

Guillaume SIORAK  
M2 IIC  
Université de Reims



## Développement et portage de jeux vidéo sur iPhone

Rapport de stage

Tuteur : M. Thomas Balitout



DotEmu  
8 passage Brûlon  
75012 PARIS

Stage Avril – Septembre 2010

# Résumé

Ce rapport présente mes travaux effectués dans le cadre de mon stage de Master II Informatique, Ingénierie de l'Image et de la Connaissance (Université de Reims). Il a été effectué dans l'entreprise DotEmu située à Paris 12<sup>ème</sup> (France). Cette société est spécialisée dans l'émulation, le portage et le *repackaging* d'anciens jeux vidéo.

# Remerciements

Je tenais tout d'abord à remercier mon tuteur de stage, Thomas Balitout, pour son aide et son partage d'expérience pendant toute la durée du stage.

Je souhaite également remercier Xavier Liard et Romain Tisserand, les deux fondateurs de la société, pour leur accueil et l'ambiance agréable au sein de leur jeune entreprise.

Enfin, un grand merci à toutes les autres équipes pour leur bonne humeur et leur gentillesse.

# Table des sigles et acronymes

- **Open Source** : la désignation *Open Source* (Code source libre ou Code source accessible) s'applique aux logiciels dont la licence respecte les critères de libre redistribution, de code source disponible et de travaux dérivés possibles, précisément établis par l'Open Source Initiative.
- **API** : Application Programming Interface (ou Interface de programmation) est une interface normalisée permettant à un logiciel de faire appel aux fonctions d'un autre programme déjà disponible sur une machine. Elle constitue une interface servant de fondement à un travail de programmation plus poussé.
- **GDD** : Game Design Document est un document décrivant le fonctionnement complet du jeu ainsi que de son interface.
- **SDK** : Software Development Kit est un ensemble d'outils permettant aux développeurs de créer des applications de type défini (ex pour iPhone ou iPod).
- **SCUMM** : Script Creation Utility for Maniac Mansion est un moteur de jeu doté de nombreuses innovations techniques pour l'époque permettant aisément de réaliser un jeu d'aventure graphique et de le porter assez rapidement sur diverses plates-formes. Il a été développé par Ron Gilbert et Aric Wilmunder (alias SCUMM Lord) en 1987 pour Lucasfilm Games dans les années 1980.

# Sommaire

<b>RESUME.....</b>	<b>2</b>
<b>REMERCIEMENTS .....</b>	<b>3</b>
<b>TABLE DES SIGLES ET ACRONYMES .....</b>	<b>4</b>
<b>INTRODUCTION .....</b>	<b>7</b>
<b>L'ENTREPRISE : DOTEMU .....</b>	<b>8</b>
1 – PRESENTATION .....	8
2 – HISTORIQUE.....	8
3 – ACTIVITES.....	9
4 – MARCHES ACTUELS ET DEVENIR DE L'ENTREPRISE .....	10
<b>LE CONTEXTE DU STAGE .....</b>	<b>11</b>
1 – LES LOCAUX .....	11
2 – LE PARC INFORMATIQUE ET LES PÉRIPHÉRIQUES.....	11
3 – L'AMBIANCE GENERALE .....	12
<b>LE STAGE .....</b>	<b>13</b>
1 – PRISE EN MAIN .....	13
1.1 – <i>Environnement de travail.</i> .....	13
1.2 – <i>Description des outils.</i> .....	13
1.3 – <i>Exercices et mises à jour.</i> .....	15
2 – GOBLIINS TRILOGY IPHONE (PRESENTATION) .....	16
2.1 – <i>Présentation générale.</i> .....	16
2.2 – <i>Présentation des jeux</i> .....	16
3 – GOBLIINS TRILOGY IPHONE (AU PREALABLE) .....	18
3.1 – <i>Conception des GDD.</i> .....	18
3.2 – <i>Etat de l'art.</i> .....	18
4 – GOBLIINS TRILOGY IPHONE (DEVELOPPEMENT) .....	20

---

4.1 – <i>Gameplay</i> .....	20
4.2 – <i>Interface</i> .....	25
4.3 – <i>Apple iPhone OS</i> .....	27
4.4 – <i>Problèmes rencontrés et solutions apportées</i> .....	28
5 – GOBLIINS TRILOGY IPHONE (GESTION DE PROJET).....	30
5.1 – <i>Gestion personnelle du projet</i> .....	30
5.2 – <i>Collaboration</i> .....	30
5.3 – <i>Difficultées rencontrées et solutions apportées</i> .....	31
 <b>CONCLUSION</b> .....	 33
1 – BILAN DU TRAVAIL REALISÉ .....	33
2 – BILAN PERSONNEL.....	34
 <b>BIBLIOGRAPHIE</b> .....	 35
 <b><u>ANNEXE : GAME DESIGN DOCUMENT DE GOBLIINS</u></b> .....	 36

