exemple typique étant un MMOFPS (Massively Multiplayer Online First Person Shooter) tel qu'Armed Assault 2 [Bohemia Interactive, 2009], une simulation de guerre poussée dans laquelle les joueurs s'entretuent en temps réel à l'aide d'armes à feu modernes. Entre autres, la gestion de la balistique y est très réaliste, et gère notamment les trajectoires, la distance, l'influence du vent, etc. autant de paramètres qui nécessitent une synchronisation des données très optimisée : si la distance qui sépare le tireur de sa cible est suffisamment grande, le temps que met une balle à franchir cette distance peut permettre à la cible de sortir de la mire, et donc de l'éviter.

5.2.2 Modèle comportemental et fonctionnel

Le monde numérique doit comporter la totalité des éléments logiques du jeu : zones, objets, outils et rôles, ainsi que l'état de la totalité de ses composants. Le monde numérique est garant de l'intégrité de la simulation du jeu, de l'état de tous ses objets, et comporte un état initial. Nous décrivons ici la structure du monde de notre architecture.

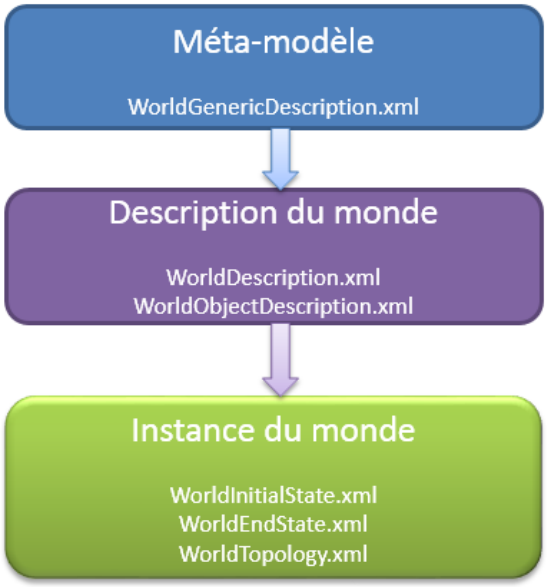


Figure 39. Méta-modèle, modèle et instance du modèle fonctionnel et comportemental

Plusieurs niveaux d'abstractions sont nécessaires pour assurer un maximum de souplesse dans le but de supporter le plus grand nombre de genre de Serious Games. Différents fichiers permettent de décrire le modèle adopté, chacun exprimant le méta-modèle du niveau inférieur:

- 1: WorldGenericDescription.xml : description du méta-modèle des classes (structure, champs) du monde
- 2: WorldObjectDescription.xml / WorldDescription.xml : description du modèle du des classes jeu (description des objets du domaine de la tâche, des zones)
- 3: WorldInitialState.xml / WorldTopology.xml / WorldEndState.xml: instanciation du jeu, à l'état initial.

Alors que ce méta-modèle est centré sur l'objet (figure 40) pour les objets de haut niveau tels que les objets, zones et outils, suivant l'approche de [Gregory, 2009] chaque agent suit ensuite une logique de composition et de propriétés plutôt que d'héritage (4.1.4.2).

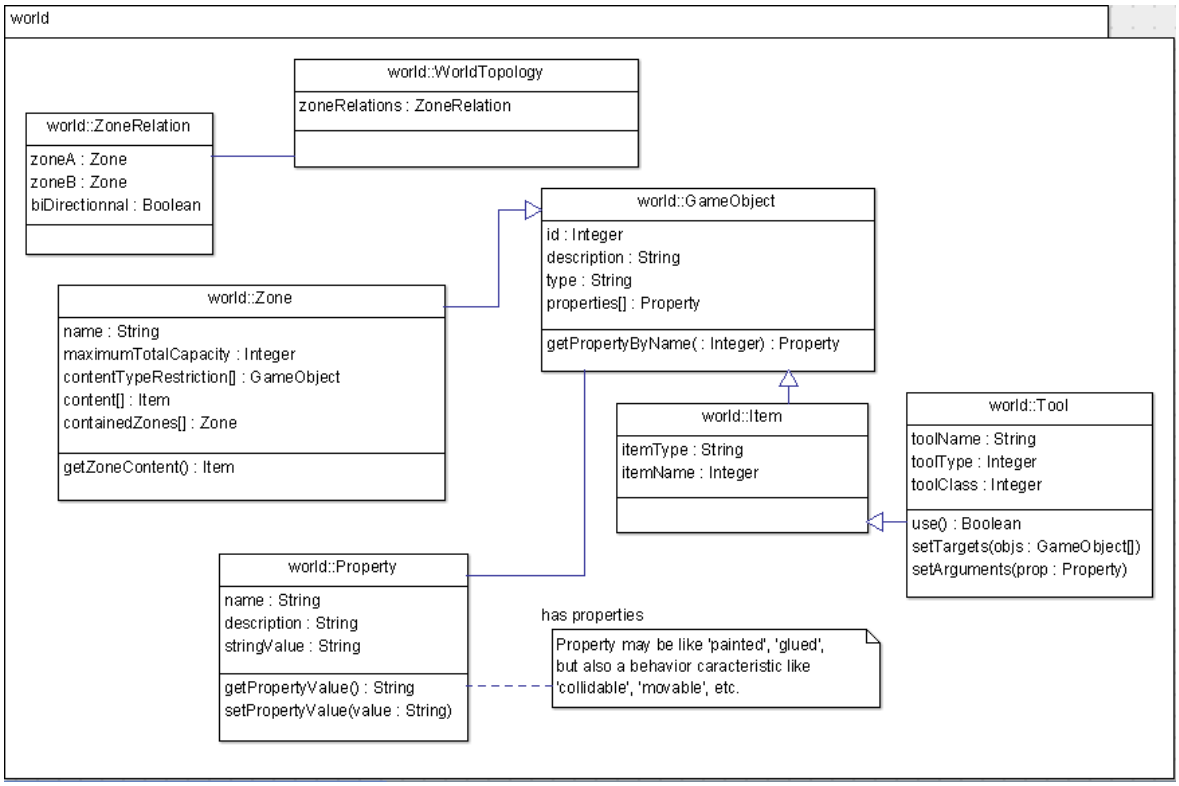


Figure 40. Meta-modèle du monde

Une fois le méta-modèle établi, nous pouvons définir nos et zones objets spécifiques. Chaque nœud 'gameObjectDescription' du fichier WorldObjectDescription.xml contient la définition d'un type de game object, les nœuds 'toolDescriptions' contenant la description d'un type d'outil.

```
<?xml version="1.0" encoding="utf-8" ?>
<worldObjectDescription>
<!-- World LearnIT object and tools -->
<!-- taken/inspired from commandObjects.xml -->
  <gameObjectDescription>
    <type>1</type>
    <name>granulat</name>
    <isLeaf>true</isLeaf>
    <description>Product entry of shaper</description>
    <properties>
      <property>
         <name>quantity</name>
         <description>Quantity of product</description>
         <type>integer</type>
         <value>15</value>
      </property>
    </properties>
  </gameObjectDescription>
  <gameObjectDescription>
    <type>2</type>
    <name>shape</name>
    <isLeaf>true</isLeaf>
    <description>Body shape that comes out from shaper machine
    </description>
    <properties>
      <property>
        <name>shapeType</name>
          <description>Shape number among {1:TRIANGLE; 2:ROUND; 3:SQUARE;}
          </description>
          <type>integer</type>
          <value>8</value>
      </property>
   </properties>
```

```
</gameObjectDescription>
```

Figure 41. Exemple de WorldObjectDescription.xml de Lea(r)nIt

```
<!-- Zones description -->
<zones>
  <zone>
    <name>Z_sto_mp</name>
    <description>Raw materials stock area</description>
    <maximumTotalCapacity>100</maximumTotalCapacity>
    <gameObjectTypeRestrictions>
      <gameObjectTypeRestriction>1</gameObjectTypeRestriction>
      <gameObjectTypeRestriction>3</gameObjectTypeRestriction>
      <gameObjectTypeRestriction>4</gameObjectTypeRestriction>
    </gameObjectTypeRestrictions>
  </zone>
...
```

Figure 42. Exemple de WorlDescription.xml extrait de Lea(r)nIt, on y trouve la zone de stockage initiale 'Z_sto_mp'.

A ce niveau, toutes les zones, les objets et outils contenus dans ces zones sont instanciés. Cet état est ensuite modifié par les acteurs au travers d'outils. Il est exprimé dans un fichier WorldInitialState.xml qui peut être sauvegardé et rechargé.

5.2.2.1 Approche Agent

Suivant [Magnusson 2011], nous avons choisi d'utiliser une programmation orientée agent (AOP). Alors que la programmation orientée objet (POO) est centrée sur le concept d'objet, il s'agît ici d'un paradigme de programmation dans lequel la construction du logiciel est centrée autour du concept d'agents, lesquels peuvent être vus comme une abstraction d'objets.

En conception, deux approches complémentaires peuvent être utilisées : l'une s'intéresse à la manière dont un système est construit et l'autre à la manière dont il fonctionne. Cette double vision, structurelle et comportementale, conditionne notre approche par l'utilisation de deux logiques : l'une multi-agent et l'autre en couche. Grace à cela, la transition entre la conception et le développement est améliorée et surveillée.

Afin de simplifier la description du système, il est possible de le décomposer en sous-systèmes. Il s'agit de lier les différents objets par une structure hiérarchique qui conduit à dégager une architecture multi-agents. Un agent est défini comme étant une entité informatique, perceptible ou conceptuelle, indépendante, à durée de vie limitée, ayant un comportement propre et capable d'agir avec son environnement. Ainsi, un agent a connaissance de son état et de son comportement et est capable d'observer et d'interagir avec les autres agents. Entre les agents cognitifs et les agents réactifs, en IHM, les agents réactifs dominent. Dans nos travaux, nous nous inspirons des architectures multi-agents comme PAC (Présentation-Abstraction-Contrôle) et AMF-C. Selon ces visions, chaque agent est défini à l'aide de trois facettes fonctionnelles : la présentation, l'abstraction et son contrôle qui décrivent respectivement le comportement perceptible/manipulable par l'agent, son modèle de données sous-jacentes et les échanges qui existent entre chacune des facettes et les agents extérieurs.

5.2.2.2 Structure d'un agent

Dans notre modèle comportemental et fonctionnel, tout objet, zone ou outil du modèle objet est un agent dont la facette présentation est étendue à l'aide des principes Caméléon et dont l'architecture est décrite dans la figure 43. Chaque agent communique avec les autres agents via sa facette « Control » et par le biais d'évènements.

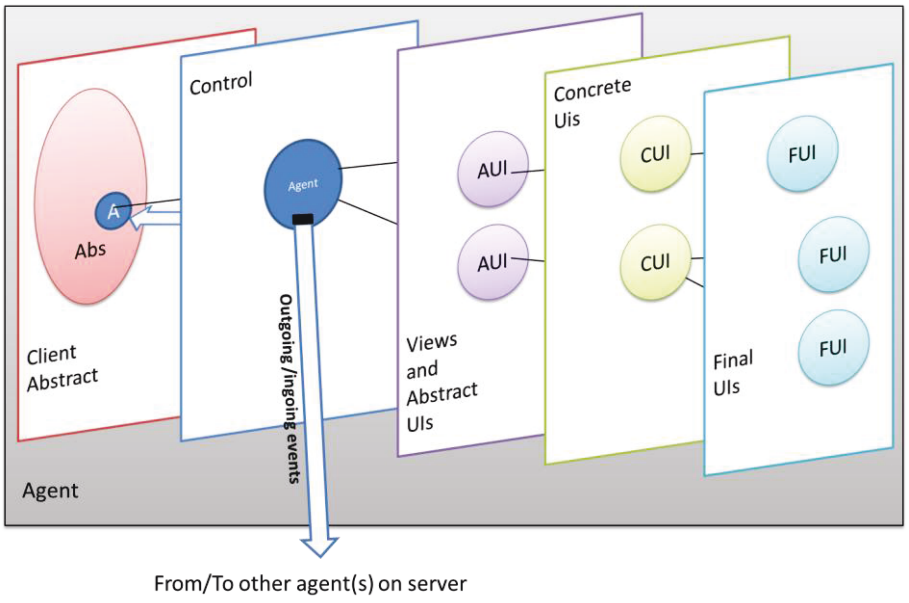


Figure 43. Structure de l'agent dans l'architecture

Tout élément relevant du modèle comportemental et fonctionnel est stocké dans la partie abstraite de l'agent (propriétés, type d'objet). Tout élément relevant de l'interface utilisateur est stocké dans la partie droite, selon la couche à laquelle elle correspond :

- Abstract UI : instance de « présentation » (au sens PAC) correspondant à la tâche en cours
- Concrete UI : couche explicitant la nature concrète de l'interface utilisateur (une image, un dessin vectoriel, etc.)
- Final UI : couche représentant l'interface finale présentée à l'utilisateur, et captant les interactions.

La facette contrôle est responsable de la communication entre la facette Abstract et les facettes de présentation, ainsi que les communications avec les autres agents ou le noyau fonctionnel de l'application.

5.2.2.3 Topologie du monde

Nous abordons ici la notion d'espace topologique du monde, soit l'agencement entre ces zones et les droits des acteurs sur celles-ci. Nous structurons le monde en zones et en relations entre ces zones, le tout formant un espace topologique discret (θ) représentable sous forme de graphe.
La relation entre deux zones définit la possibilité de mouvement d'objet entre ces zones (utilisation de l'outil « déplacement »), indépendamment de la distribution physique ultérieure de celles-ci, et peut être:

- Unilatérale : les objets ne peut se déplacer que dans un sens (flèche unidirectionnelle)
- Bilatérale : les objets peuvent se déplacer dans les deux sens (flèche bidirectionnelle)

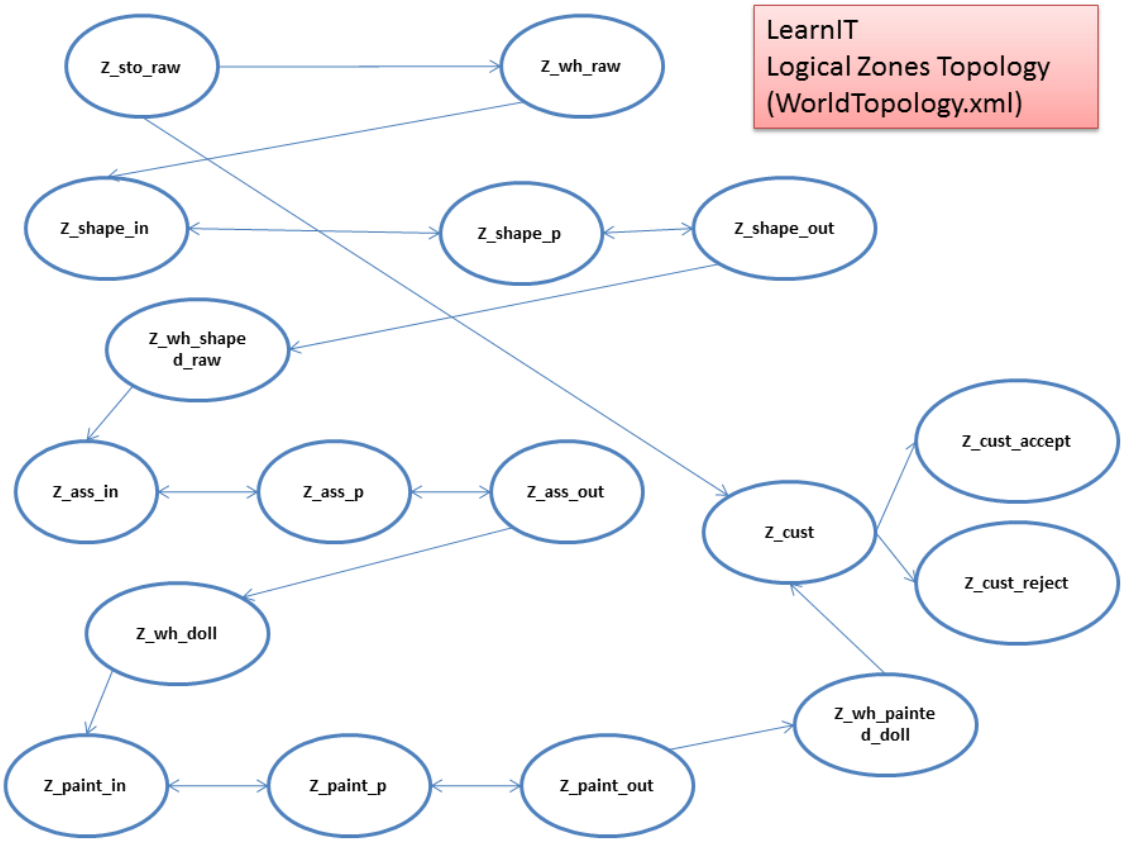


Figure 44. Topologie des zones logiques dans Lea(r)nIT sous forme de graphe. Chaque flèche indique une relation entre les zones.

La topologie est stockée sur le serveur dans un fichier WorldTopology.xml

5.2.2.4 Distribution des zones logiques et numériques

Il nous faut distribuer la structure du monde établie sur les dispositifs. Pour assurer la flexibilité de la répartition (pouvant à terme être dynamique) nous introduisons la notion de zone numérique, intervenant entre zones physiques et zones logiques. Une zone numérique représente à un instant ou dans un contexte donné une ou plusieurs zones logique(s) sur un ou plusieurs dispositif(s). Un dispositif est associé à une ou plusieurs zone(s) numérique(s), elle(s)-même(s) associée(s) à un ensemble de zones logiques. Chaque rôle a accès à un ensemble de dispositifs. Cette logique de zones numérique permet notamment d'exprimer sous forme de graphe (figure 45) la distribution des différentes zones, dans le but d'être ensuite modifiable facilement à l'aide d'un éditeur (5.3).

La figure 45 illustre comment sont construits les chemins possibles entre rôles (et donc, acteur puis dispositif puis zone numérique) et les zones logiques.

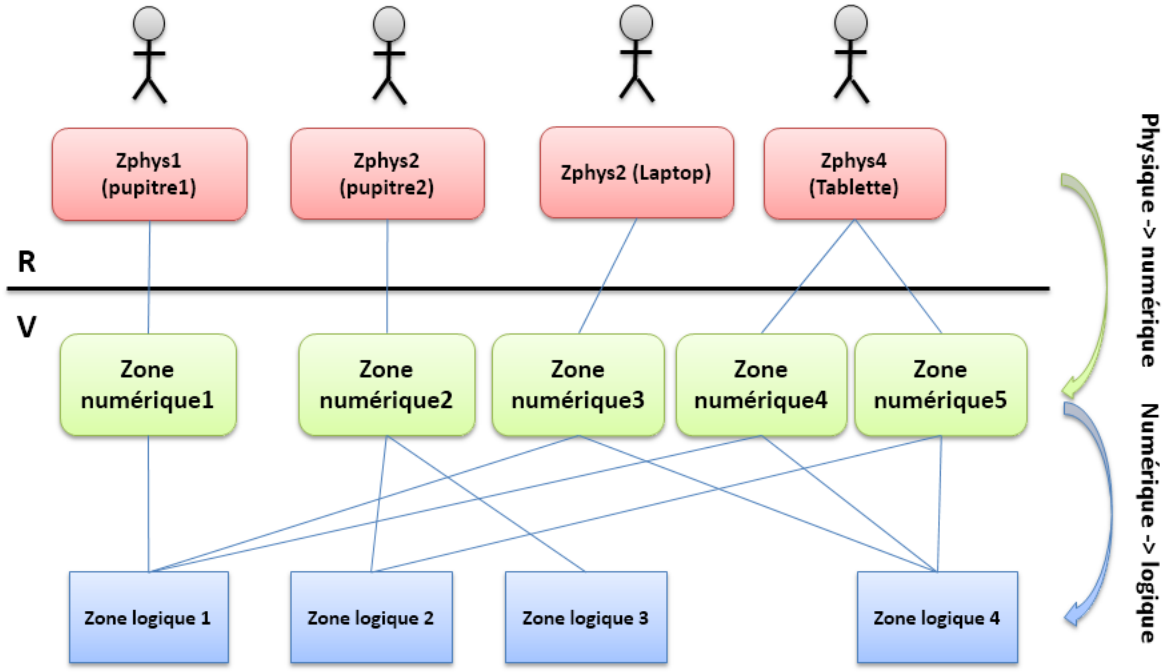


Figure 45. Distribution des zones logiques sur dispositifs par le biais de zones numériques.

De cette manière, une zone logique (et les objets logiques contenus dans ces zones) peut être répartie sur plusieurs dispositifs. Dans la figure 45, la zone logique 1 est déployée à la fois sur le dispositif 1 via la zone numérique 1, mais également sur la tablette via la zone numérique 4 : la zone logique est alors partagée. Des zones logiques peuvent également avoir des représentations différentes sur deux dispositifs différents. Toujours selon l'exemple de la figure 45 les zones numériques 3 et 4 montrent les mêmes zones logiques, mais leur disposition est différente selon que l'on interagisse avec sur un Laptop (ZN 3) ou sur une tablette (ZN 4), aux résolutions et interactions possibles différentes. Enfin, lorsqu'un dispositif peut en afficher plusieurs, seule une zone numérique n'est active à un instant donné. Dans ce cas, un mécanisme de contextualisation (5.2.2.5) est utilisé pour déterminer la zone numérique à afficher.

5.2.2.5 Contextualisation physique

Lorsqu'un dispositif a accès à plusieurs zones numériques (cas du manutentionnaire dans Lea(r)nIt, lequel est mobile) un mécanisme de contextualisation physique permet d'afficher la zone numérique appropriée selon la localisation physique. Ce mécanisme permet de s'assurer que la personne et son dispositif sont physiquement proches d'une zone physique donnée. L'opération de contextualisation consiste à afficher la zone numérique attachée à une zone physique quand le dispositif hôte s'en approche.

Pour rendre possible cette fonctionnalité, nous avons opté pour une solution manuelle basée sur l'utilisation de caméra et d'accéléromètre embarqués et de QR codes imprimés sur papiers et disséminés dans la pièce (ou bien directement affichés sur d'autres zones numériques). Chaque QR code contient un texte unique portant le nom de la zone physique : lorsque l'utilisateur scanne le QR code, l'application lui demande de poser sa tablette. Une fois la tablette posée, le client affiche alors la zone numérique correspondante (4.2.3). L'acteur peut alors travailler dans sa zone numérique. Dès que la tablette est reprise en main, l'accéléromètre déclenche automatiquement la décontextualisation, affichant alors la zone numérique par défaut du rôle.

Le système aurait pu également être amélioré en utilisant un système basé sur les puces RFID, déclenchant automatiquement l'opération de contextualisation dès que le dispositif mobile s'approche ou s'éloigne de la puce.

Figure 46. Lors des expérimentations Lea(r)nIt, la tablette est contextualisée avec le magasin (table surface). La tablette montre alors les objets que la personne « à table » dépose dans la zone logique commune. Les modalités d'affichage différentes d'une même zone logique, sur la tablette et sur la table surface, sont rendues possibles par le concept de zone numérique.

5.2.3 Interface utilisateur : Caméléon

Caméléon est un cadre de travail créé par [Calvary, Coutaz & Thevenin, 2003] visant à améliorer la plasticité des interfaces utilisateur et propose des méthodes et environnements permettant de supporter des applications sensibles au contexte d'utilisation. Caméléon propose d'utiliser une architecture en couches, de la plus abstraite (abstract UI) à la plus concrète (final UI).

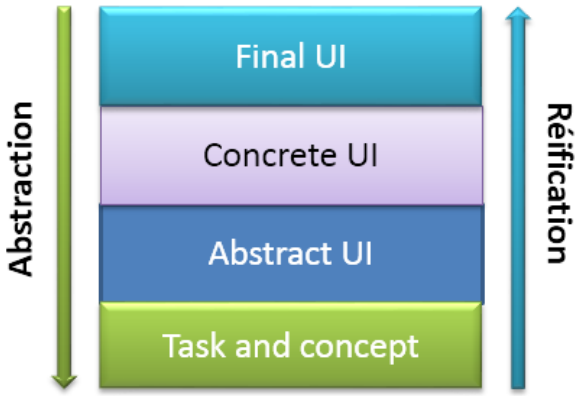


Figure 47. Couches Caméléon

- Tâche et concepts : provient du modèle des tâches et des concepts. La tâche peut par exemple être « renseigner la date de naissance », le concept pouvant dans ce cas être l'information « date de naissance »
- Interface utilisateur abstraite : est une expression canonique des concepts et tâches du domaine indépendamment des interactions disponibles
- Interface utilisateur concrète : transforme l'interface utilisateur abstraite en une expression dépendante des interactions. Pour reprendre notre exemple, à ce niveau existe le *champ* éditable « date de naissance »
- Interface utilisateur finale : générée à partir de l'interface concrète, elle est l'expression finale de l'interface telle que perçue par l'utilisateur. L'interface sera ici différente selon que le champ est à afficher sur un écran 22 pouces d'ordinateur personnel, ou sur un écran 7 pouces d'un smartphone.

Grâce à Caméléon, on peut à partir d'un arbre des tâches et des concepts extraire le code directement exploitable d'une interface utilisateur. Ces concepts ont été repris dans la couche haute « Interface utilisateur » de notre moteur.

5.3 eIRVO+ comme outil de déploiement des agents numériques

La distribution des zones numériques exposée en 5.2.2.4 permet d'être exprimé sous forme de graphes, et de fichiers XML. Nous avons conçu un éditeur WYSIWYG basé sur Microsoft Visio et sur les gabarits IRVO permettant de décrire de façon visuelle cette distribution de zones physiques, numériques, et logiques. Une fois les zones définies, l'utilisateur peut ainsi générer les fichiers de description de déploiement directement utilisables par le serveur SEGAREM grâce à un bouton appelant une macro VBA intégrée dans le modèle Visio.

La figure 48 montre le résultat graphique d'un déploiement possible, et les figures 49 et 50 montrent les fichiers XML résultants, où l'on peut voir les liens ainsi créés.

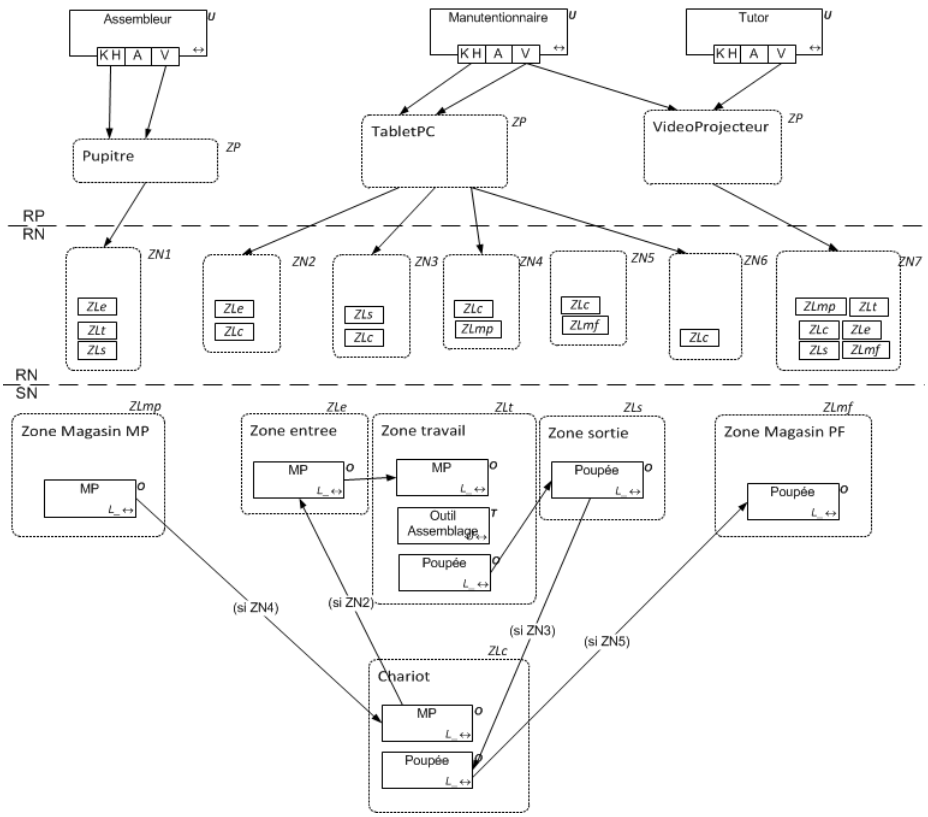


Figure 48. Les possibilités de distribution des zones numériques et logiques de Lea(r)nIT dans eIRVO+

La partie CDATA de la figure 50, décrivant les attributs de FUI des zones logiques, a été ajouté à posteriori, mais une extension future de eIRVO+ serait capable de les y inclure. Alors que la version actuelle du serveur nécessite d'être redémarrée pour prendre en compte la description du déploiement, nous pouvons à terme ainsi distribuer les zones logiques et numériques de façon dynamique.

```
<physicalZone>
    <name>Shaper_wo</name>
    <numericalZone physicalContext="default">ZNshape_wo</numericalZone>
</physicalZone>
<physicalZone>
    <name>Assembler</name>
    <numericalZone physicalContext="default">ZNass</numericalZone>
</physicalZone>
<physicalZone>
    <name>Impregnator</name>
    <numericalZone physicalContext="default">ZNimpr</numericalZone>
</physicalZone>
<physicalZone>
    <name>Warehouseman</name>
    <numericalZone physicalContext="P_shape_in">ZNwh_shape_in</numericalZone>
    <numericalZone physicalContext="P_shape_out">ZNwh_shape_out</numericalZone>
    <numericalZone physicalContext="P_ass_in">ZNwh_ass_in</numericalZone>
    <numericalZone physicalContext="P_ass_out">ZNwh_ass_out</numericalZone>
    <numericalZone physicalContext="P_impr_in">ZNwh_impr_in</numericalZone>
    <numericalZone physicalContext="P_impr_out">ZNwh_impr_out</numericalZone>
    <numericalZone physicalContext="P_sto_mp">ZNwh_exchange</numericalZone>
    <numericalZone physicalContext="P_quality">ZNwh_quality</numericalZone>
    <numericalZone physicalContext="default">ZNwh_alone</numericalZone>
</physicalZone>
<physicalZone>
    <name>Chuck</name>
```

Figure 49. Le passage du dispositif à la zone numérique et la contextualisation sont exprimés dans un fichier physical2Numerical.xml

```
<numericalZonesToLogicalZones>
    <numericalZone>
        <name>ZNass</name>
        <finalUI><![CDATA[
            <pxWidth>1024</pxWidth>
            <pxHeight>768</pxHeight>
            <bgAssetFile>bg_atelier_2.png</bgAssetFile>
        ]]></finalUI>
        <logicalZone>
            <name>Z_ass_in</name>
            <finalUI><![CDATA[
                <pcX>0</pcX>
                <pcY>0</pcY>
                <pcWidth>24</pcWidth>
                <pcHeight>66</pcHeight>
                <bgAssetFile></bgAssetFile>
                <qrCodeFile>P_ass_in.png</qrCodeFile>
                <qrCodePosition>TOP_LEFT</qrCodePosition>
                <properties>
                    <property>
                        <name>itemsDisplayMode</name>
                        <type>String</type>
                        <value>loose</value>
                    </property>
                </properties>
            ]]></finalUI>
        </logicalZone>
        <logicalZone>
            <name>Z_ass_process</name>
            <finalUI><![CDATA[
                <pcX>24</pcX>
                <pcY>0</pcY>
```

Figure 50. Le passage de la zone numérique à la zone logique, est exprimé dans un fichier Numerical2Logical.xml.

5.4 Architecture technique

Afin de garantir l'intégrité du noyau fonctionnel un serveur centralise les données du monde numérique et communique par le biais d'évènements avec les applications clientes. Hormis sur la table PixelSense (pour le magasin) et sur l'IPhone (pour le Blade Runner qui teste les réplicants), le client utilisé est le même pour tous les autres dispositifs, seul le rôle change, donnant alors accès aux zones numériques correspondantes aux tâches que le rôle est sensé effectuer.

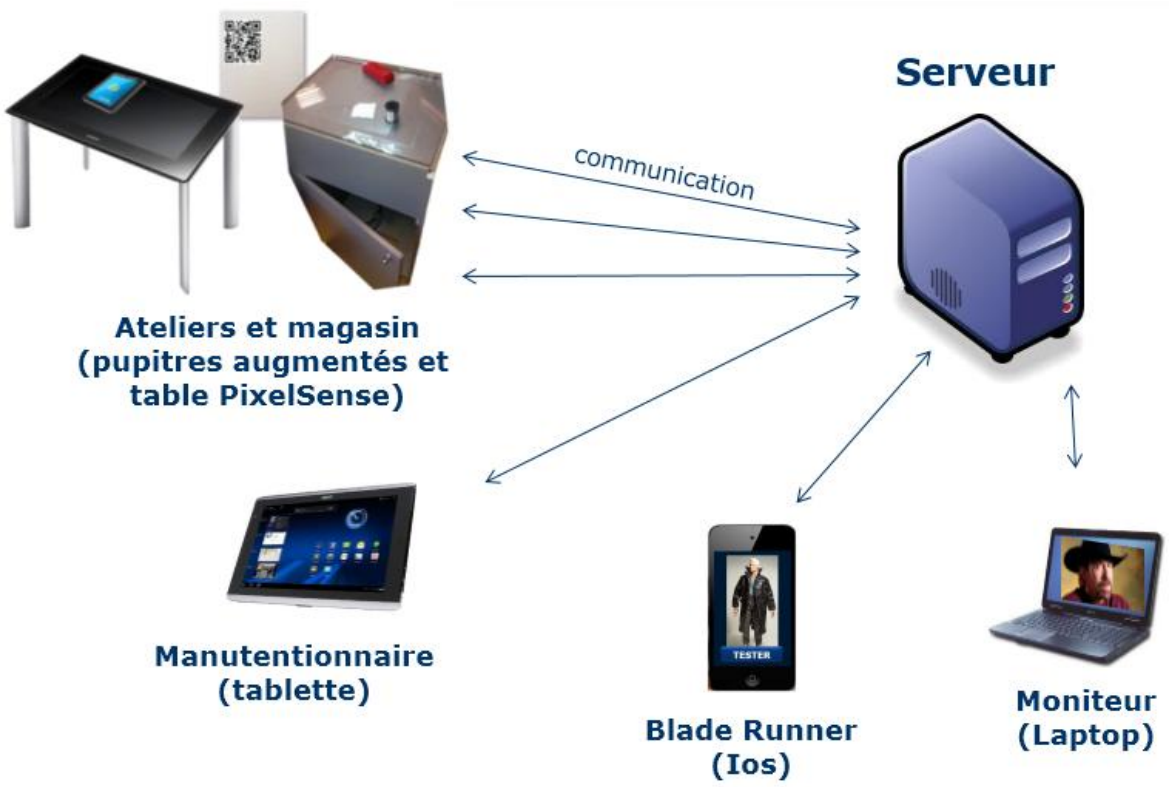


Figure 51. Architecture générale de Lea(r)nIt

5.4.1 Architecture serveur

Le serveur s'exécute sous le logiciel Electroserver sous forme d'un plugin Java. Le plugin a été développé par la société Symetrix. Les principaux rôles du serveur sont :

- d'assurer la communication entre les clients
- de coordonner les clients
- de contrôler la cohérence du monde
- de stocker les données

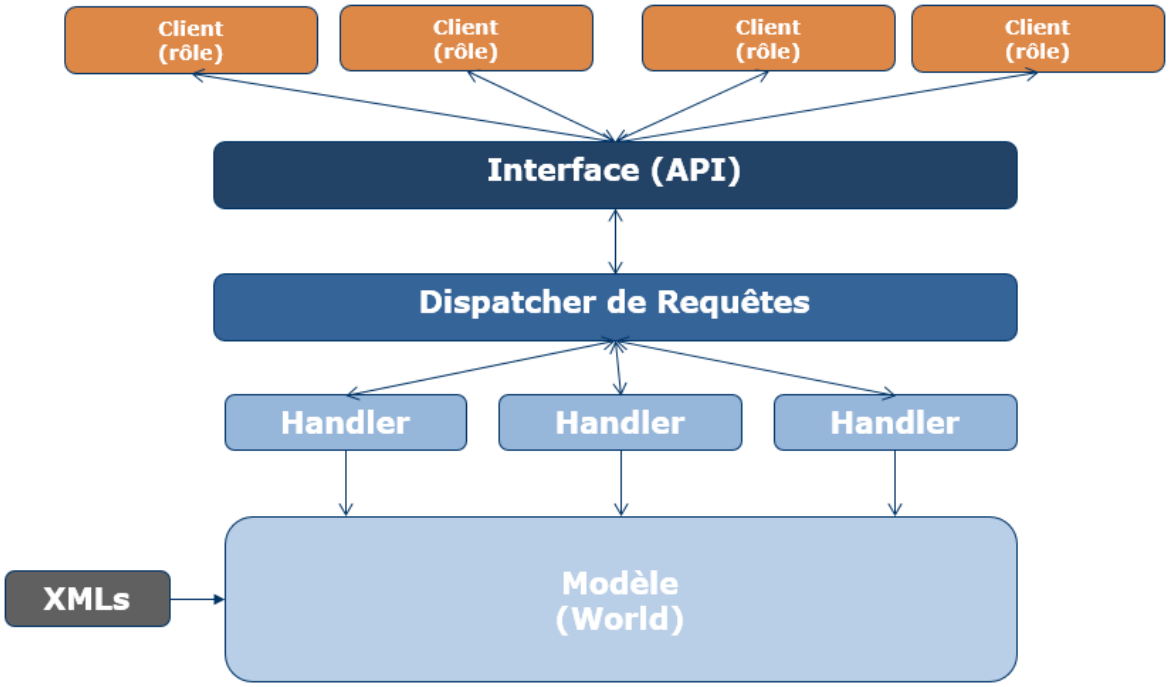


Figure 52. Architecture du serveur SEGAREM

Dès que le serveur reçoit une commande du client, par exemple « créer une pièce » ou bien « déplacer cette pièce dans cette zone », s'opère la séquence suivante :