# Introduction

Aujourd’hui, en 2010, le domaine du jeu vidéo est à son apogée. En effet, chaque année, de nouvelles technologies font leur apparition et permettent ainsi l’immersion totale du joueur dans un monde virtuel. D’ailleurs, ils sont de plus en plus beaux et réalistes. On peut ajouter qu’ils rejoignent l’art et plus particulièrement le cinéma.

Les jeux vidéo, bien que jeunes, possèdent tout de même leur histoire. Certes elle ne dépasse pas encore le siècle, mais pourtant, son évolution est nettement visible. Souvenez-vous de ces jeux comme Pong, Pac-man, Space Invaders et bien d’autres. Souvenez-vous des anciennes consoles comme l’Amstrad, l’Atari ST puis la Nintendo NES, la Master System et la MegaDrive. Observez maintenant les nouvelles technologies et les consoles nouvelles générations comme la Wii et la Playstation 3.

Des années se sont écoulées, et pourtant, il existe encore une communauté de joueurs et de développeurs, dont la seule et unique mission est de faire durer dans l’histoire ces anciens jeux. Une d’entre elles se prénomme **DotEmu**, dont je détaillerai plus l’activité dans les rubriques suivantes. **DotEmu** est l’entreprise dans laquelle j’ai effectué mon stage de fin d’étude de Master 2 Informatique. Elle remet au goût du jour d’anciens jeux vidéo (appelés jeux rétro), afin qu’ils ne sombrent pas dans l’oubli.

Dans un premier temps, je présenterai cette entreprise ainsi que l’environnement de travail. Puis, je commencerai par détailler les outils qui ont été mis à ma disposition afin de créer des applications sur iPhone. J’expliquerai par la suite la conception des jeux Gobliins en partant de la réflexion papier au développement. Ensuite, j’analyserai ma gestion de projet durant mon stage. Enfin, je concluerai ce rapport par un bilan du travail réalisé ainsi que par mon bilan personnel en ce début de vie professionnelle.

# L'entreprise : DotEmu

## 1 – Présentation

**DotEmu** est une *start-up* fondée en février 2007 par Xavier Liard et Romain Tisserand, deux ingénieurs diplômés de l'Université de Technologie de Compiègne. L'activité principale de cette entreprise est de remettre au goût du jour d'anciens jeux vidéo. Vous pouvez voir le logo de la société sur la figure 1 ci-dessous.



Figure 1 : « Logo de la société »

## 2 – Historique

En 2007, lors de la création de l'entreprise en février, Xavier Liard et Romain Tisserand, les deux fondateurs de **DotEmu**, se sont lancés sur plusieurs activités. Leur principal corps de métier était à l'époque l'émulation. Durant cette première année, ils développent le site [www.easyretro.com](http://www.easyretro.com), un portail de jeux JAVA qui va ouvrir les portes de leurs premières émulations destinées au web. En septembre, un stagiaire rejoint l'équipe pour développer des prototypes de portage sur *smartphone*.

En février 2008, l'équipe continue de s'aggrandir avec l'arrivée de Thomas Balitout, mon tuteur de stage. L'entreprise change de lieu et s'installe dans les locaux du passage Brûlon dans le 12<sup>ème</sup> arrondissement en avril 2008. Cette année là connaît également l'arrivée d'une nouvelle activité : le *repackaging* de jeux vidéo sur le pôle PC. Ils s'occupent alors du *remastering* pour le compte de Microïds en mai 2008. En juin paraît un des plus gros hits de la société, le jeu Nicky Boom sur *smartphone*. Celui-ci est également adapté sur PC en aout 2008. Puis en novembre, un autre gros titre rejoint son grand frère, celui de Krypton Egg (PC). Enfin, en décembre, Nicky Boom est le premier jeu iPhone porté par l'entreprise disponible sur l'App Store d'Apple.

En 2009, un changement majeur fait son apparition dans les pôles d'activités de **DotEmu**. En effet, à la sortie de Krypton Egg sur iPhone début 2009, l'équipe se rend compte

de la difficulté du marché sur *smartphone*. Elle décide ainsi de concentrer ses forces sur le pôle PC et le pôle iPhone pour la téléphonie mobile. Concernant le pôle web et le portail des jeux JAVA, il évolue en portail de jeux flash. L'équipe connaît son plus gros hit, tout jeu confondu avec le portage du très grand *Street Fighter II* en flash pour le pôle web. Enfin, en septembre, *Golvellius* sort sur iPhone.

En 2010, plusieurs projets ont pu voir le jour. Le premier est la mise en ligne de la boutique **DotEmu** qui permet à l'entreprise d'avoir son propre portail de distribution de jeux vidéo (voir figure 2 ci-contre). D'autres gros projets ont également vu le jour, notamment sur iPhone, avec la sortie prochaine de **R-Type**, un jeu de *shoot* très populaire de la fin des années 80 qui était sorti sur borne d'arcade. De plus, l'entreprise continue de s'agrandir avec l'arrivée d'un 2<sup>ème</sup> local. De gros projets avec certains des plus grands noms du jeu vidéo sont actuellement en cours. On peut donc voir ici l'évolution de cette entreprise dans ce domaine.

Voyons maintenant plus en détails ces activités.



Figure 2 : « Exemple de jeux vendus sur [www.dotemu.com](http://www.dotemu.com) »

### 3 – Activités

L'émulation est l'activité principale de **DotEmu** qui possède un savoir-faire efficace et nécessaire dans sa vie de tous les jours. L'émulation d'un système consiste à faire fonctionner par le biais d'un procédé logiciel (l'émulateur) des applications (ou ROM) provenant d'un système aujourd'hui obsolète sur un système moderne. L'émulateur reproduit ainsi le fonctionnement interne du système émulé et permet d'offrir des solutions offrant la même expérience de jeu que sur le système original.

Plusieurs émulateurs ont été développés par l'entreprise et une dizaine de machines sont actuellement émulées, que ce soit des consoles ou bien des ordinateurs des années 80-90. Grâce à cela, des milliers de jeux sont supportés par cette technologie.

L'autre activité reconnue dans le palmarès de **DotEmu** est la solution dite de portage. Le portage informatique consiste à mettre en œuvre un logiciel, une fonctionnalité, voir un système d'exploitation dans un autre environnement que celui d'origine. Ici, dans le

domaine du jeu vidéo, on parle donc de portage quand on adapte un jeu d'un système (en général celui d'origine) à un autre.

Enfin, un des autres services proposés par **DotEmu** est le *repackaging*. Cela consiste à rendre compatible d'anciens jeux, avec les ordinateurs actuels (systèmes d'exploitation XP, Vista, Seven). Ce service permet aussi le changement du comportement du jeu. Par exemple, certains jeux sont prévus pour fonctionner avec le disque présent dans le lecteur. A la demande des éditeurs, cette contrainte peut être levée afin de pouvoir distribuer le jeu en version numérique (en téléchargement sur le Web). La boutique **DotEmu** ([www.dotemu.com](http://www.dotemu.com)) permet ainsi de distribuer tous ces jeux.

## 4 – Marchés actuels et devenir de l'entreprise

**DotEmu**, bien que reprenant d'anciens jeux vidéo, se situe bel et bien dans le marché du jeu vidéo et y devient un acteur de plus en plus important. Leurs activités d'émulation et de portage de vieux jeux sont belles et bien présentes et efficaces avec les technologies employées.

Avec les jeux sur iPhone, **DotEmu** s'inscrit directement dans l'économie du jeu vidéo avec ce marché émergent et grandissant. De nouveaux joueurs apparaissent, mais n'oublions pas les nostalgiques qui aimeraient récupérer leurs anciens jeux favoris sur leur téléphone.

**DotEmu** a depuis peu lancé son portail de téléchargement de jeux. Ainsi, son activité de *repackaging* est en ce moment au cœur de l'entreprise, tandis que les autres projets suivent leur cours.

Une des plus grandes problématiques de l'entreprise subsiste dans l'obtention des droits d'exploitation de jeux. Par conséquent, des négociations avec d'éventuels éditeurs, ayants droits sont souvent en cours. En plus de tout cela, **DotEmu** continue ses recherches en R&D afin d'améliorer ses technologies pour offrir des remises à niveaux de ces anciens jeux, dans le but de rester le plus fidèle possible aux originaux.