- Réception du message par le serveur
- Dispatch de la requête vers le bon handler
- Vérification du rôle
- Traitement de la requête
    - (Modification du modèle)
- Notification des clients

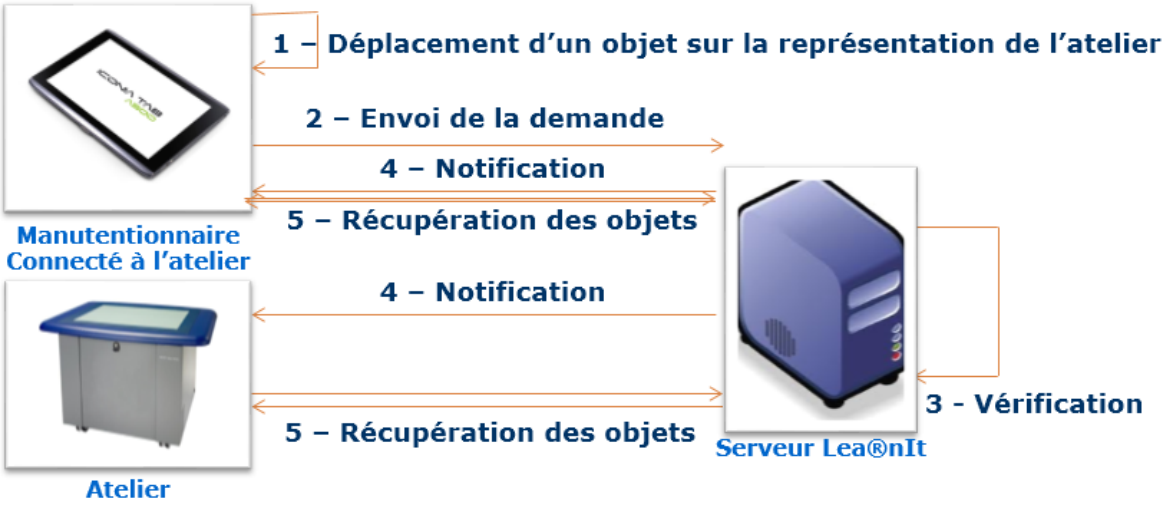


Figure 53. Echanges de messages entre clients et serveur lors d'un dépôt de pièce par le manutentionnaire dans une zone d'entrée d'un atelier.

5.4.2 Architecture client

L'application cliente est elle-même structurée comme un agent dont la facette présentation est constituée de l'ensemble des zones numériques avec lesquelles le rôle attaché a le droit d'interagir, n'affichant alors qu'une zone numérique active à la fois. A chaque instance du client est affecté un rôle. Le client est écrit en ActionScript et déployé avec AIR.

La partie « application control » est responsable de la communication avec le serveur et les autres instances client.

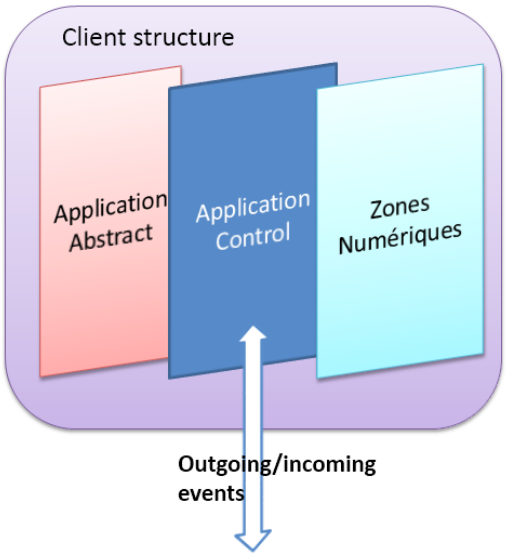


Figure 54. Structure d'une instance cliente

5.4.3 Protocole de communication

Le serveur et le client utilisent la couche de communication fournie par Electroserver, basé sur des messages. Les messages sont encapsulés dans un 'ESObject', un format de message créé par ElectroTank dans le but de standardiser les échanges existants entre des applications clientes et les plugins côté serveur.

Un « ESObject » supporte de nombreux types de données tels que 'booléens', 'entiers', 'chaînes de caractère', etc. et peut encapsuler d'autres ESObject ainsi que des tableaux de données.

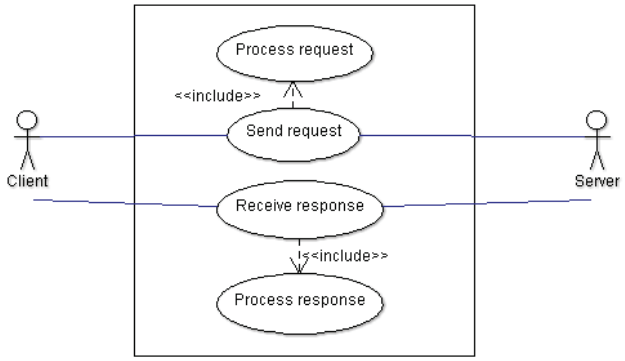


Figure 55. Cas d'utilisation d'échanges de requêtes et de réponses entre client et serveur

Le protocole défini entre Symetrix qui a écrit la partie serveur et le laboratoire qui a écrit la partie cliente est basé sur un ensemble de commandes simples, de requêtes et de réponses, dont la structuration des champs est la suivante, que ce soit le serveur qui l'envoie ou un client :

- Cmd : identifie le type de commande parmi :
  - REQUEST_GET_ZONE_CONTENT_BY_ZONE : demande au serveur de retourner la liste des objets contenus dans une zone. Nécessite le nom de la zone logique en paramètre
  - REQUEST_MOVE_CONTENT : demande au serveur de déplacer l'objet (désigné par son ID en paramètre) dans la zone cible donnée en paramètre
  - REQUEST_GET_ZONES_BY_ROLE : demande au serveur de retourner la liste des zones numériques en fonction du rôle attaché au client
  - REQUEST_USE_TOOL : demande au serveur d'utiliser l'outil donné en paramètre sur les objets cibles identifiés par leurs ID (également donnés en paramètre). La syntaxe d'appel d'outil est explicité en 5.4.4.
  - REQUEST_PAUSE_GAME : demande au serveur de mettre le jeu en pause. Les clients affichent alors un écran de pause
  - REQUEST_RESUME_GAME : demande au serveur de reprendre un jeu mis en pause

    - REQUEST_SAVE_WORLD : demande au serveur de sauvegarder l'état du monde dans un fichier
- Argument : identifie les arguments en fonction des requêtes effectuées :
    - zoneName : nom de zone logique ou numérique
    - id : identifiant d'un objet
    - gameObjects[] : liste de game objects cibles d'une commande

Chaque fois qu'une requête est envoyée au serveur, ce dernier produit une réponse au client qui l'a envoyée, ainsi qu'aux clients pouvant interagir avec la zone logique concernée par la requête. Le serveur indique notamment dans sa réponse si la commande a été couronnée de succès ou non. Le format de la réponse est le suivant :

- cmd : identifie le type de commande qui a été transmis dans la requête (mêmes valeurs que lors de la requête)
- success : booléen qui indique si la commande a été correctement effectuée.
- reason : chaîne de caractère expliquant pourquoi la commande demandée a échoué le cas échéant.
- From : identifie le client qui a demandé la requête
- request : contient l'ESObject de la requête initiale

5.4.4 Outils

Etant les moyens d'actions des utilisateurs sur des objets selon les règles du jeu, et donc similaires à des fonctions, nous pouvons définir la syntaxe d'utilisation d'un outil suivante :

Soit:
- D : l'application cliente/dispositif (unique ID)
- R : le rôle qui utilise l'outil
- Zl : la zone logique où l'action prend lieu
- T : l'outil utilisé
- Arg* : une liste d'arguments
- C : l'objet ou l'ensemble d'objets cible(s), sur lequel (lesquels) l'on agit

Syntaxe d'appel d'un outil T : (D, R, Zl, T, arg*, C)

Arg* peut contenir une certaine logique telle que : « si propriété « active » de la cible est égale à « true » alors assigner la valeur « false » à la propriété « active » de la cible. Alors que le modèle outils n'est basé que sur des structures de contrôle simples, il devrait être possible à terme d'y utiliser un langage de script plus évolué et d'y définir des fonctions mathématiques, dans l'esprit des travaux de Magnusson (3.5.2).

5.4.5 Interactions utilisateurs

Toutes les interactions utilisateurs trouvent leur point de départ dans la FUI de chaque agent à travers le protocole TUIO. La FUI transmet ensuite les informations à la CUI, puis à l'AUI, enfin au contrôle qui décide quoi faire : mettre à jour la partie abstraite de l'agent et éventuellement propager l'évènement aux autres agents. Les évènements TUIO gérés peuvent provenir de deux sources en particulier :

- Tactile, TUIO permet de gérer les évènements TuioTouchEvent suivants:
  - TOUCH_DOWN : quand un doigt est posé sur la surface
  - TOUCH_UP : quand un doigt posé est relevé
  - TOUCH_MOVE : quand un doigt posé se déplace
  - ROLL_OVER : quand un doigt posé se déplace jusqu'à se superposer à une forme
  - ROLL_OUT : quand un doigt posé se déplace jusqu'à sortir d'une forme
  - TOUCH_TAP : un doigt en posé puis rapidement retiré
  - TOUCH_DOUBLE_TAP : deux TOUCH_TAP rapidement exécutés
  - TOUCH_HOLD : quand un doigt est maintenu

- Tangible, les évènements TUIO TuioFiducialEvent suivants étant alors envoyés par D'Fusion (Figure 58) :
  - ADD : quand un objet est posé sur la surface
  - REMOVE : quand un objet posé est enlevé
  - OVER : quand un objet se déplace jusqu'à se superposer à une forme
  - OUT : quand un objet se déplace jusqu'à sortir d'une forme
  - ROTATE : quand un objet subit une rotation

5.4.5.1 Gestion des interfaces tangibles

TUIO est un cadre de développement ouvert qui définit un protocole et des interfaces de programmation destinés aux interfaces multi-touch et tangibles. Il permet ainsi de capter dans le but d'interpréter des évènements utilisateurs.

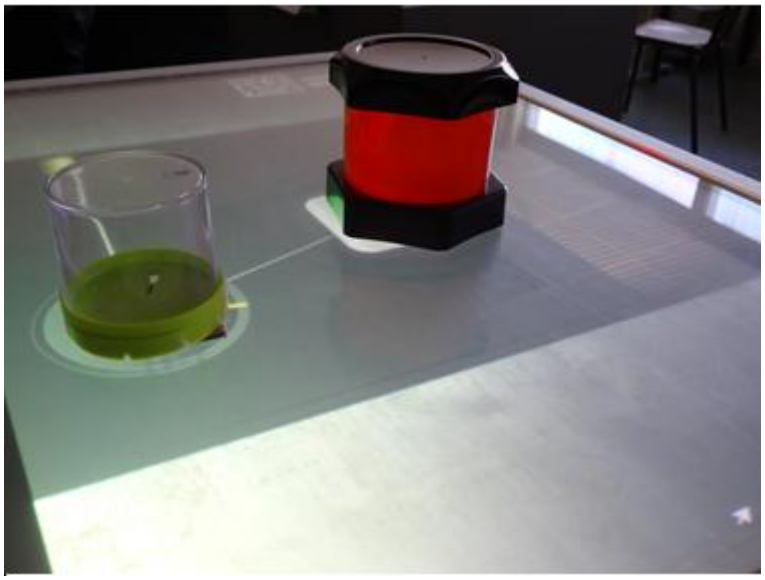


Figure 56. Deux objets tangibles permettant de manipuler les objets numériques

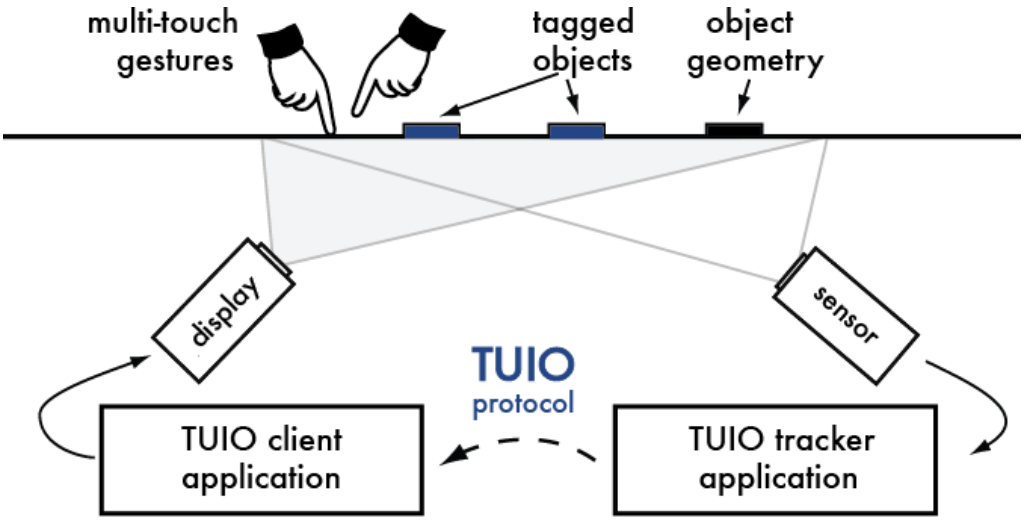


Figure 57. Vue matérielle d'un dispositif utilisant TUIO

Dans le cas spécifique des interfaces tangibles, le plugin D'fusion est utilisé, un langage de communication de la présence des objets ainsi qu'un langage de script LUA interviennent en amont pour produire des évènements TUIO qui seront enfin envoyée à la FUI de l'agent :

Figure 58. Chaîne de protocole pour l'utilisation d'objets tangibles

Nous avons présenté notre architecture. A présent, montrons comment elle a pu répondre aux besoins de Lea(r)nIt, un Serious Game en réalité Mixte issu d'un Serious Game « classique », et ayant fait l'objet de deux expérimentations dans le cadre du projet SEGAREM.

Chapitre 6

Mise en œuvre et évaluation d'un Learning Game

Dans ce chapitre décrivant le cas Lea(r)nIt, nous montrons comment l'architecture a pu répondre aux besoins d'un Serious Game en réalité Mixte issu d'un Serious Game « classique », et ayant fait l'objet de deux expérimentations grandeur nature dans le cadre du projet SEGAREM.

6.1 Le jeu LEAN

6.1.1 Présentation et objectifs pédagogiques

Destiné à une audience d'étudiants ou de managers dans les entreprises, et devant se dérouler en mode présentiel, le jeu LEAN original, ou « Buckingham Lean Game », est un jeu d'entreprise ayant pour objectif l'application des méthodes de Lean management, soit la méthode de gestion de la production recherchant la performance par l'amélioration continue et l'élimination des gaspillages. Les principes fondamentaux du jeu sont la simulation d'une entreprise de production qui doit répondre à une demande client, ainsi que la mise en œuvre d'une démarche d'amélioration, le tout inscrit dans une pédagogie active.

Les objectifs principaux d'un tel jeu sont de :

- Savoir mesurer des informations, à l'aide d'indicateurs de qualité, de temps
- Apprendre à voir les gaspillages, les situations à Non-Valeur Ajoutée
- Construire une vue globale collectivement à partir des points de vue personnels
- Mettre en situation de prise de décision collective
- Illustrer les concepts du Lean Manufacturing
- Favoriser l'apprentissage par la pratique, notamment de réaliser les difficultés crées par des opérations simples
- Expérimenter les conséquences des améliorations, afin de maitriser les concepts en jeu, de savoir expliquer à quelqu'un d'autre (rôle de formation ultérieur), de ressentir l'amélioration pour l'opérateur, et enfin de réaliser que des petites mesures peuvent mener à des changements radicaux
- Dérouler une démarche d'amélioration : utiliser des décisions collectives pour prioriser les améliorations (court terme / long terme), les mettre en application et enfin mettre en avant le côté itératif, et le côté continu de l'amélioration

6.1.2 Déroulement

Le jeu original prend place dans un atelier fictif de production de poupées, utilisant des briques de LEGO pour représenter les matières premières et transformées. Les apprenants y jouent les rôles d'opérateurs et de clients, assemblant, transportant et contrôlant ces matières à des postes définis : magasin pour stocker les « matières premières », poste d'assemblage de premier niveau, puis de deuxième et troisième niveaux (ajout de plus en plus de détails via des briques de LEGO), poste de contrôle qualité où les pièces assemblées sont jugées conformes ou non, et enfin le

client qui doit être servi conformément à sa commande et à temps. Les postes sont physiquement répartis dans la pièce en une disposition initiale (volontairement la moins optimale, voir la figure 61) et une personne joue le rôle de manutentionnaire, personnage central responsable du transport des matières entre les postes.