**DotEmu** est également très présent à l'étranger. La société profite des salons du jeu vidéo dans le monde entier pour négocier des droits ou pour chercher de nouveaux clients. (E3 2010 de Los Angeles, voyage au Japon)

En résumé, **DotEmu** s'inscrit aujourd'hui parfaitement dans le marché du jeu vidéo et offre un large panel d'activités avec ses trois pôles majeurs. Sa réputation devient de plus en plus connue à l'étranger et donne ainsi à l'entreprise une bonne vague de prospérité pour les années à venir.

# Le contexte du stage

## 1 – Les locaux

Lors de mon arrivée dans l'entreprise, **DotEmu** ne possédait qu'un seul bureau. Cependant, avec l'arrivée des nouveaux stagiaires, il fallait trouver de la place pour accueillir tout le monde. Un deuxième bureau (situé dans le même couloir) a donc été aménagé à cette occasion. Ceux-ci ont été décorés tout au long de mon stage afin de les habiller à l'image de l'entreprise en vue notamment de réunions avec de nouveaux clients. (voir figure 3 et 4 ci-dessous)



Figure 3 & 4 : « Mon bureau (à gauche) et le 2<sup>ème</sup> local avec la borne d'arcade (à droite) »

## 2 – Le parc informatique et les périphériques

Voyons maintenant le matériel mis à disposition. Pour ma part, faisant du développement sur iPhone, un iPod Touch était mis à ma disposition. Cependant, possèdant également un iPhone personnel, je me suis par la suite servi du mien pour pouvoir tester les applications développées. Concernant le système d'exploitation, Apple ne permet pas de développer les applications iPhone sur un autre système que Mac OS. Ainsi, j'avais également à ma disposition un Mac Mini avec le système d'exploitation approprié pour le développement d'applications iPhone. (voir figure 5 ci-dessous)



Figure 5 : « Mon bureau avec les outils mis à ma disposition (iPod, Mac mini...) »

### 3 – L'ambiance générale

Concernant l'ambiance générale dans l'entreprise, celle-ci a été excellente tout au long du stage. En effet, s'agissant d'une jeune *start-up*, la moyenne d'âge doit être aux alentours des 24 ans. Cette jeune ambiance à laquelle s'ajoute l'humour de ses divers protagonistes a pu ainsi me permettre de m'épanouir pleinement dans cet environnement de travail. De plus, un vrai partage d'expérience et une réelle communication étaient présents dans cette société. En effet, il m'est arrivé plusieurs fois pendant le stage de demander des avis, sur des aspects des jeux développés. Les collègues prenaient alors le temps de me répondre et une vraie discussion débutait me permettant d'avancer par la suite.

De plus, ne connaissant que très peu le domaine de l'émulation lors de mon entrée dans l'entreprise, j'ai pu discuter avec certains collègues de ces techniques et ainsi en apprendre un peu plus avec eux.

# Le stage

## 1 – Prise en main

## 1.1 – Environnement de travail

Comme signalé auparavant, mon stage ayant porté sur le développement de jeux sur iPhone, mon environnement de travail était le système d'exploitation MAC OS X d'Apple. Possédant un MacBook pour mon utilisation personnelle et ayant pu porter quelques projets de l'Université sur Mac OS, cet environnement de travail ne m'était pas totalement inconnu. Cependant, j'ai pu me familiariser un peu plus avec les outils mis à disposition pour le développement sur iPhone. En effet, n'ayant jamais travaillé sur ce périphérique pendant mes études, je ne connaissais pas du tout les subtilités et le fonctionnement du développement sur celui-ci. Vous trouverez une image de mon environnement de travail sous MAC OS, sur la figure 6 suivante.

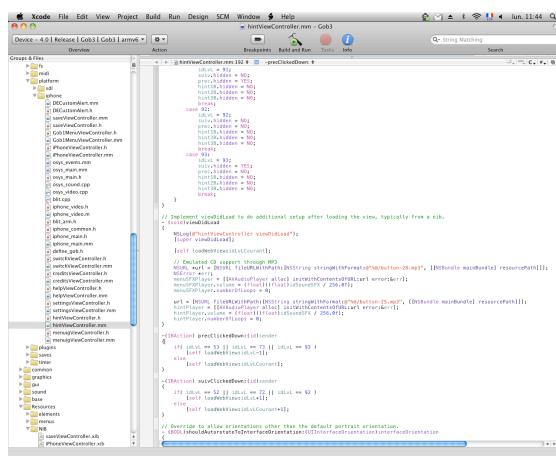


Figure 6 : « Environnement de programmation : Xcode sous MAC OS X »

## 1.2 – Description des outils

Tout d'abord, le premier outil indispensable pour programmer sur iPhone est Xcode. C'est un environnement de développement pour Mac OS, possédant deux principales API de programmation : Carbon et Cocoa. Carbon permet l'utilisation des langages C/C++. Cocoa, lui, permet d'utiliser de l'AppleScript, du Java et de l'Objective-C. C'est d'ailleurs ce dernier langage qui est le plus utilisé pour le développement sur iPhone. J'ai ainsi dû, pendant le

stage, me familiariser avec la syntaxe et le fonctionnement de ces outils. Vous pourrez voir une image représentant *Interface Builder*, l'outil de développement d'interface graphique intégré à *Xcode* sur la figure ci-après (figure 7).

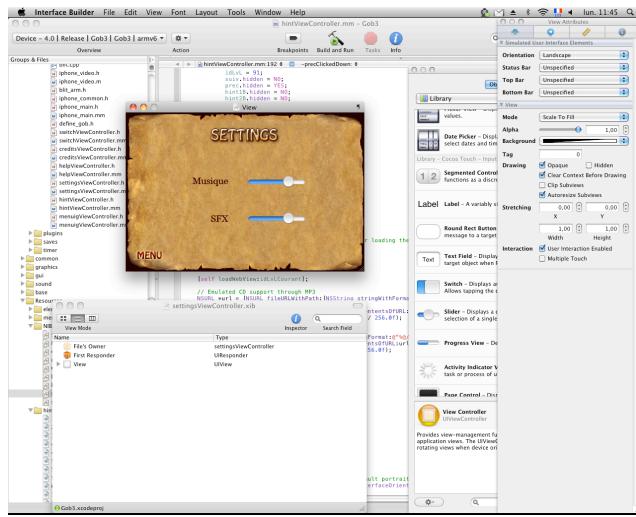


Figure 7 : « Outil pour les interfaces graphiques : Interface builder »

Lors du début de mon projet principal, à savoir le portage et l'adaptation sur iPhone des jeux vidéo Gobliins (1,2,3), j'ai tout d'abord dû comprendre comment était généré le jeu. En effet, lorsque l'entreprise négocie les droits pour des anciens jeux aux ayants droit, elle ne récupère en général que les binaires de ces jeux. De ce fait, certains jeux sont tellement anciens que les codes source ont pu être perdu. Cependant, les techniques d'émulation et les machines virtuelles nous permettent de faire fonctionner le jeu d'origine. Il suffit ensuite de le porter sur le nouveau système, ici l'iPhone.

Lors de mon arrivée dans l'entreprise, l'outil ScummVM a été mis à ma disposition pour développer les versions iPhone des Gobliins. ScummVM est une machine virtuelle qui permet de jouer aux jeux utilisant le système **SCUMM**. ScummVM est un logiciel libre sous la licence **GNU GPL**. Il prend en charge divers systèmes d'exploitation, dont l'iPhone. Les jeux Gobliins utilisant le système SCUMM, ScummVM permet de générer le jeu sur un iPhone. Je reviendrai plus en détails sur ma mission lors du portage de ces jeux et mon rôle dans l'utilisation de ScummVM dans une partie suivante.

Voyons maintenant les premières missions qui m'ont été données à mon arrivée ainsi que la prise en main du système.

### 1.3 – Exercices et mises à jour

Ne connaissant pas le langage de programmation Objective-C au début de mon stage, mon tuteur m'avait concocté quelques exercices pour me mettre en condition. En effet, je devais créer de simples interfaces permettant de me familiariser avec l'API d'Apple. Sa documentation étant très complète et possédant de nombreux exemples, j'ai pu me familiariser assez rapidement avec cet outil. Ces exercices, représentés par la figure 8 ci-après, étaient de simples formalités qui se sont avérées essentielles par la suite.