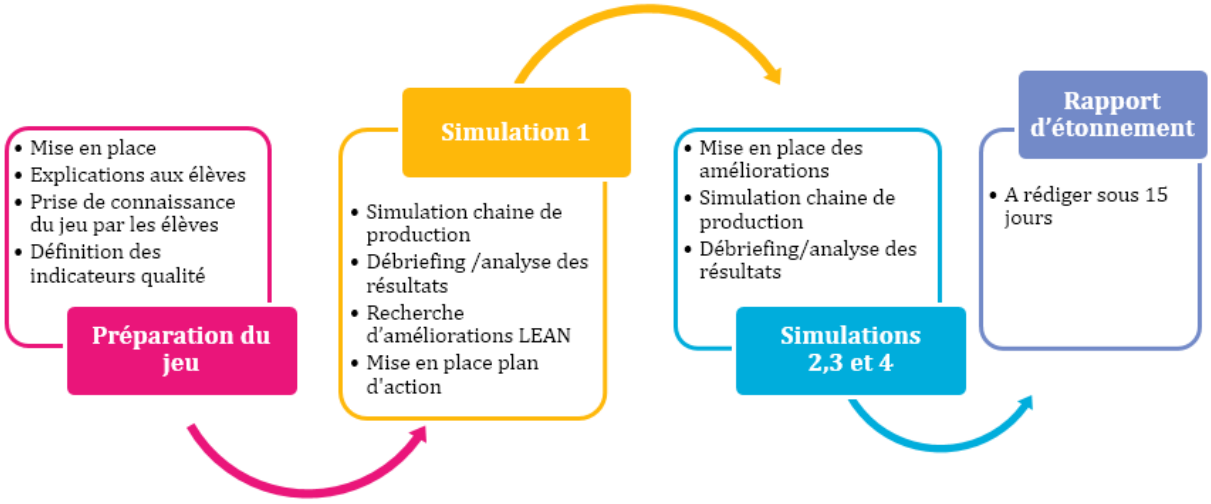


Figure 59. Déroulement du jeu

La figure 59 représente les étapes de déroulement du jeu. Ainsi, après une phase de préparation du jeu, les apprenants vont effectuer une première simulation durant 15 minutes lors de laquelle ils doivent produire et livrer au mieux. A l'issue de cette simulation, une phase de débriefing et d'analyse de la chaîne sera conduite par le formateur pour les mener à une recherche d'améliorations LEAN et un plan d'action. Ces simulations et recherche d'améliorations seront réitérées plusieurs fois afin d'obtenir à la fin une chaîne de production optimale, en termes de :

- Temps de production d'une pièce, de toutes les pièces
- Qualité du produit
- Flux physique des matières au sein de l'atelier
- Respect des temps de livraison
- Etc…

Les améliorations mises en place entre les itérations peuvent être de différentes natures : tri des pièces de LEGO à l'entrée des postes, déplacement physique d'un poste pour réduire les trajets du manutentionnaire, voire de plusieurs postes (pouvant potentiellement mener à terme à la suppression totale du manutentionnaire), de répartition du travail d'assemblage, etc…

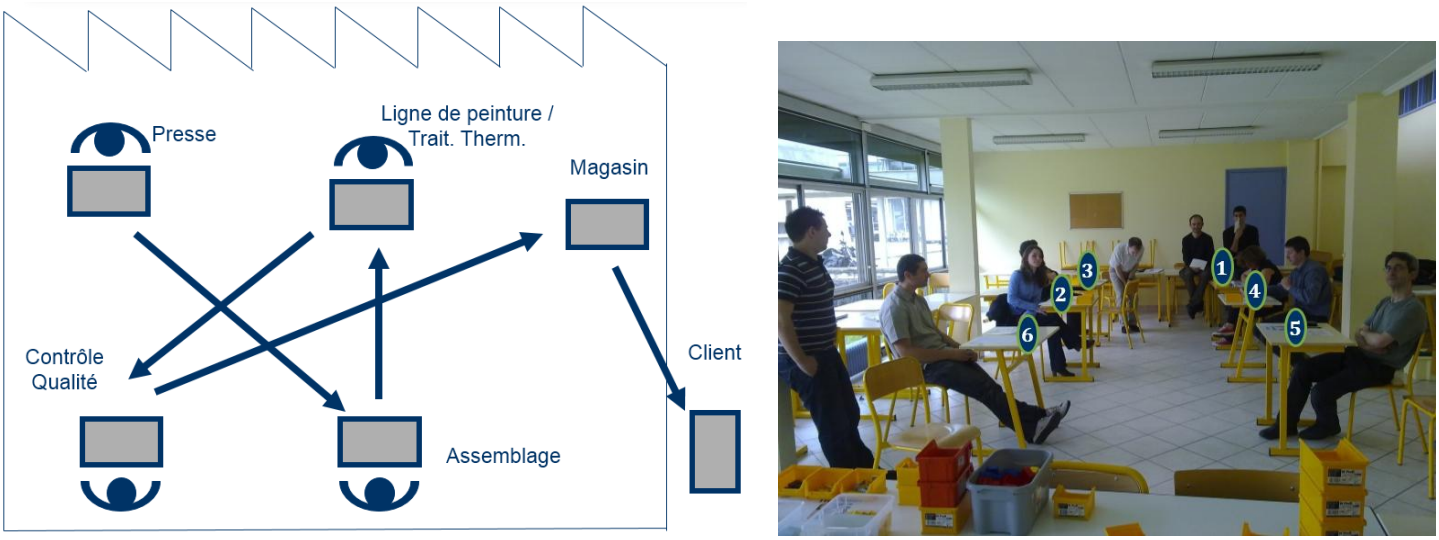


Figure 61. Disposition physique initiale de l'atelier

Figure 60. Une disposition physique après une amélioration possible dans laquelle tous les postes sont alignés, ici le manutentionnaire n'intervient plus qu'en bout de chaîne (et semble s'ennuyer un peu, n'ayant plus beaucoup de « travail » à faire)

6.1.3 Limites du jeu original

Malgré un succès certain du point de vue de la pédagogie, le jeu dans cette version originelle souffre de plusieurs lacunes. D'une part les activités sont décontextualisées : les apprenants doivent imaginer que les LEGO sont de vraies pièces, ce qui requière un certain niveau d'abstraction. De plus la représentation des contraintes matérielles de l'usine est affaiblie : impossible de déplacer une machine telle que la presse alors qu'il ne s'agit que d'une table à déplacer dans le cadre de la simulation; Les contraintes de coût sont également mal prises en compte. D'autre part, le dispositif souffre d'une certaine rigidité : il est impossible de modifier les pièces (couleurs, formes, etc…) et difficile d'introduire des défauts qualités (on peut difficilement casser des pièces de LEGO) ainsi que des pannes machines. Il n'est pas possible d'utiliser d'outils ni d'opérations complexes. Le stock de pièces est à priori limité : en fin d'itération les apprenants sont obligés de démonter des pièces produites pour créer de nouvelles matières premières. Enfin, le formateur n'a pas d'indicateurs « consolidés » pendant le déroulement de l'itération.

Porter le jeu sur un support informatisé classique (un clavier/écran/souris par personne) ajouterait une plus grande souplesse dans le contenu du jeu, ainsi qu'une plus grande puissance statistique, mais introduirait également une distance trop grande entre la situation d'apprentissage et la situation réelle. Cette dernière induisant notamment une collaboration étroite telle que l'on peut en trouver au sein d'un atelier ainsi que des gestes techniques difficiles à transposer dans des conditions purement virtuelles.

Le projet SEGAREM tente de pallier à ces problèmes en proposant d'utiliser un environnement de réalité mixte dans le jeu LEAN.

6.2 Lea(r)nIt : un Serious Game en Réalité Mixte

Un premier objectif de l'utilisation de la réalité mixte dans le Serious Game est d'accroître l'immersion des apprenants, notamment en réduisant la distance qui existe entre la situation d'apprentissage et la situation réelle. La réalité mixte rend également possible de représenter certaines contraintes associées à l'exécution de gestes techniques. Enfin, il s'agît de faciliter l'orchestration pédagogique tout en dynamisant la démarche d'amélioration, en rendant mieux compte des contraintes de coûts, temps et performances des configurations envisagées lors des débriefings ainsi que pendant l'exécution, bénéficiant alors de la puissance informatique capable d'exploiter les traces automatiquement générées par les tâches des apprenants.

6.2.1 Univers fictionnel du jeu

En cohérence avec les contraintes du jeu initial, un nouveau scénario et univers de jeu ont été imaginés pour illustrer ces diverses modalités d'interaction ainsi que pour favoriser l'immersion des apprenants. Ils sont inspirés du film « Blade Runner » de Ridley Scott, lui-même adapté du livre de Philip K. Dick « Do Androids Dream of Electric Sheep? ».

« Nous sommes en 2032, la fin de l'espèce humaine est annoncée : les hommes ne peuvent plus se reproduire après qu'une série de catastrophes nucléaires a rendu stérile la totalité de l'humanité. Les maternités ne sont plus que ruines, les écoles rapidement abandonnées, les gouvernements n'ont plus de prises sur les populations, la guérilla urbaine s'est installée durablement…
Dans ce contexte de fin du monde, Eldon Tyrell, génial cyber-généticien, redonne une lueur d'espoir : il parvient à créer des androïdes en tout point semblables aux êtres humains... les

réplicants. Le procédé industriel utilisé crée néanmoins des défauts parmi ces réplicants : certains sont en effet dénués de sensibilité et d'émotion et nécessitent d'être retirés avant d'être libérés.

Saurez-vous améliorer le processus de production de l'usine Tyrell et ainsi permettre de produire un maximum de réplicants avant que n'advienne l'extinction de l'humanité?! »

Voici donc exposé le thème fictionnel (ou « Background ») du jeu, le but consistant alors à « repeupler l'humanité » via la production en masse de « réplicants » en améliorant le processus de production de l'usine Tyrell par des méthodes d'amélioration continue.

6.2.2 De nouvelles modalités d'interaction

Pour parvenir à introduire la réalité mixte dans le jeu, des supports numériques et interactifs autorisant la reconnaissance d'objets tangibles sont utilisés pour les différents postes. Les postes existants sont modifiés pour utiliser la réalité mixte et deux nouveaux postes font leur apparition, difficiles à implémenter dans la version originale :

- Poste d' « imprégnation » : effectuant une opération liée à l'univers de jeu, l' « imprégnation de personnalité », et mettant l'accent sur l'utilisation d'objets tangibles pour manipuler le système.
- Poste « moniteur » sur lequel le formateur peut observer en temps réel les statistiques ainsi qu'en extraire certaines pour appuyer la partie débriefing.

Diverses modalités d'interaction sont prises en compte : tactile, objets tangibles, tracking, ainsi qu'une grande diversité de dispositifs : pupitres, tablette, PC, smartphones, dont les détails sont décrits en 6.5.

6.2.3 De nouvelles « erreurs de fabrication » possibles

Alors que le scénario originel ne permettait que d'introduire que des défauts d'assemblage de LEGO, la version mixte de LEAN permet de nouvelles façons de « mal produire », renforçant l'aspect ludique du dispositif et l'attention de chaque opérateur. Ainsi à chaque étape de fabrication, plusieurs types de défauts peuvent être introduits dans le produit, défauts qui seront répercutés sur le produit final, influençant le déroulement de la phase de test qualité avant livraison. De la même façon les postes, s'ils sont « mal utilisés », peuvent désormais être « en panne », empêchant d'utiliser les outils disponibles, voire introduisant des défauts imperceptibles à l'œil nu dans le produit. Les défauts pouvant être présents dans le produit et pannes de machine, sont expliqués dans la section de chaque poste.

6.3 Déroulement de Lea(r)nIt

Cette section présente le déroulement du jeu Lea(r)nIt.

6.3.1 Modélisation Orchestra du jeu

Nous avons modélisé le déroulement du jeu Lea(r)nIt selon le formalisme Orchestra (4.1.3.1).

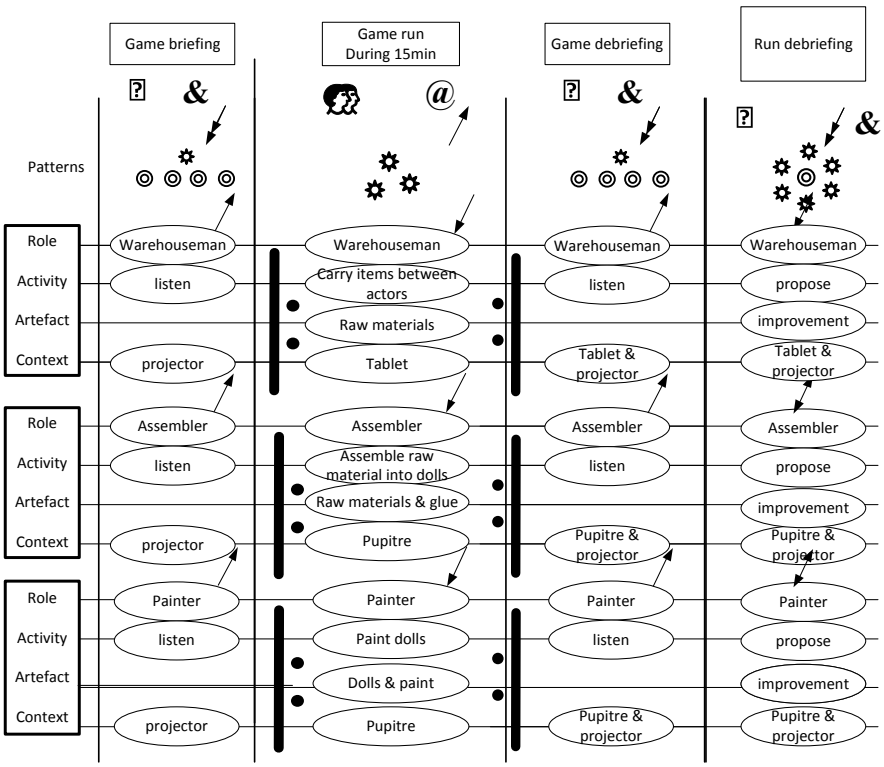


Figure 62. Modélisation Orchestra du jeu

6.3.2 Les différents rôles

Rôle	Poste	Tâches	Modalités d'interaction	Entrée	Sortie
Tuteur	Moniteur (PC)	Responsable de la pédagogie et anime le déroulement du jeu	Souris, clavier, écran et vidéoprojecteur		
Operateur de presse	Presse (pupitre)	Actionne la presse pour produire les corps des réplicants	Tactile	Sacs de granulats	Corps de réplicants
Operateur d'assemblage	Assemblage (pupitre)	Pose des points de colle à l'aide d'un pistolet à colle	Tactile, interface tangible	Corps et membres de	Réplicants assemblés

		tangible puis assemble les membres sur les corps selon un standard de travail.	(pistolet à colle)	réplicants, réserve de colle	
Operateur d'imprégnation	Imprégnation (pupitre)	Imprègne le corps assemblé d'une personnalité à l'aide d'une technologie révolutionnaire !	Interfaces tangibles (potentiomètre, Catalyseur de rayon)	Réplicants assemblés	Réplicants imprégnés (produit fini)
Testeur qualité	Contrôle qualité (Table surface et smartphone)	Contrôle que les réplicants produits ne comportent pas de défauts à l'aide du test de Voigt-Turing.	Tactile	Réplicants imprégnés	Réplicants imprégnés
Client	??	Effectue et réceptionne des commandes de réplicants			
Manu-tentionnaire	Chariot manutentionnaire (Tablette)	Achemine les matières premières et transformées d'un poste à l'autre	Tactile, contextualisation physique	matières	matières

Figure 63. Les différents rôles impliqués, et une courte description textuelle de leurs scénarii

6.3.3 Description des postes

Les postes d'opérateurs disponibles utilisés sur les pupitres suivent une structure commune et comportent:

- **Une zone d'entrée** : accueille les matières premières ou transformées nécessaire au travail à effectuer
- **Une zone de travail** : là où s'effectue le travail à proprement parler (pressage, assemblage, imprégnation)
- **Une zone de sortie** : où sont déposées les matières transformées
- **Une zone de contrôle** : où diverses opérations spécifiques au poste peuvent être effectuées
- **Des QR codes :** utilisés par le manutentionnaire pour contextualiser sa tablette

Chaque poste d'opérateur a une zone numérique correspondante. Le manutentionnaire a lui accès à des zones numériques spécifiques, lui permettant de prendre ou déposer des objets dans les zones logiques d'entrée et de sortie des différents postes. Un QR code affiché sur la zone en question permet au manutentionnaire de contextualiser son chariot (voir 5.2.2.5) avec la zone concernée.

6.3.3.1 Poste de Presse

Utilisant un pupitre interactif (6.5.1), le poste de presse accueille un utilisateur responsable de la production de « corps » de réplicant, issus d'une presse à injection utilisant des granulats en entrée et étant la base du réplicant. L'opérateur de presse doit d'abord sélectionner le moule adéquat pour produire la forme souhaitée (la forme prescrite est affichée dans la zone d' « ordres de fabrication »). Une fois le moule en place, il actionne la presse en maintenant ses doigts sur les boutons rouge jusqu'à ce que la production de la forme soit complète. Le corps créé apparaît alors dans la zone de sortie.

Pour actionner la presse il faut laisser les deux boutons rouges « appuyés » simultanément (voir figure 64). Le joueur peut recharger la jauge de granulats disponibles affichée au centre en glissant (par drag'n drop) la recharge disponible sur la jauge. Les corps créés apparaissent dans la « zone de sortie » située sur la droite. L'utilisateur doit gérer la quantité de granulats disponibles sous peine de déclencher une panne machine (lorsqu'il n'y a plus de granulats dans la machine) ou d'introduire des défauts dans le corps produit (s'il reste des granulats dans la machine mais pas assez pour remplir complètement le moule). Lorsque la machine est en panne, la presse ne peut être utilisée pendant une minute, pénalisant alors toute la chaîne de production.

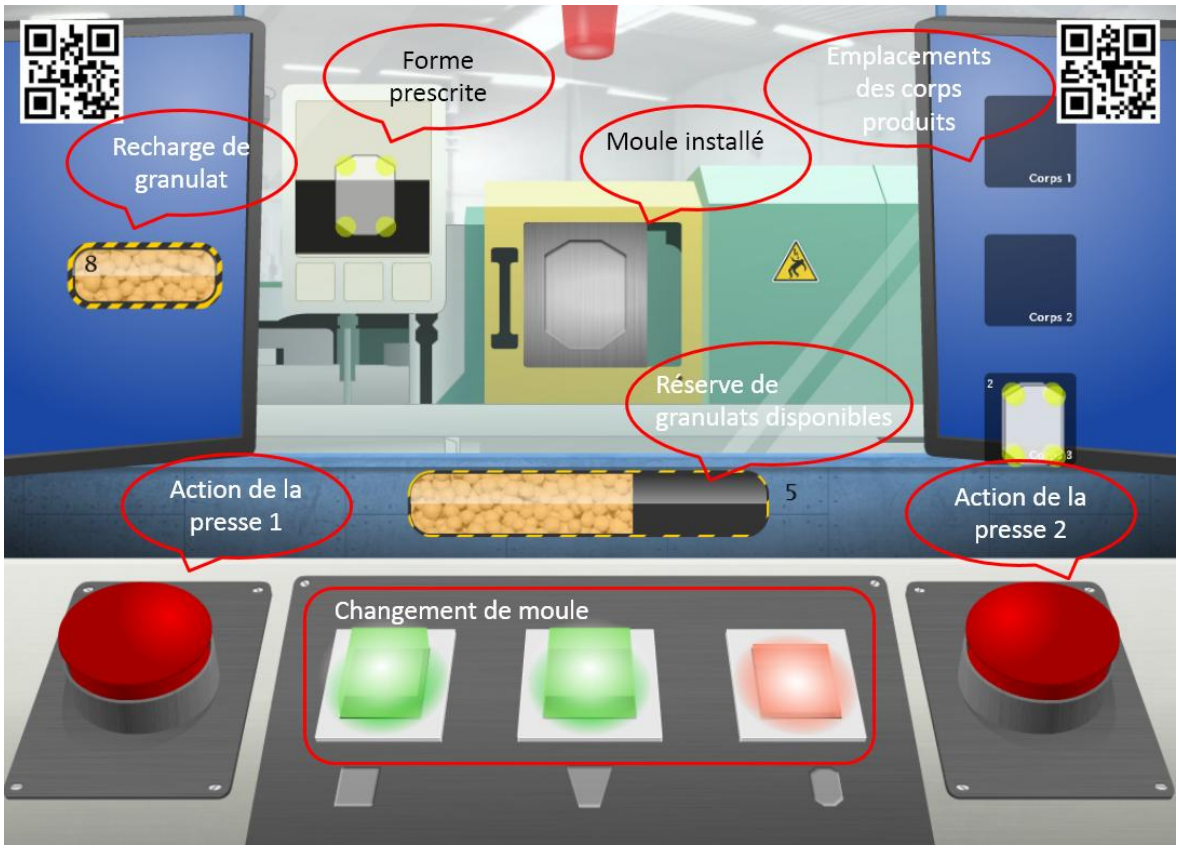


Figure 64. Ecran du poste de presse tel qu'affiché sur le pupitre (les bulles ne sont pas affichées à l'exécution).

Figure 65. L'opérateur a provoqué une panne de l'outil en ayant essayé de produire une pièce après avoir épuisé ses réserves de granulats chargées dans la presse, rendant le poste inutilisable pendant une minute

6.3.3.2 Poste d'assemblage

Sur un pupitre interactif, l'opérateur d'assemblage a pour rôle de coller et d'assembler les corps (produits par la presse) et les membres (matières premières stockées dans le magasin) afin de créer le corps complet d'un réplicant. Pour cela il est muni d'un vrai pistolet à colle augmenté d'une LED qui peut être actionnée par un interrupteur placé sous la gâchette du pistolet et alimentée par une pile logée dans le corps du pistolet. La figure 24 est une photo en situation réelle de l'opérateur d'assemblage lors des expérimentations est montrée.

Figure 66. Le pistolet à colle augmenté (en haut), le mécanisme interrupteur (en bas à gauche), la LED « cachée » dans l'embout de collage (en bas à droite).

L'operateur d'assemblage doit poser sur son espace de travail (au centre) les pièces nécessaires à l'assemblage d'un type de réplicant donné, dont l'agencement est montré par une figure d'exemple sur un support papier, selon le type de réplicant à produire.

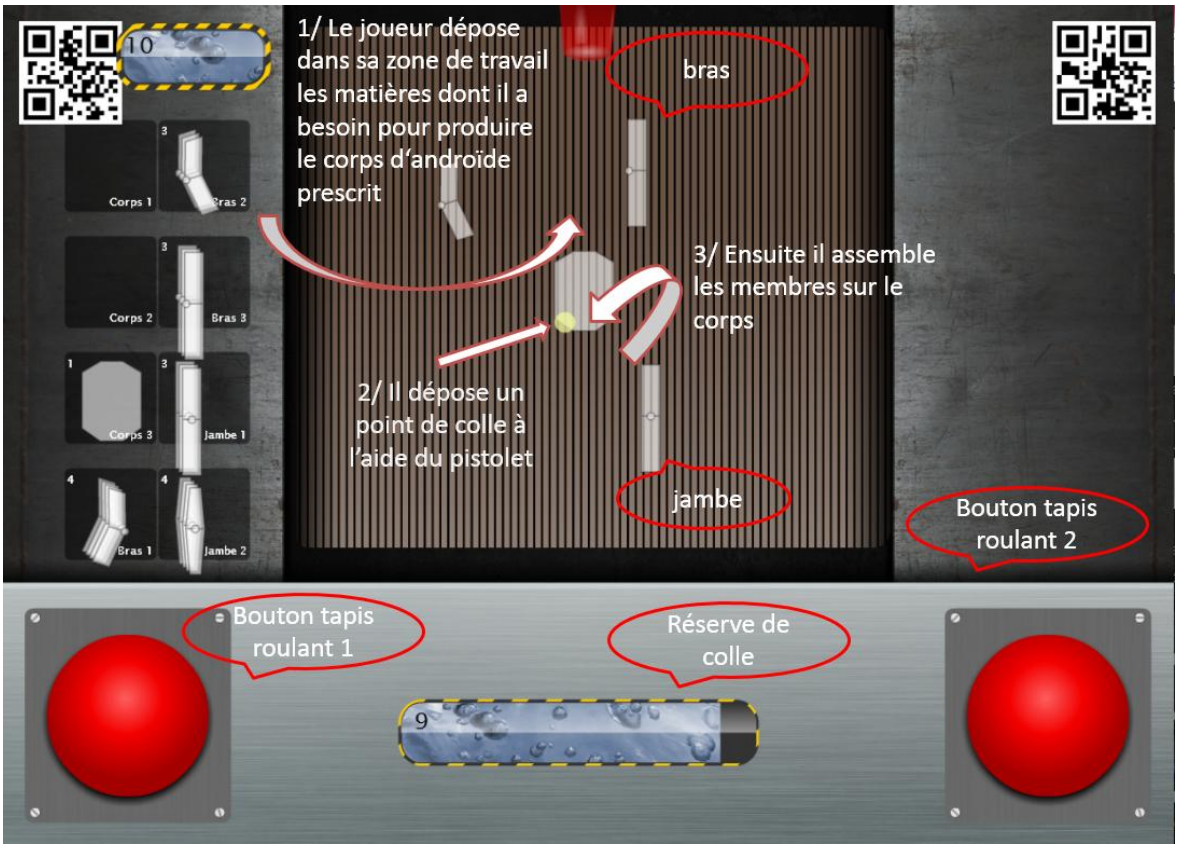


Figure 67. Le poste d'assemblage. Le joueur dépose les matières sur le tapis. A l'aide du pistolet à colle, il dépose des points de colle sur le corps, puis assemble à l'aide d'un drag'n drop les membres au corps

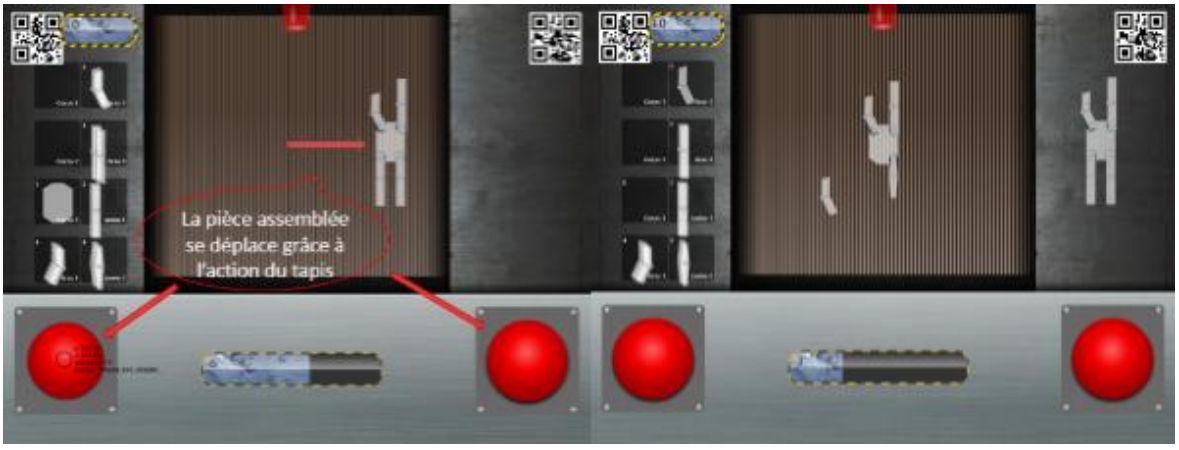


Figure 68. (A gauche) En appuyant sur les deux boutons, le tapis roulant actionné déplace la pièce assemblée dans la zone de sortie. Le joueur peut alors travailler sur un nouvel assemblage (à droite)

Une fois déposée, la colle met 10 secondes à sécher : si le joueur assemble un membre sur un point de colle « sec », il introduit alors un nouveau défaut de fabrication dans le produit. S'il utilise le pistolet alors que sa réserve de colle est vide, il « casse » alors le pistolet, l'empêchant de l'utiliser pendant une minute (et pénalisant ainsi la chaîne de production complète). Une fois le corps et les membres assemblés, le joueur doit appuyer sur les deux boutons rouges en même temps, ce qui déclenche le tapis roulant, lequel déplace alors l'assemblage en zone de sortie.

Dans une première itération, les membres en zone d'entrée ne sont pas triés, rendant le travail de l'assembleur plus complexe. Lors du débriefing et faisant partie d'une amélioration en phase avec les concepts « LEAN », les apprenants ont proposé d'eux même de ranger les membres en zone d'entrée par type (telle que sur la figure 67), ainsi qu'une aide de guidage montrant l'endroit où déposer la colle sur les corps, ce qui a effectivement amélioré la productivité de l'assembleur.

6.3.3.3 Poste d'imprégnation

Sur un pupitre interactif, l'opérateur d'imprégnation va « imprégner » le corps inerte produit par le poste d'assemblage d'une « personnalité » en contrôlant la puissance et dirigeant un rayon laser numérique, afin d'en faire un réplicant complet et doté d'une intelligence, d'une mémoire et d'un comportement. Pour représenter cet état « complet », l'image représentant l'assemblage se transforme peu à peu en une image plus « réaliste », montrant la forme finale de l'un des trois produits possibles selon la forme du corps (un réplicant « Roy », « Pris » ou « Zohra » selon l'univers fictionnel de « Blade Runner »). Pour parvenir à imprégner le corps, l'opérateur utilise deux interfaces tangibles posées sur le pupitre dans la zone de travail : l'une pour contrôler la puissance du rayon d'imprégnation, sensibles à la rotation impulsée par l'utilisateur, l'autre pour en contrôler la direction de par son positionnement.

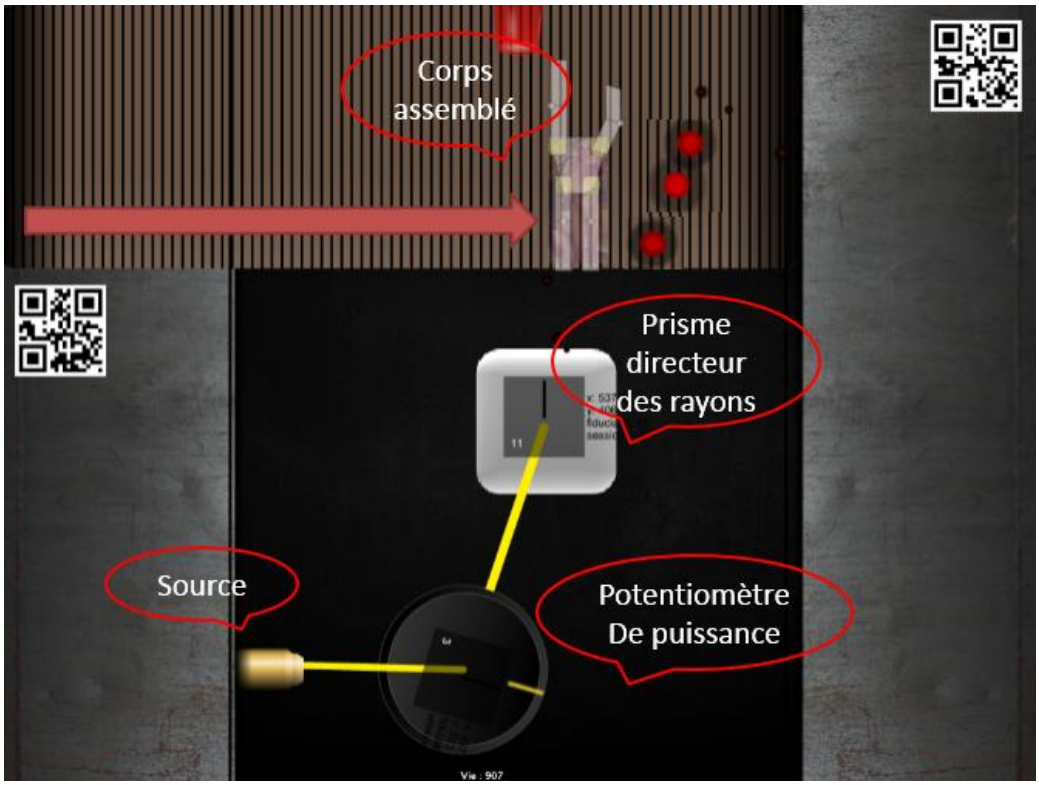


Figure 69: Le poste d'imprégnation tel qu'affiché sur le pupitre : les corps assemblés se déplacent automatiquement vers la sortie. Le potentiomètre de puissance est manipulé par la rotation de l'objet vert (voir figure 70), et le prisme directeur par le positionnement de l'objet rouge

Figure 70. L'imprégnatrice en action

L'opérateur doit gérer la puissance du rayon : si le réplicant n'est pas assez ou trop « imprégné », un défaut y est introduit. Lorsque les rayons touchent autre chose qu'un corps assemblé, cela endommage l'appareil, qui devient en panne pendant une minute.

6.3.3.4 Poste de manutentionnaire

Le manutentionnaire à un rôle central dans la chaîne de production, acheminant les matières entre les différents postes. Il utilise pour cela une tablette comportant des cases où il peut stocker les matières qu'il transporte.

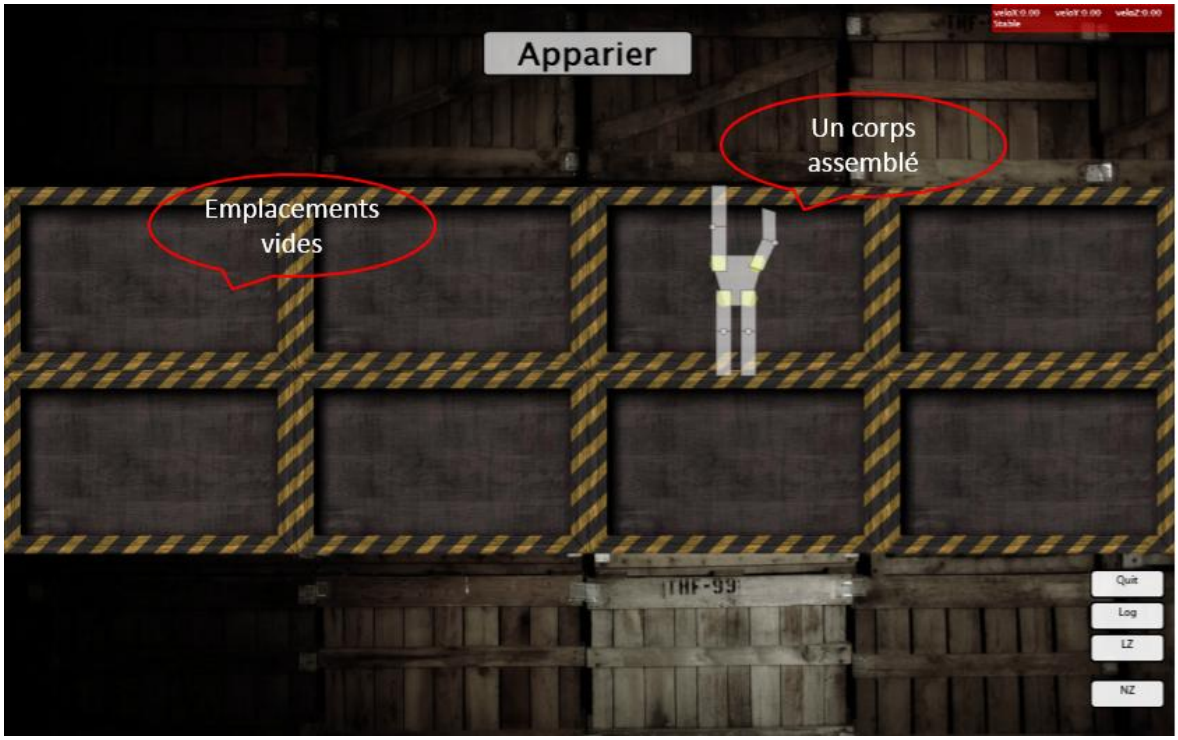


Figure 71. L'écran de la tablette décontextualisée

Lorsque la tablette est contextualisée avec une zone d'un poste, elle affiche le stock de son chariot ainsi que la zone contextualisée. Le manutentionnaire peut ainsi y prendre ou y déposer les matières par drag'n drop. La figure 72 montre la tablette contextualisée avec divers postes.

Figure 72. A gauche, la tablette est contextualisée avec la zone du magasin: le manutentionnaire peut prendre des matières premières en les glissant avec le doigt. A droite elle est contextualisée avec la zone d'entrée du poste d'assemblage.

Figure 73. Le manutentionnaire dépose des pièces que l'assembleur attendait pour pouvoir finir d'assembler le réplicant prescrit. Lorsque le manutentionnaire reprend sa tablette et la soulève, elle est automatiquement décontextualisée, n'affichant alors que le contenu de son stock « local »

6.3.3.5 Magasin

Le poste du magasin est déployé sur la table Surface Microsoft. Le magasin a pour rôle de stocker des matières premières ainsi que des réplicants. Trois rôles sont assignés à ce poste : le manutentionnaire qui vient prendre des matières premières et déposer les réplicants produits, le magasinier qui gère les stocks et met à disposition les matières premières pour le manutentionnaire, et enfin le testeur qualité chargé de tester les produits finis à l'aide de son application de test de Voigt-Turing sur IPhone.

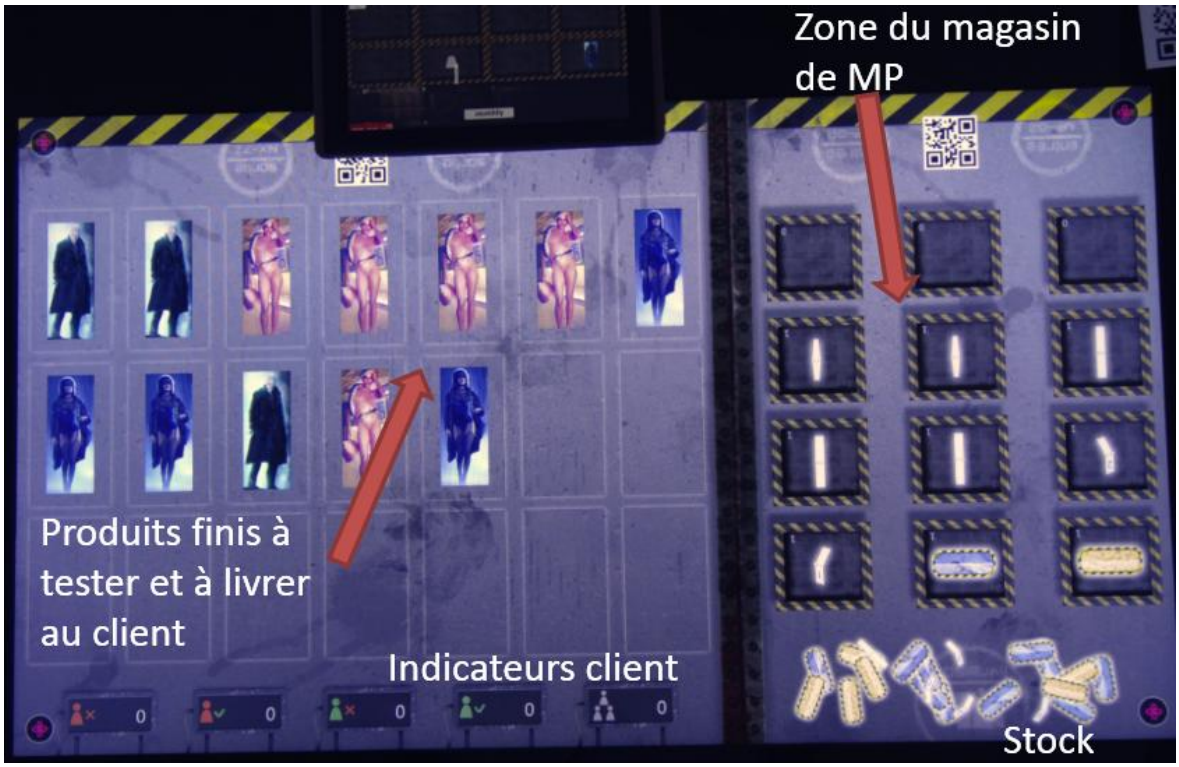


Figure 74.Photo de la table Surface. A gauche est la zone dédiée aux produits finis. En bas à droite, celle du stock que le magasinier met à disposition du manutentionnaire en les glissant dans la zone du magasin de matières premières (en haut à droite).