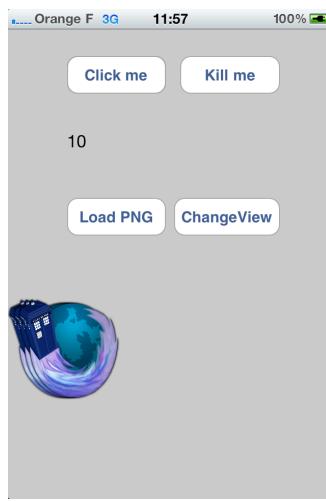


Figure 8 : « Exemple d'un exercice m'ayant permis de prendre en main les divers outils »

La deuxième mission que j'ai dû effectuer lors de ma première semaine était la mise à jour de deux jeux iPhone : *Nicky Boom* et *Nicky Boom 2*. Pour ceux-là, l'entreprise utilisait à l'époque un *middleware* nommé *Bit Rabbit*. Ma mission ne consistant qu'à rajouter de simples boutons pour sélectionner un deuxième *pad* de contrôle, je n'ai pas eu besoin de coder en Objective-C. Cependant, c'était très intéressant de voir les retours des utilisateurs concernant l'utilisation du *pad*. Certains préféraient l'ancienne version et d'autres la nouvelle. J'ai ainsi ajouté les deux avec un bouton de sélection dans le menu des options. (voir figure 9)



Figure 9 : « Illustration du pad sur le jeu Nicky Boom sur iPhone »

Lors de ma deuxième semaine de stage, je me suis attaqué à un gros projet, le portage des jeux vidéo des Gobliiins sur iPhone. Voyons maintenant tout cela plus en détails.

## 2 – Gobliiins Trilogy iPhone (Présentation)

### 2.1 – Présentation générale

Après avoir acquis les droits pour *repackager* les jeux Gobliiins sur PC, **DotEmu** a voulu se lancer dans le portage de ces jeux sur iPhone. En effet, l’iPhone étant un pôle important de la société, il aurait été dommage que les Gobliiins n’y figurent pas, *Puzzle games* emblématiques des années 90.

Avec l’utilisation de ScummVM pour faire tourner le jeu sur iPhone et après vérification et correction de ce moteur, l’adaptation de ces jeux pouvait commencer.

Au préalable, nous parlions de créer une seule application iPhone qui aurait été nommée *Gobliiins Trilogy*. Après réflexion et à cause de certaines contraintes techniques, nous avons décidé de sortir 3 applications différentes. Un *pack* contenant les 3 jeux existera certainement lors de la sortie de ces derniers sur l’App Store.

### 2.2 – Présentation des jeux

Concernant les jeux Gobliiins, ceux-ci sont des jeux de type *Puzzle/Adventure Games*. En effet, dans la lignée des *Chevaliers de Baphomet (Broken Sword)* et des *Monkey Island*, les Gobliiins sont des jeux *Point & Click* ayant vu le jour dans les années 90 sur *DOS*.

Dans *Gobliiins*, vous dirigez une équipe de trois petits gobliiins (d'où les trois « i » du nom du jeu), devant résoudre les énigmes de chaque scène, l'une après l'autre, afin de sauver le roi et leur royaume des mains d'un horrible sorcier. Ils partent alors en quête d'objets afin d'anéantir le mauvais sort. Ce jeu étant le premier de la lignée, paru en 1991, celui-ci est le moins connu des trois. Sa difficulté est d'ailleurs très grande, comme la plupart des jeux d'époque.

Dans *Goblins 2 : The Prince Buffoon*, vous dirigez cette fois une équipe de deux gobliins. A leur tour, ces derniers devront retrouver le prince Buffoon (personnage éponyme du jeu) ayant été capturé par un vilain démon. Ce jeu est paru en 1992. Sa difficulté, très élevée également, est tout de même plus facile d'accès que celle de son ancêtre. Enfin, le jeu possède une bande son très comique qui vous accompagnera tout au long de la quête.

Pour conclure, dans *Goblins 3*, vous dirigez un seul goblin qui s'armera d'une équipe de personnages rencontrés sur le chemin pour tenter de résoudre le conflit entre le roi Bodd et Université de Reims

la reine Xina. Ce jeu, paru en 1993, est le plus populaire de ces trois jeux. Sa difficulté est semblable à celle du deuxième volet, s'agissant du même moteur de jeu. Vous pouvez voir la pochette d'origine des trois jeux sur la figure 10 suivante.



Figure 10 : « Pochettes des trois jeux Gobliins d'origine »

Voyons maintenant plus en détails le début du projet, de la réflexion à la conception.

## 3 – Gobliiins Trilogy iPhone (Au préalable)

### 3.1 – Conception des GDD

Avant la conception proprement dite des applications, je suis passé par une étape appelée le **GDD**. Le **GDD** (*Game Design Document*) est un document décrivant le fonctionnement complet du jeu ainsi que de son interface. Celui-ci aboutit à la vision finale du projet par les différents corps de métier comme le développeur et le graphiste.