Les indicateurs clients sont mis à jour lorsque le testeur qualité fini un test. Les compteurs représentent (de gauche à droite sur la figure 74) :

- Réplicants défectueux et rejetés par le testeur
- Réplicants défectueux mais acceptés par le testeur
- Réplicants sans défaut mais rejetés par le testeur
- Réplicants sans défaut et acceptés par le testeur
- Réplicants livrés

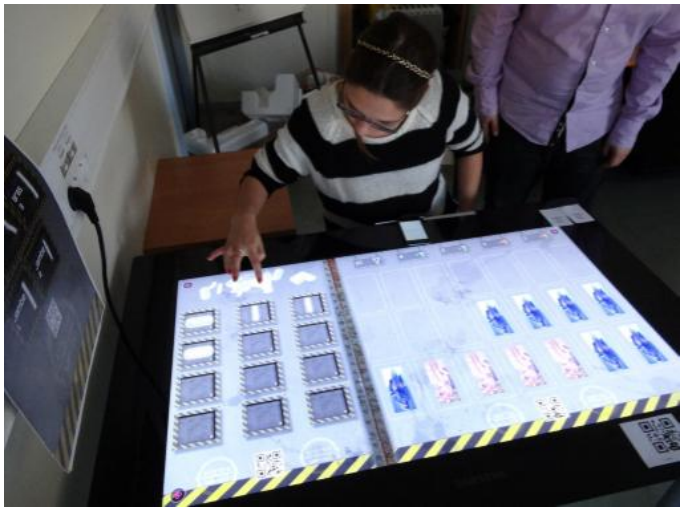


Figure 75. Photo du magasin

6.3.3.6 Testeur Qualité

Inspiré du « Blade Runner », le testeur qualité teste les réplicants produits en leur faisant passer le test de « Voigt-Turing », déterminant si le réplicant créé est doté de capacités empathiques. Le test consiste en une série de questions auxquelles le réplicant répond automatiquement, le testeur doit juger en fonction de la réponse si elle démontre une certaine empathie ou plutôt une « froideur » indiquant ainsi probablement un défaut dans sa fabrication. A tout moment le testeur peut décider de livrer le produit fini au client (le jugeant bon) ou de le rejeter, il peut consulter autant de questions/réponses qu'il le souhaite avant de prendre sa décision (mais le client attend !). Pour effectuer le test sur un réplicant le testeur qualité pose l'IPhone sur la table surface dans le magasin, sur la case du réplicant à tester (voir figure 80).

Figure 76. Une fois les informations transférée, un visuel apparaît et le testeur fait défiler les questions et les réponses du réplicant et décide à tout moment de le livrer ou de le retirer.

Exemples de questions/réponses (en gras la réponse attendue d'un réplicant sans défaut) :

- Votre fils vous montre ses papillons, et le bocal où il les tue : **Je le fais examiner** / Je le félicite
- Quelqu'un vous offre un portefeuille en vachette : **Je le refuse** / Je l'accepte
- Vous êtes dans le désert, devant vous une tortue. Elle est retournée et le soleil lui brûle le ventre : Je la regarde / **Je la retourne**
- Etc. (il existe une vingtaine de questions)

6.3.3.7 Poste moniteur

Le poste moniteur est utilisé par le formateur pour avoir des indicateurs (6.4) sur l'itération en cours, ainsi que pour déployer les modifications du jeu (6.3.4) que les apprenants peuvent proposer. Il permet également des fonctionnalités de plus haut niveau telles que sauvegarder l'état du jeu ou mettre tous les clients en pause.

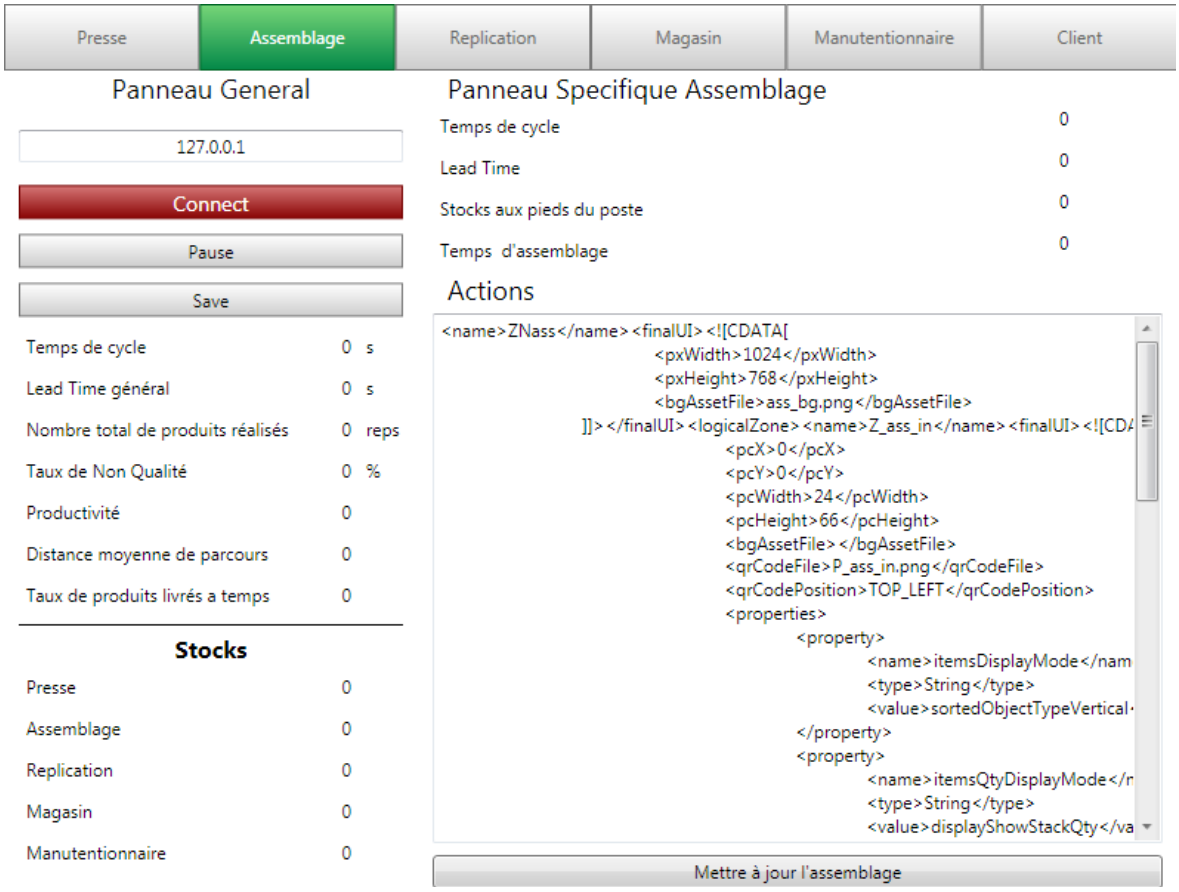


Figure 77. Ecran moniteur

6.3.4 Modifications

Entre chaque itération du jeu, les apprenants font un débriefing de l'itération précédente avec le formateur, dans le but d'exprimer les problèmes qu'ils ont constatés ainsi que des d'améliorations possibles. Plusieurs types d'améliorations impliquant des modifications dynamiques sont prévus, ayant chacun plusieurs niveaux.

Postes	Description de l'amélioration	Types d'améliorations	Niveaux de l'amélioration
Améliorations génériques pour tous les postes	Amélioration des indicateurs donnés aux utilisateurs	1 - Amélioration Feedback utilisateur	0: Aucun affichage sur les quantités de MP 1: Affichage des quantités de MP courantes 2: Affichage des quantités de MP requises + courantes
	Possibilités de trier des objets présents	2 - Tri des objets (Zone d'entrée et de sorties)	0: Aucun tri 1: Tri des objets selon leurs types 2: Tri et affichage du nombre d'objets
	Ajout d'élément graphique aidant le guidage	3 - Guidage (Aide à la transformation)	0: Aucune aide 1: Aide de guidage spécifique à la tâche
	Modification de la topologie	4 - Changement des zones logiques	
Presse	Amélioration des indicateurs donnés aux utilisateurs	1-0	0: Aucun affichage sur les quantités de MP 1: Affichage des quantités de MP courantes 2: Affichage des quantités de MP requises + courantes
	Changement de visualisation des sélections de moules	3-1	0: Textuel 1: Iconique
	Affichage du prochain ordre de fabrication	3-2	0: Aucun affichage 1: Affichage du prochain ordre
	Possibilités de trier des objets présents	2-0	0: Aucun tri 1: Tri des objets selon leurs types 2: Tri et affichage du nombre d'objet
Collage	Ajout d'une zone de séchage	4-1	0: Pas de zone de séchage 1: Zone de séchage
	Ajout d'un patron pour aider le guidage	3-1	0: Aucune aide 1: Aide de guidage à la pose de colle
	Ajout des niveaux de séchage de la colle	3-2	0: Aucune aide 1: Aide de guidage au séchage de colle
	Amélioration des indicateurs donnés aux	1-3	0: Aucun affichage sur les quantités de MP

	utilisateurs		1: Affichage des quantités de MP courantes 2: Affichage des quantités de MP requises + courantes
	Possibilités de trier des objets présents	2-1	0: Aucun tri 1: Tri des objets selon leurs types 2: Tri et affichage du nombre d'objets
Imprégnation	Amélioration des indicateurs donnés aux utilisateurs	1-4	0: Aucun affichage sur les quantités de MP 1: Affichage des quantités de MP courantes 2: Affichage des quantités de MP requises + courantes
	Possibilités de trier des objets présents	2-2	0 : Aucun tri 1: Tri des objets selon leurs types 2: Tri et affichage du nombre d'objets
Manutention-naire	Amélioration des indicateurs donnés aux utilisateurs	1-5	0: Aucun affichage sur les quantités de MP 1: Affichage des quantités de MP courantes 2: Affichage des quantités de MP requises + courantes
	Possibilités de trier des objets présents	2-3	0: Aucun tri 1: Tri des objets selon leurs types 2: Tri et affichage du nombre d'objets
	Présence du manutentionnaire	4-0	0: Présence du manutentionnaire entre deux postes
			1: Absence du manutentionnaire entre deux postes

Figure 78. Formalisation des modifications dynamiques du jeu

La figure 78 montre les modifications prises en compte par l'architecture, dont on distingue quatre types principaux :

- Amélioration du feedback utilisateur : chaque niveau donne plus d'information à l'utilisateur concernant sa performance (quantités en entrée et en sortie)
- Tri d'objets : donne la possibilité d'avoir des piles d'objets triés par type en entrée et en sortie d'un poste. Réduit donc le temps qu'il faut à l'utilisateur pour prendre les pièces dont il a besoin pour effectuer sa tâche.
- Guidage visuel à la tâche : donne des indications visuelles aidant l'opérateur dans sa tâche (par exemple monter les endroits où déposer la colle évite à l'opérateur d'assemblage de consulter son standard de travail pour coller)
- Modification de topologie : des modifications impactant le flux des matières, par exemple les apprenants peuvent décider de mettre deux postes côte à côte, supprimant le rôle du manutentionnaire entre eux.

Suite au débriefing, le formateur sélectionne les améliorations sur le poste moniteur, qui seront mises en place à la prochaine itération du jeu.

6.4 Indicateurs de performance et traces utilisateur

Un aspect important de l'informatisation du jeu Lean est de pouvoir tracer l'activité des joueurs, dans le but d'évaluer leur progression pendant le jeu, ainsi que de fournir des indicateurs de performance pouvant appuyer les phases de débriefings entre les itérations et ainsi permettre aux apprenants de constater (ou non) l'impact des améliorations construites ensemble. Ces indicateurs sont visualisables en temps réel par le formateur sur le poste moniteur.

Nous distinguons deux types d'indicateurs : des indicateurs généraux permettant d'évaluer la performance globale de la chaîne de production, et des indicateurs spécifiques à chaque poste.

6.4.1 Indicateurs généraux

- **Temps de cycle :** Temps moyen qui s'écoule entre la production de 2 réplicants (en sortie du contrôle QUALITE « Testeur »)
- **Lead time général :** durée moyenne de production d'un réplicant (durée comprise entre le démarrage de la chaîne – début de moulage de la pièce - et la fin du cycle – arrivée du réplicant en magasin - divisé par le nombre de réplicants produits en tout)
- **Nb total de produits réalisés (avant contrôle) :** nb total de produits sur une simulation (nb de produits réalisés et arrivés dans le magasin)
- **Taux de non qualité :** Le taux de rebut est le nombre de produits rejetés par le testeur qualité divisé par le nombre de produits testés sur un cycle de simulation (et *100 si on le veut en %)
- **Productivité :** nombre total de pièces produites divisé par le nombre d'opérateurs sur la chaine (5)
- **Stocks au pied de chaque poste :** Nombre total de pièces immobilisées sur chaque atelier à la fin de la simulation
- **Distance moyenne de parcours :** Ici il faut prévoir que chaque pièce traverse une distance de tant de mètres (disons 15m). Après activation d'une amélioration qui consiste à rapprocher les ateliers assemblage et réplication des ateliers presse et assemblage, on descendrait alors à 5m.
- **Le taux de produits livrés à temps :** nombre de produits livrés à temps divisé par le nombre total de produits livrés (à temps et en retard).

6.4.2 Indicateurs spécifiques aux postes

Les indicateurs suivants sont fournis pour chaque poste :

- **Temps de cycle** : Temps moyen passé entre deux sorties de produits
- **Lead Time** : Temps moyen passé par une pièce entre sa sortie de la zone d'entrée et son entrée dans la zone de sortie
- **Stocks au pied du poste** : Nombre moyen de produits immobilisés sur le poste : se calcule en prenant le nombre de pièces présentes sur le poste à chaque sortie de pièces

6.5 Dispositifs et technologies

Nous décrivons ici les infrastructures matérielles et logicielles spécifiques utilisées pour le prototype.

6.5.1 Pupitres interactifs

Les pupitres interactifs, au nombre de 4, ont été conçus au laboratoire par Florent Delomier et permettent à un ou plusieurs utilisateurs de travailler simultanément sur un pupitre, interagissant principalement via des interfaces tactiles et/ou tangibles. Ils comportent chacun:

- Un dispositif de présentation de l'information (un vidéoprojecteur)
- Un dispositif d'entrée d'information, notamment de prise en compte de position d'objets tangibles ou tactiles. Ce dispositif de tracking est composé de LED infrarouges et d'une caméra infrarouge.
- Une unité centrale à laquelle sont connectés ces composants.

Caractéristiques:
Processeurs E6800, 2GB Ram
Windows 7
Résolution 1024 * 768.

Le choix de la technologie s'est porté sur l'utilisation de la technologie DSI, qui utilise une propriété spécifique du plexiglass Endlighten, contenant des microparticules responsables d'une diffusion homogène des rayons infra-rouges dans l'ensemble de la plaque. Les doigts sont détectés par diffusion de la lumière infrarouge au contact avec la surface du plexiglas. Dans le cas Lea(r)nIt, les pupitres ont été utilisés pour les rôles d'opérateurs fixes (presse, assemblage) et impliquant des interfaces tangibles, en particulier le poste d'opérateur d'imprégnation (6.3.3.3).

Figure 79. Le premier prototype de pupitre (à gauche), et celui réalisé par un menuisier (à droite)

6.5.2 Tablette Android

L'utilisation de tablette est dédiée aux rôles mobiles. Dans le cas Lea(r)nIt il s'agît du rôle « manutentionnaire », chargé d'acheminer les matières premières et transformées d'un poste à l'autre. La tablette est notamment dotée d'une caméra dorsale permettant de scanner les QR codes pour l'opération de contextualisation (5.2.2.5). En dehors de cette spécificité, les interactions utilisateurs sur la tablette sont exclusivement d'ordre tactile. La partie cliente de l'architecture a été déployée sur la tablette via le format APK utilisé pour les déploiements logiciels sous Android.

Caractéristiques:
Acer Iconia Tab A 510
Quad core Tegra 3 NVIDIA
OS Android Ice Cream Sandwich
Résolution 1280 * 800

6.5.3 Table Surface Microsoft

La table Surface Microsoft (maintenant appelée PixelSense) a été introduite dans les expérimentations Lea(r)nIt pour le poste magasin qui stocke les matières premières et les produits finis. La table permet des interactions tactiles et détecte également les objets tangibles. De plus il était possible, en posant un IPhone sur la table et sur une case contenant un réplicant, de transférer des informations sur l'IPhone (figure 80) via NFC (Near Field Communication).

L'application a été développée par Symetrix en C# WPF, utilisant le .NET Framework 4.0 et Surface SDK 2.0.

Caractéristiques:
Windows 7 Professional 64 bits for Embedded Systems
Résolution : 1920 * 1080
Athlon X2 Dual-Core 245^{e} (2.9GHz)
GPU: AMD HD 6750M
DD: 320Go SATA2
RAM : 4Go DDR3

6.5.4 IPhone

Un IPhone a été utilisé pour le rôle de « Blade Runner », responsable de la qualité des réplicants produits. Un mini-jeu sous forme de questions-réponses permet de tester le réplicant, à la suite desquelles le joueur décide de garder ou retirer le réplicant (6.3.3.6). L'application sur IPhone a été développée par Symetrix.