Au début de la création de l'application Gobliiins, j'ai donc effectué un premier jet de ce GDD. J'ai tout d'abord réalisé un *powerpoint*, présenté aux membres de l'équipe, afin d'exprimer les idées que j'avais de l'application. Par la suite, un simple fichier .doc a été créé pour résumer toutes les idées à la création de cette application.

On trouve dans ce document les informations primordiales liées au jeu, à savoir son nom, son genre, son type, sa durée de vie, sa difficulté, son public. Puis, j'ai par la suite détaillé la jouabilité (*gameplay*) de ce jeu sur iPhone. Un premier test de l'interface a également été renseigné avec l'aide d'un storyboard et d'un diagramme de flux représentant les actions du joueur sur l'interface et le jeu.

Bien entendu, ces documents ont été mis à jour tout au long de mon stage afin d'avoir une version complète et propre des idées et du visuel utilisés dans les jeux. A la fin de ce rapport, vous trouverez le GDD qui a été effectué pour Gobliiins en annexe.

### 3.2 – Etat de l'art

Une des premières étapes, en plus de la création des GDD, a été de se renseigner sur les jeux existants. Une recherche de l'existant a donc été effectuée. Les *Monkey Island* étant disponibles sur iPhone ainsi que le premier des *Chevaliers de Baphomet (Broken Sword)*, nous avons pu tirer certaines conclusions de ces derniers pour les Gobliiins.

La première recherche s'est effectuée en rapport avec la difficulté. En effet, comme énoncé précédemment, les jeux Gobliiins sont des jeux très difficiles. Il fallait donc trouver un moyen de simplifier ce jeu le plus possible.

Lors de la sortie des jeux d'origine, un manuel était disponible dans le coffret. Celui-ci comportait un *hint book*. Ce document est en fait le guide des astuces des jeux afin d'aider le joueur à terminer les quêtes. Nous avons repris ce principe et nous l'avons implanté dans les Université de Reims

jeux iPhone afin que le joueur puisse à tout moment avoir des indications sur la marche à suivre pour résoudre telle ou telle énigme.

Une deuxième recherche intéressante et importante de l'existant s'est portée sur les *hotspots*. Un *hotspot* est en fait un « point chaud » ou « point clé » présentant les éléments du jeu sur lesquels le joueur peut interagir. Ces éléments sont souvent connus des jeux de type *Point & Click* car ils permettent de donner des indications en temps réel au joueur, notamment lorsque celui-ci se trouve bloqué en pleine énigme. Ainsi, nous avons également ajouté cette fonctionnalité aux jeux Gobliiins pour trouver un moyen de contrer cette difficulté trop importante.

Enfin, une dernière recherche a été faite sur le mode de contrôle de ce type de jeu sur iPhone. En effet, s'agissant d'un périphérique tactile, et le *gameplay* de base de ce genre de jeu étant l'exploration et le déplacement, il aurait été dommage de ne pas utiliser le touché du doigt pour effectuer une action. Cependant, nous verrons plus tard que le système de curseur n'est pas à proscrire pour ces jeux, même sur une machine tactile.

Voyons maintenant le développement proprement dit effectué sur le portage de ces jeux et sur la création des applications iPhone.

## 4 – Gobliiins Trilogy iPhone (Développement)

Durant cette partie, j'essaierai de vous présenter au mieux le travail que j'ai effectué pendant mon stage. J'agrémenterai les notions abordées par des exemples.

Comme déjà mentionné précédemment, le développement des applications iPhone pour les jeux Gobliiins s'est effectué à plusieurs niveaux. Tout d'abord, il y a eu une gestion préalable de la configuration de ScummVM pour faire tourner les jeux sur ce périphérique. Ensuite, il a fallu créer les applications, à savoir la conception des menus du jeu, les liens entre le jeu et ces menus, puis bien entendu, réfléchir et implémenter une jouabilité particulière (*gameplay*) pour ce type de jeux sur iPhone.

Je commencerai à traiter de la partie *gameplay* concernant les trois jeux, puis nous détaillerons un peu plus les interfaces avant d'expliquer les mécanismes propres à l'iPhone qui ont été intégrés ainsi que les problèmes rencontrés.

### 4.1 – Gameplay

Dans cette partie *gameplay*, nous différencierons plusieurs points. Je commencerai par la gestion du jeu proprement dite, à savoir le déplacement des personnages, comment effectuer une action avec ceux-ci, etc. J'expliquerai par la suite les différentes aides intégrées pour aider les joueurs pendant leur quête.