Caractéristiques :
iOS 6.0 ou supérieur (iPhone 3GS et sup et iPod Touch 4ème génération et sup)
Résolutions supportées : 320*480, 640 * 960 et 640 * 1136

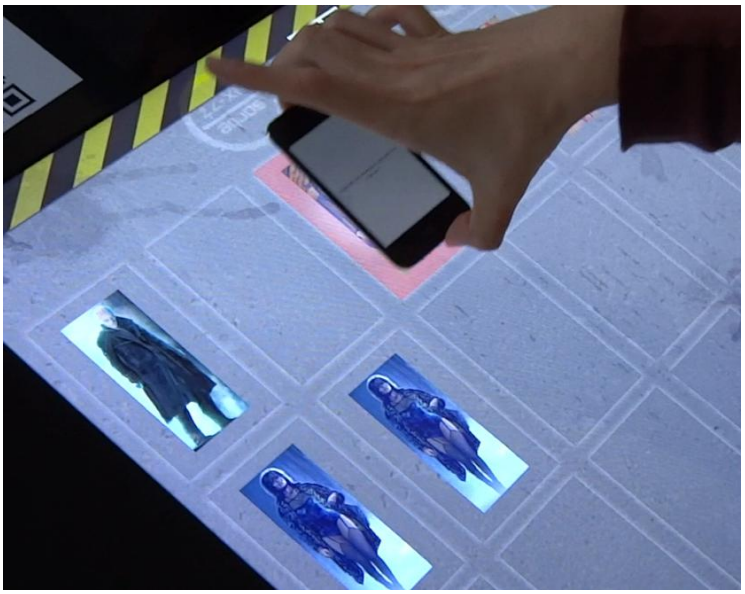


Figure 80. Lors des expérimentations Lea(r)nIt, la personne pose l'IPhone sur la table Surface, qui le détecte et lui transfère alors les informations relatives réplicant sous-jacent afin de pouvoir lui faire passer le test de « Voigt-Turing ».

6.5.5 Développement

Nous listons ici les produits utilisés pour les développements. La partie cliente 'Atelier SEGAREM' a été développée en ActionScript 3.0, compilée avec Flex SDK 4.6 sous FlashDevelop 4.0.4.

Elément	Lien
FlashDevelop 4.0.4	http://www.flashdevelop.org
Flex 4.6	http://www.adobe.com/devnet/flex/flex-sdk-download.html
TortoiseSVN 1.7.10	http://tortoisesvn.net/downloads.html
AIR 3.2	http://get.adobe.com/fr/air/
Flash player et debugger 11.1	http://www.adobe.com/support/flashplayer/downloads.html
TUIO 0.8 for AS	http://tuio.org/?software
CCV	http://ccv.nuigroup.com/
Tweenlite 11	https://www.greensock.com/tweenlite/
ZXing 2.0	http://code.google.com/p/zxing/
Monster debugger 3.02	http://demonsterdebugger.com/
Electroserver 5.3.3	http://www.electrotank.com/resources/downloads.html

Figure 81. Liste des produits utilisés pour le développement

- ActionScript est un langage de script orienté objet et prototype basé sur le standard de programmation ECMAScript (la version 3.0 d'ActionScript est conforme à 100% avec la norme ECMA-262, révision 3). ActionScript permet notamment d'ajouter de l'interactivité aux animations flash en pilotant des « movieclips » (conteneurs graphiques permettant de hiérarchiser les animations), et les différents objets multimédias (images, son, vidéo…).
- FlashDevelop est un environnement de développement gratuit et open-source permettant d'éditer du code ActionScript.

- Flex est un kit de développement logiciel permettant de créer et de déployer des applications internet riches utilisant le langage ActionScript.
- Flash Player permet d'exécuter du code ActionScript. AIR (Adobe Integrated Runtime) assure le déploiement multiplateforme des applications Flash et Flex pour les pupitres, tablettes et PC clients.
- TUIO (Tangible User Interface Object Protocol) est un cadre de développement ouvert qui définit un protocole basé sur UDP et des interfaces de programmation destinés aux interfaces multi-touch et tangibles. TUIO for AS est une librairie ActionScript implémentant TUIO.
- Community Core Vision (CCV) est une plateforme open source de traitement vidéo optique ou infra-rouge. Il permet à partir d'un flux vidéo d'extraire des informations et des évènements pouvant être utilisés par la suite. Les informations extraites sont des coordonnées d'objets ainsi que leurs tailles. Il est possible de mettre en même temps des objets ayant une forme spécifique, des objets possédant des tags fiduciels et des doigts. CCV reçoit en entrée un flux vidéo issue de la caméra en transforme ce flux en un message qu'il envoie sur le réseau selon un langage (TUIO) et un protocole de communication (UDP).
- Tweenlite est une librairie permettant de gérer des animations complexes d'objets et conteneurs graphiques Flash.
- ZXing est une librairie développée par Google et permettant de générer et de lire et décoder des codes-barres 2D ou QR codes.
- Monster Debugger est un debugger haut niveau pour les applications Flash.
- Electroserver est un moteur d'exécution de jeu à vocation multi-joueurs. Il peut être étendu à l'aide de plugins Java ou ActionScript et permet de centraliser les données et de gérer la couche de communication entre les clients (5.4.3).

Subversion est un logiciel de gestion de version hébergé par la fondation Apache. TortoiseSVN est un client Subversion intégré au Shell Windows. Le repository est disponible à l'adresse https://svn.liris.cnrs.fr/learnit/learnit . La version utilisée pour la deuxième expérimentation est 700.

6.6 Expérimentations Lea(r)nIt

Trois expérimentations impliquant trois groupes différents de 8 personnes chacun ont eu lieu pour évaluer les hypothèses du projet SEGAREM : Un premier groupe de 8 personnes a suivi un cours sur le LEAN en utilisant le jeu LEAN original basé sur l'utilisation de LEGO, adapté au scénario de Lea(r)nIt. Ce premier groupe sert de groupe témoin.
Les deuxième et troisième groupes ont suivi le même cours, mais en utilisant la version Lea(r)nIt du jeu. Chaque séance a duré en moyenne 3 heures. Le cours a dans les trois cas été conduit par le même formateur, expert en concept LEAN. Le panel est constitué pour la majeure partie d'étudiants en Génie Industriel, les concepts LEAN sont donc au programme de leur cursus.

Le but était de pouvoir évaluer sous deux angles :

- l'expérience utilisateur : l'apprenant est-il plus motivé par l'expérience Lea(r)nIt, quel est son ressenti ?

- l'utilité : la qualité de l'apprentissage est-elle, sinon meilleure, au moins aussi bonne avec les dispositifs conçus pour Lea(r)nIt qu'avec le jeu original ?

6.6.1 Evaluation de l'expérience utilisateur

Deux axes principaux ont été évalués :

- l'acceptabilité : est-ce que les moyens mis en œuvre donnent envie d'utiliser le dispositif ?
- l'utilisabilité : a-t 'on facilement accès à une fonction donnée ? le système est-il intuitif ?

6.6.2 Evaluation des connaissances LEAN

Chaque apprenant a au préalable rempli un questionnaire de « pré-test » devant évaluer ses connaissances des principes du LEAN avant le cours. A la fin de la session ils ont également rempli un questionnaire de « post-test » identique, devant évaluer leurs connaissances des principes LEAN après avoir suivi le cours.

Les questionnaires sont en cours d'analyse, et les résultats n'ont pas pu être disponibles pour être inclus dans ce rapport. Néanmoins et à titre d'exemple, deux questionnaires post-test de connaissances LEAN et d'expérience utilisateur ont été mis en annexe.

6.6.3 Bilan des expérimentations

Malgré la complexité inhérente au développement de systèmes exigeants en interactions (comme les jeux vidéo), nous avons pu faire une expérimentation de Lea(r)nIt dans des conditions d'utilisabilité acceptables. Nous avons pu constater lors des sessions que le système de distribution des zones numériques permettait une grande souplesse dans le déploiement de l'application et des modifications dynamiques, et que l'architecture a pu répondre aux besoins du jeu. Néanmoins nous n'avons pas pu, par manque de temps, implémenter tous les aspects génériques de l'architecture, et certains comportements, en particulier la description du fonctionnement des outils qui a dû être simplifiée, ont dû être codés soit en utilisant des enchaînements d'utilisation d'outils (menant à une description plus complexe), soit « en dur » pour parvenir à fournir toutes les fonctionnalités requises à temps.

Afin d'évaluer la généricité de notre architecture, il nous faudrait y projeter et implémenter d'autres types de jeux. Dans un cas simple on pourrait imaginer une sorte de morpion partagé dont chaque joueur ne voit qu'une partie des cases, un troisième joueur changeant à chaque tour les zones de perception des deux adversaires. Mais les capacités d'interactions utilisateur et de collaboration de l'architecture permettent de faire bien plus qu'un simple morpion, et une évaluation sur un autre cas au moins aussi complexe que celui de Lea(r)nIt serait nécessaire.

Conclusion

La création de l'architecture fut un travail exigeant mais aussi très enrichissant, impliquant la collaboration d'acteurs de domaines différents et néanmoins complémentaires, dans la conception et le développement de la solution. Un travail au cours duquel, suivant notre démarche itérative et incrémentale, de nombreux éléments ont été continuellement remis en cause, nécessitant généralement d'être réécrits afin de pouvoir en tester rapidement les aspects tant logiciels qu'interactionnels, toujours dans le but de produire la meilleure expérience ludique et pédagogique possible.

Grâce à cette approche et malgré quelques difficultés rencontrées, l'architecture a tenu ses promesses de support d'une activité collaborative distribuée, aux interactions utilisateur riches, et nous permet d'imaginer aller encore plus loin pour faciliter le processus de création de tels serious games. Aller plus loin au niveau des outils pour permettre une génération plus poussée du code, à l'aide d'éditeurs graphiques décrivant toujours plus finement les fonctionnalités souhaitées : ainsi eIRVO+, comme « éditeur de niveaux », pourrait aller jusqu'à la description de l'interface finale d'une zone logique dans une zone numérique donnée (taille, position, contextualisation, etc..), le modèle de description des outils logiques pourrait bénéficier d'un éditeur de « briques de comportements » tel que celui de Virtools. Enfin, aller plus loin au niveau des modèles : en les enrichissant de « game patterns », offrant des cadres de solutions issues de bonnes pratiques en réponse à des problèmes ludiques typiques, et dont le vocabulaire peut être compris par les différents acteurs de la conception de jeux pédagogiques, de l'ergonomie et du développement. Ainsi la génération du cadre d'un serious game pourrait être encore facilitée, incitant à des prototypages encore plus rapides, dans un processus toujours plus industrialisé de production de jeux pédagogiques en environnement informatisé.

Le jeu est un médium de communication puissant, particulièrement d'un point de vue pédagogique : chaque partie nouvelle qui commence y ouvre un temps neuf, mais pas tout à fait sans mémoire car demeure le souvenir des éventuels échecs antérieurs. Si ces derniers peuvent n'être considérés que comme des essais, ratés, peut-être, sitôt que le joueur en a compris les raisons, ils sont aussi des leçons pour ne plus, désormais, se laisser prendre au dépourvu. Les simulateurs de vol à l'échelle 1 ont prouvé qu'un apprentissage en réalité mixte peut être utile, et il est probable que nous allions encore plus loin dans l'immersion de l'Homme dans un monde numérique. Alors il faudra se demander si à l'instar du malin génie de Descartes nous ne nous tromperions pas d'objectif, en fournissant une illusion si indissociable de la réalité qu'elle cesserait alors d'être une illusion. Ici réside toute l'ambiguïté du jeu : en dehors de la puissance des dispositifs utilisés, de l'immersion qu'ils permettent et de leur finalité pédagogique qu'est-ce qui, en fin de compte, se joue là ? Une part de la vie ou bien la vie elle-même ? Il semble possible de perdre un lien avec les enjeux de la réalité dans laquelle, pour reprendre la métaphore du jeu vidéo, l'on a qu'une seule vie et aucun mécanisme de sauvegarde permettant de recharger la « partie » en cas d'échec. Cependant tant que les hommes parviennent à repérer la frontière entre leur activité ludique et leur confrontation avec la réalité du monde, celle des autres, des situations et des choses, ils ne courent aucun risque de se méprendre.

Bibliographie

Banks D., 2010. *U.S. Army Turns to Videogames for Training.* [en ligne] Disponible sur http://www.wired.com/geekdad/2010/10/jtcoic/all/1,Consulté le 12/12/2012

Beedle M. et al., 2001. *Agile Manifesto*. [en ligne] Disponible sur http://agilemanifesto.org/iso/fr/, consulté le 10/12/2012

Bohemia Interactive Studio, 2007. *Virtual Battlespace 2.* [en ligne] Disponible sur http://products.bisimulations.com/products/vbs2/overview, consulté le 12/12/2012

Bohemia Interactive Studio, 2009. *Armed Assault 2*. [en ligne] Disponible sur http://www.arma2.com/home_en.html, consulté le 10/12/2012

Bouwsma O. K., 1949. *Descartes' Evil Genius.* The Philosophical Review, Cornell University, Vol. 58, No. 2. (Mar., 1949), pp. 141-151. [en ligne] Disponible sur http://home.sandiego.edu/~baber/analytic/Bouwsma1949.pdf

Bruce B. et al, 2002. *AR Quake*. [en ligne] Disponible sur http://wearables.unisa.edu.au/projects/arquake/, consulté le 10/12/2012

Caudell T. P. & Mizell D. W., 1992. *Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes*. Proceedings of 1992 IEEE Hawaii International Conference on Systems Sciences, pp 659-669.

CAILLOIS R., 1957. *Les jeux et les homes*. Gallimard

Calvary G., Coutaz, J., Thevenin, D., 2003. *A Unifying Reference Framework for Multi-Target User Interfaces*. Interacting with Computer 15, 3 289–308

Chalon R., 2004. *Réalité Mixte et Travail Collaboratif : IRVO, un modèle de l'Interaction Homme-Machine*. Thèse en informatique. Lyon : Ecole Centrale de Lyon, 212 p.

Coutaz J., Nigay L., 2001. *Architecture logicielle conceptuelle des systèmes interactifs*. In : Kolski C. Eds. Analyse et conception de l'IHM. Hermès, Paris, chapitre 7, pp 207-246.

Couture R.N., 2010. *Interaction Tangible, de l'incarnation physique des données vers l'interaction avec tout le corps*. HDR, Université de Bordeaux 1, 90 p.

Dassault Systems, 2009. *Virtools.* [en ligne] Disponible sur http://www.3ds.com/products/3dvia/3dvia-virtools/, consulté le 10/12/2012

David B., Chalon R., Delotte O. and Masserey G., 2007. *ORCHESTRA: formalism to express static and dynamic model of mobile collaborative activities and associated patterns*. HCI International 2007 Beijing 22-27 July 2007 In J. Jacko (Ed.): Human-Computer Interaction, Part I, HCII 2007, LNCS 4550, pp. 1082–1091

Delomier F., David B., Chalon R., Bénazeth C., 2012. *Conception et mise en œuvre de Learning Games Collaboratifs Contextualisés*. SYSCO - Première conférence francophone sur les systèmes Collaboratifs, 28-30 sept. 2012, Sousse, Tunisie

Dick P. K., 1966. *Do Androids Dream of Electric Sheep?* Double Day

Djaouti D., 2011. *Serious Game Design Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire*. Thèse, Université de Toulouse 330 p.

Epic Entertainment, 1998. *Unreal Development KIT*. [en ligne] Disponible sur http://www.unrealengine.com/udk/, consulté le 12/12/2012

Fitzmaurice G. and Buxton, W., 1997. *An empirical evaluation of graspable user interfaces: Towards specialized, space-multiplexed input*. In Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems, volume 1 of PAPERS: Handy User Interfaces, pages 43-50

Florida Virtual School, 2009. [en ligne] Disponible sur http://www.flvs.net/Pages/default.aspx, consulté le 10/12/2012

François-Gilles Ricard, *in* Mariais C., Dupin C., 2010. *Le Serious Game Session : les ressorts de jeu au service du présentiel*, présentation effectuée lors du Serious Game Expo, 23/11/2010, Lyon.

Fuchs P., Moreau G., Papin J. P., 2001. *Le traité de la réalité virtuelle.* Ed. Techniques - Ingénieur 517 p.

Gregory J., 2009. *Game Engine Architecture Book*. Taylor and Francis Group, LLC 853 p.

Jehng J. C. J., 1997. *The psycho-social processes and cognitive effects of peer-based collaborative interactions with computers*. Journal of Educational Computing Research, 17(1):19-46.

Johnson D. W., and Johnson, R. T. eds. 1996. *Cooperation and the use of technology*. Handbook of Research for Educational Communication and technology New York, Macmillan : 1017-1044.

Juul J., 2003. *The game, the player, the world: looking for a heart of gameness*. In C. Marinka & R. Joost (Eds.), Level Up Conference Proceedings: Proceedings of the 2003 Digital Games Research Association Conference (pp. 30-45). Utrecht: University of Utrecht.

Juul J., 2005, *Half-real: Video Games between Real Rules and Fictional Worlds*. MIT Press.

Kitchen G., 1985, *Gamemaker.* [en ligne] Disponible sur http://www.garrykitchen.com/product_history/garry_kitchens_game maker.html, consulté le 10/12/2012

Leitner et al., 2008. *IncreTable, a mixed reality tabletop game experience* ACE '08 Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology pages 9-16

Liu, W. et al., 2007. *Mixed reality classroom* In Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts - DIMEA ’07. Perth, Australia, p. 65.

Mackay W., Velay G., Carter K., Ma C., et Pagani D., 1993. *Augmenting reality: adding computational dimensions to paper*, Commun. ACM, vol. 36, p. 96-97.

Magnusson L. V., 2011. *Game Mechanics Engine*. Master of thesis, Ostfold University College

Malone T. W., 1980. *What makes things fun to learn? A study of intrinsically motivating computer games*. Technical report, Xerox Palo Alto Research Center, Palo Alto, Calif., forthcoming

Mariais C., Pernin J. P., Michau F., 2011. *Learning Role-Playing Games : Méthodologie et formalisme de description pour l'assistance à la conception. Premiers résultats d'expérimentation.* Proceedings of EIAH'2011 Conférence : Environnements Informatiques pour l'Apprentissage Humain, 95-110, Editions de l'UMONS.

Marshall P., 2007. *Do tangible interfaces enhance learning?* In Proceedings of the 1st international conference on Tangible and embedded interaction, Baton Rouge, Louisiana, USA.

Michaud L., 2008. *Etude sur les Serious Games*, IDATE

Microsoft, 2012. *PixelSense.* [en ligne] Disponible sur http://www.microsoft.com/en-us/pixelsense/default.aspx, consulté le 12/12/2012

Milgram P. & Kishino F., 1994. *A taxonomy of Mixed Reality visual displays.* IEICE Transactions on Information and Systems E series D, 77, p.1321–1329.

MIT Labs, 2007. *Scratch.* [en ligne] Disponible sur http://scratch.mit.edu/, consulté le 10/12/2012

Norman D., 1989. *The Design of Everyday Things*. Double Day Currency, New York

Onlineformapro, 2010. *SGTools.* [en ligne] Disponible sur http://www.onlineformapro.com/serious_game.php, consulté le 10/12/2012

Orliac C., 2012. *Quels outils pour la conception de jeux pédagogiques en réalité mixte ?* ErgoIHM 2012

Ouadou K., 1994. *AMF : Un modèle d'architecture multi-agents multi-facettes pour Interfaces Homme-Machine et les outils associés.* Thèse de doctorat, Ecole Centrale de Lyon.

Overmars M., 1999. *Game Maker.* [en ligne] Disponible sur http://www.yoyogames.com/gamemaker/studio, consulté le 10/12/2012

Perry J. et al., 2008. *AR gone wild: two approaches to using augmented reality learning games in Zoos.* In Proceedings of the 8th international conference on International conference for the learning sciences - Volume 3. Utrecht, The Netherlands: International Society of the Learning Sciences, p. 322-329.

Platon, *La République.* VII 536c 537a, in Œuvres Complètes, trad. Léon Robin, Paris, Gallimard, Bibliothèque de la Pléiade, tome 1, 1950, p.1133.

Plummer J. 2004. *A Flexible and expandable architecture for computer games*, Thesis, ARIZONA STATE UNIVERSITY

Roschelle J., & Teasley S. D., 1994. *The construction of shared knowledge in Collaborative problem solving.* Nato ASI Series F Computer and Systems Sciences 128, pp 68.

Smith K., Johnson D. W., Jonson R. T., 1981. *Can conflict be constructive? Controversy versus concurrence seeking in learning groups.* Journal of Educational Psychology 73(5): 651-663.

Sommer R., 1969. *Personal space: the behavior basis of design.* Englewood Cliffs, Prentice-Hall, 177p.

Schrier K., 2006. *Using augmented reality games to teach 21st century skills.* In ACM SIGGRAPH 2006 Educators program on SIGGRAPH '06. ACM SIGGRAPH 2006 Educators program. Boston, Massachusetts, p. 15.

Squire K.D. & Jan M., 2007. *Mad City Mystery: Developing Scientific Argumentation Skills with a Place-based Augmented Reality Game on Handheld Computers.* Journal of Science Education and Technology, 16(1), p.5-29.

Stewart D., 1999. *The Musician's Guide to Reading and Writing Music*. Backbeat 117 p.

Susi T., Johannesson M., et Backlund P., 2007. *Serious games – An overview*, Technical Report University of Skövde, Sweden: School of Humanities and Informatics.

Tarpin-Bernard F., David B., 1999. *AMF : un modèle d'architecture multi-agents multi-facettes.* Techniques et Sciences Informatiques. Hermès. Paris. Vol. 18. No. 5. pp. 555-586.

Ullmer B. & Ishii H., 2000. *Emerging Frameworks for Tangible User Interfaces*. In HCI in the New Millenium, John M. Carroll, Ed. 579–601

U.S. Army, 2002. *Americas Army.* [en ligne] Disponible sur http://www.americasarmy.com/, consulté le 10/12/2012

Vygotsky L. S., 1978. *Mind in society: The development of higher psychological processes.* Chapter 6 Interaction between learning and development (79-91). Cambridge, MA: Harvard University Press

Wagner D. et Barakonyi I., 2003. *Augmented Reality Kanji Learning.* Proceedings of the 2nd IEEE/ACM Symposium on Mixed and Augmented Reality (ISMAR 2003), Tokyo, Japan, Oct. 7-10, 2003.

Wikipedia, *Gameplay.* [en ligne] Disponible sur http://fr.wikipedia.org/wiki/Gameplay, consulté le 10/12/2012

Zuckerman O., Arida S. & Resnick M., 2005. *Extending tangible interfaces for education: digital Montessori-inspired manipulatives.* In Proceedings of the SIGCHI conference on Human factors in computing systems. p. 859–868.

Annexe 1

Questionnaires Post-test

Expérimentation LearnIT
Projet SEGAREM

~~26/10/2012~~
15/11/2012

QUESTIONNAIRE POST-TEST SUR LES CONNAISSANCES EN LEAN DES PARTICIPANTS

Nom du participant : _____

Session d'expérimentation : ☐ matin ☒ après-midi

1. Les techniques d'organisation de la production reposent sur 2 piliers essentiels : quels sont-ils ?

- Juste à temps
- qualité

2. Citez 3 principes majeurs de la pensée LEAN.

- rigueur
- leadership

3. Qu'est-ce que le LEAN

☐ Une méthode d'amélioration
☒ Une manière de pensée
☒ Un ensemble d'outils visant à éliminer les gaspillages
☐ Un objectif que l'on vise

4. Parmi la liste des verbes suivants, lesquels traduisent une activité à valeur ajoutée du point de vue du client :

Détruire – Stocker - Attendre - Manutentionner - Sortir du magasin - Compter – Recopier - Dépanner - Grouper - Transformer -Chercher - Ranger - Transcrire - Trier - Signer - Déplacer - Nettoyer - Régler - Annuler - Changer l'outil - Contrôler - Réparer – Vérifier - Recommencer - Surveiller - Imprimer un état - Démonter - Ecrire

5. Citez les différents types de gaspillages que vous connaissez.

- stocks
- sur-production
- attente

6. Qu'est-ce que le PDCA ?

Plan / DO / check / Act

7. Qu'est-ce que le concept d'équilibrage des postes ? A quoi sert-il ?

↳ pour pas qu'il y ait trop de stocks en fin de poste car ce poste demande moins de temps que les autres

PICOLLET R

Expérimentation LearnIT 26/10/2012

Projet SEGAREM

---

8. Quelles sont les caractéristiques d'un bon standard ?

9. A quoi sert la mesure du Lead Time (temps de traversée) ?

Le Lead Time est un indicateur permettant de savoir globalement le temps d'obtention d'un produit.
On cherche à le diminuer.

10. Quels phénomènes sont à l'origine de stocks trop importants ?

- sur-product°
- mon équilibrage des charges
- vitesse de product° plus grande que la demande

Expérimentation LearnIT 26/10/2012

Projet SEGAREM

---

11. Qu'est-ce que la méthode des 5S et à quoi sert-il de l'appliquer sur un poste de travail ? Donnez un exemple pour illustrer vos propos.

5S → permet d'améliorer son poste comme ranger, trier, classer, nettoyer..

ex : poste assemblage

↓

problème de stock de mat. première mal rangé du coup méthode des 5S permet ainsi de le rendre plus performant

Questionnaire d'évaluation de Learnit

Votre nom: .....

Quel est votre statut (Etudiant master / (GI) / Chercheur) : .....
Etes vous : un homme (une femme)
Quel Age avez vous : ..... 21 ans .....
Quelle formation avez vous : ..... INSA Lyon (Premier cycle + GI)

Conception générale du jeu

Pour comprendre les concepts du lean, le jeu vous a semblé *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

Utile ?	1	2	3	(4)	sans avis
Pertinent ?	1	2	(3)	4	sans avis
Amusant ?	1	2	3	(4)	sans avis
Facile?	1	2	3	(4)	sans avis
Avez vous le sentiment d'être plus compétent pour comprendre les concepts généraux du Lean ?	1	2	3	(4)	sans avis
Avez vous le sentiment d'être plus compétent pour appliquer les concepts du Lean en entreprise ?	1	2	3	(4)	sans avis
L'organisation de l'activité vous a-t-elle semblée trop contraignante ?	1	(2)	3	4	sans avis

Avez vous apprécié *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

les phases de simulation	1	2	3	(4)	sans avis
les phases de débriefing en équipe	1	2	(3)	4	sans avis
le fait de pouvoir tester des améliorations sur le poste de travail	1	2	3	(4)	sans avis

Vous êtes vous senti capable de réaliser *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

les phases de simulation	1	2	(3)	4	sans avis
les phases de débriefing en équipe	1	2	(3)	4	sans avis
le fait de pouvoir tester des améliorations sur le poste de travail	1	2	3	(4)	sans avis

Les contacts avec l'enseignant étaient *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

agréables	1	2	3	(4)	sans avis
formateurs	1	2	3	(4)	sans avis
fréquents	1	2	3	(4)	sans avis
nécessaires	1	2	3	(4)	sans avis

Avez vous l'impression d'avoir découvert des domaines *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)*:

La réalité mixte	1	2	3	(4)	sans avis
Les ateliers de production	1	2	(3)	4	sans avis
Les métiers de la production	1	(2)	3	4	sans avis
Les jeux en éducation	1	2	(3)	4	sans avis
Autre :	1	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Vous posiez vous des questions sur ce que pouvait être :

La réalité mixte	1	2	3	**4**	sans avis
Les ateliers de production	1	**2**	3	4	sans avis
Les métiers de la production	1	2	**3**	4	sans avis
Les jeux en éducation	1	2	**3**	4	sans avis
Autre :	1	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Quels éléments n'aviez vous jamais manipulé *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

Des tables tactiles	1	2	3	**4**	sans avis
Des tablettes	**1**	2	3	4	sans avis
Des objets tangibles pour piloter des actions virtuelles	1	2	3	**4**	sans avis
Autre :	1	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Quelles type d'émotion avez vous ressenti *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

Joie,	1	2	3	**4**	sans avis
Plaisir,	1	2	3	**4**	sans avis
Excitation,	1	2	3	**4**	sans avis
Euphorie,	1	**2**	3	4	sans avis
Tristesse,	**1**	2	3	4	sans avis
Dégoût,	**1**	2	3	4	sans avis
Peur,	**1**	2	3	4	sans avis
Colère,	**1**	2	3	4	sans avis
Surprise,	1	2	**3**	4	sans avis
Mépris,	**1**	2	3	4	sans avis
Déception,	**1**	2	3	4	sans avis
Gêne	**1**	2	3	4	sans avis
Perplexité.	**1**	2	3	4	sans avis

Avez vous apprécié les interactions sociales inhérentes au jeu ?	1	2	3	**4**	sans avis
Votre relation aux autres a-t-elle changée	**1**	2	3	4	sans avis
Avez vous l'impression d'avoir gagné en popularité	**1**	2	3	4	sans avis
Avez vous eu l'impression d'avoir bien gardé le contrôle sur ce que vous aviez à faire	1	2	**3**	4	sans avis
Avez vous l'impression d'avoir développé votre potentiel personnel	1	2	3	**4**	sans avis
Avez vous un meilleure estime de vous suite à cette formation	1	2	3	**4**	sans avis
Pensez vous que cette formation peut vous faire gagner de l'argent à moyen ou long terme	1	2	**3**	4	sans avis
Aimeriez vous suivre a nouveau une formation sous cette forme ?	1	2	3	**4**	sans avis

1. Avez vous des conseils pour rendre le jeu plus agréable et efficace ?

Je n'ai pas d'idées car je l'ai trouvé très agréable et très sympa à utiliser

Les tests / questionnaires pourraient être remplis sur tablette !

Votre poste de travail

2. Quel était **votre premier** poste de travail : ..... Assemblage

3. Concernant **votre premier** poste de travail,
Est ce que l'activité vous a semblé *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

Cohérente ?	1	2	3	(4)	sans avis
Pertinente ?	1	2	3	(4)	sans avis
Représentative de l'activité effective ?	1	2	3	(4)	sans avis
Utile ?	1	2	3	(4)	sans avis
Amusante?	1	2	3	(4)	sans avis
Avez vous apprécié le Design graphique ?	1	2	3	(4)	sans avis
Le fonctionnement des éléments sur votre poste de travail vous a-t-il plu ?	1	2	3	(4)	sans avis
Est ce que les informations présentées étaient claires ?	1	2	(3)	4	sans avis
Est ce que le temps de réaction de l'interface était bon ?	1	2	(3)	4	sans avis

4. Quel était **votre deuxième** poste de travail : ..... Contrôle Qualité .....

5. Si vous avez changé de poste, sur **votre deuxième** poste de travail,
Est ce que l'activité vous a semblé *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

Cohérente ?	1	2	3	(4)	sans avis
Pertinente ?	1	2	3	(4)	sans avis
Représentative de l'activité effective ?	1	2	(3)	4	sans avis
Utile ?	1	2	(3)	4	sans avis
Amusante?	1	2	3	(4)	sans avis
Avez vous apprécié le Design graphique ?	1	2	3	(4)	sans avis
Le fonctionnement des éléments sur votre poste de travail vous a-t-il plu ?	1	2	3	(4)	sans avis
Est ce que les informations présentées étaient claires ?	1	2	(3)	4	sans avis
Est ce que le temps de réaction de l'interface était bon ?	1	2	(3)	4	sans avis

6. Quel était **votre troisième** poste de travail : .....

7. Si vous avez changé de poste, sur **votre troisième** poste de travail,
Est ce que l'activité vous a semblé *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui)* :

Cohérente ?	1	2	3	4	sans avis
Pertinente ?	1	2	3	4	sans avis
Représentative de l'activité effective ?	1	2	3	4	sans avis
Utile ?	1	2	3	4	sans avis
Amusante?	1	2	3	4	sans avis
Avez vous apprécié le Design graphique ?	1	2	3	4	sans avis
Le fonctionnement des éléments sur votre poste de travail vous a-t-il plu ?	1	2	3	4	sans avis
Est ce que les informations présentées étaient claires ?	1	2	3	4	sans avis
Est ce que le temps de réaction de l'interface était bon ?	1	2	3	4	sans avis

8. Les éléments suivants ont-ils provoqué un sentiment de stress *(1 : non, 2 : plutot pas, 3 : oui partiellement; 4 : oui)*:

Faire une erreur dans la production des pièces	1	2	(3)	4	sans avis
Ne pas manipuler correctement les interfaces	(1)	2	3	4	sans avis
Suivre la cadence	1	2	(3)	4	sans avis
Autre :	1	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Est ce que le **déplacement** des éléments était *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui) :*

difficile au début puis j'ai compris comment faire.....	1	2	3	(4)	sans avis
facile a réaliser.....	1	2	(3)	4	sans avis
amusant.....	1	2	3	(4)	sans avis
stressant	1	(2)	3	4	sans avis

Pouvez vous donner des exemples des problèmes que vous avez eu avec les déplacements d'éléments ?

sur la table (contrôle qualité) lorsque je prenais un objet, un ou deux objets venaient également avec celui que j'ai sélectionné

Est ce que la **production** des éléments (moulage, collage/assemblage, imprégnation) était *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui) :*

difficile au début puis j'ai compris comment faire.....	1	2	(3)	4	sans avis
facile a réaliser.....	1	2	(3)	4	sans avis
amusant.....	1	2	3	(4)	sans avis
stressant	1	2	(3)	4	sans avis

Pouvez vous donner des exemples des problèmes que vous avez eu avec la production d'éléments

lors de l'assemblage, je devais faire 2 ou 3 fois l'action "mettre de la colle" avant d'y arriver

Avez vous des propositions d'amélioration pour réaliser
-les déplacements d'éléments

Selon moi, les déplacements des éléments dépendent notamment des supports. J'ai trouvé que c'était bien mais ayant l'habitude d'utiliser un Iphone et un Ipad, je trouve que le déplacement des objets étaient un peu plus dur

- la production des pièces

Est ce que la **le pistolet à colle** était *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui) :*

difficile à utiliser au début puis j'ai compris comment faire.....	(1)	2	3	4	sans avis
facile à utiliser	1	2	3	(4)	sans avis
amusant.....	1	2	3	(4)	sans avis
stressant	(1)	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Si le **pistolet à colle** avait été un objet numérique (manipulable sur l'écran avec les doigts) pensez vous que l'activité aurait été

Plus simple	(1)	2	3	4	sans avis
Plus précis.....	(1)	2	3	4	sans avis
Plus facile	(1)	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Est ce que **la bobine rouge** était *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui) :*

difficile à utiliser au début puis j'ai compris comment faire.....	1	2	(3)	4	sans avis
facile à utiliser	1	2	(3)	4	sans avis
amusant.....	1	2	3	(4)	sans avis
stressant	(1)	2	3	4	sans avis
Autre :	1	2	3	4	sans avis

Si la **bobine rouge** avait été un objet numérique (manipulable sur l'écran avec les doigts) pensez vous que l'activité aurait été

Plus simple	1	(2)	3	4	sans avis
Plus précis.....	1	(2)	3	4	sans avis
Plus facile	1	(2)	3	4	sans avis
Autre :	1	2	3	4	sans avis

Est ce que la **bobine verte** était *(1 : non, 2 : plutôt pas, 3 : oui partiellement; 4 : oui) :*

difficile à utiliser au début puis j'ai compris comment faire.....	1	2	3	(4)	sans avis
facile à utiliser	1	(2)	3	4	sans avis
amusant.....	1	2	(3)	4	sans avis
stressant	1	(2)	3	4	sans avis
Autre :.....	1	2	3	4	sans avis

Si la **bobine verte** avait été un objet numérique (manipulable sur l'écran avec les doigts) pensez vous que l'activité aurait été

Plus simple	1	2	(3)	4	sans avis
Plus précis.....	1	(2)	3	4	sans avis
Plus facile.....	1	2	(3)	4	sans avis
Autre :	1	2	3	4	sans avis

Liste des figures

Résumé

Conception de learning games en réalité mixte

Mémoire d'Ingénieur C.N.A.M., Lyon 2013

RESUME

La réalisation de Learning Games permettant des situations collaboratives et utilisant des techniques d'interactions hommes-machines avancées impose l'utilisation de méthodes de conceptions et d'outils adéquats pour faciliter leur production notamment de par la complexité de leur contexte social et environnemental. Le présent travail a pour vocation d'appliquer les méthodes et outils issus de l'ingénierie logicielle provenant tant du domaine du jeu vidéo que du domaine des interfaces homme-machine pour fournir un environnement de conception et d'exécution de jeux pédagogiques collaboratifs en réalité mixte. Ainsi nous proposons différents concepts et modèles supportant la conception, que nous projetons sur une architecture de Learning Games collaboratif basée sur une approche multi-agent et en couches, et incluant un mécanisme de distribution et de contextualisation dynamique des zones de travail numériques. Enfin nous implémentons cette architecture dans un cas d'étude : un jeu pédagogique en réalité mixte destiné à l'apprentissage des concepts LEAN.

Mots clés : Apprentissage collaboratif, Jeu pédagogique, Réalité mixte, Contextualisation, IHM, Architecture, Modèle, Conception

---

ABSTRACT

Making Learning Games that allow collaborative situations and use of advanced human-computer interaction techniques requires adequate methods and design tools, especially with regards to the complexity of their social and environmental context. The current work aims at applying software ingeneering methods and tools coming from both video-games and human-computer interfaces domains in order to provide a mixed-reality and collaborative learning games design and runtime environment. We then propose various concepts and models supporting the design phase, then we apply them on a collaborative Learning Game architecture based on multi-agent and layers approach, and also include a dynamical numerical working zones distribution and contextualisation mecanism. We then finally implement the architecture in a case study : a mixed-reality learning game aimed at learning LEAN concepts.

Key words : Game based Learning, Collaboratif Learning, Mixed Reality, Context-awareness, HCI, Architecture, model, design