#### 4.1.1 – Comment jouer ?

Comme il s'agissait de jeux de type *Point & Click* à l'époque, la souris était utilisée alors qu'aujourd'hui, sur un iPhone, il n'y a pas de souris. Cependant, s'agissant d'un périphérique tactile, le doigt la remplace très bien.

Après discussion, je me suis donc lancé pour un mode de contrôle nommé *touch mode*, c'est-à-dire un mode de contrôle de jeu au touché. Pour cela, l'API d'iPhone étant bien faite, des mécanismes de base étaient disponibles. Il fallait donc maintenant les adapter aux jeux.

Dans Gobliiins, nous pouvons contrôler trois personnages différents. Ces derniers possèdent également des actions à effectuer. Certaines sont communes, comme le déplacement, et d'autres sont différentes. Ainsi avec l'iPhone, sans souris, donc sans curseur, il ne restait plus qu'à détecter les coordonnées du doigt sur l'écran pour faire se déplacer le personnage à l'endroit indiqué. Pour les autres actions, comme utiliser un objet pour le gobliiins technicien, combattre pour le gobliiins guerrier et lancer un sort pour le

magicien, j'ai décidé de les représenter par des boutons. Le joueur n'a donc plus qu'à « toucher » le bouton correspondant à l'action pour que le gobliiin l'effectue. Vous pouvez voir une illustration de ces actions sur les figures 11 a,b et c ci-contre.

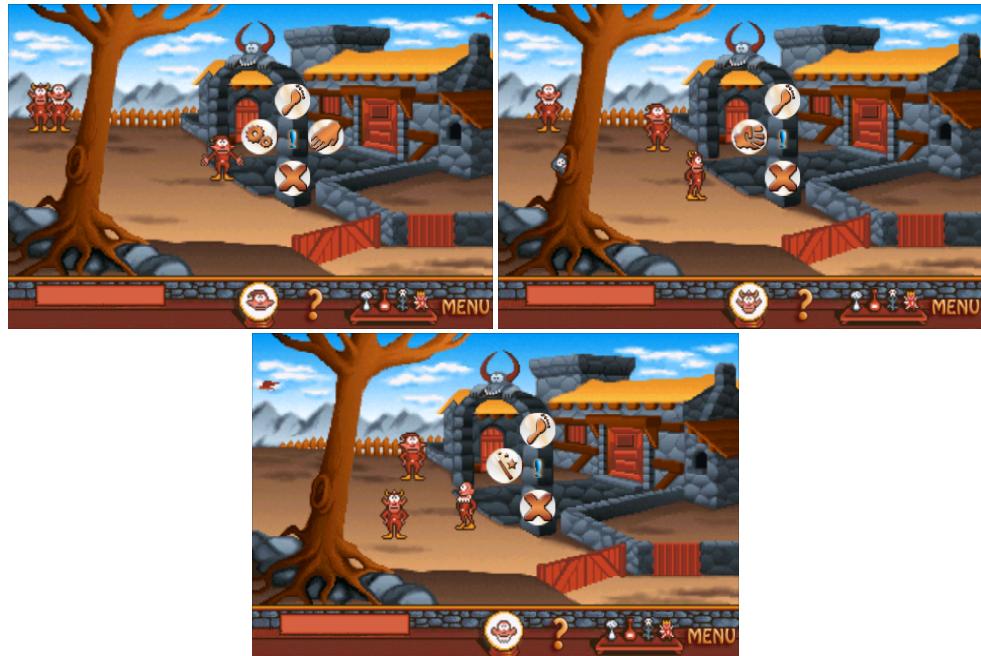


Figure 11 a,b et c : « Illustrations représentant les actions de chaque gobliiin. En haut à gauche, actions du technicien. En haut à droite, actions du guerrier. En bas, action du mage »

Mais s'agissant de boutons, où va donc s'effectuer l'action ? C'est là que les *hotspots* (points chauds ou points clés) entrent en jeu. Je détaillerai ces éléments par la suite. Mais, lorsque le joueur clique sur le bouton d'action pour un *hotspot* donné, les actions s'effectueront à l'endroit où se situent les *hotspots*. Si par contre celui-ci relâche son doigt sur une zone libre (ne contenant donc pas de *hotspot*), la même liste d'action sera alors proposée au joueur. Il pourra donc effectuer l'une d'entre elles dans cette zone vide. Vous pouvez voir l'explication en images sur les figures 12 a et b ci-contre.



Figure 12 a et b : « A gauche, le point d'exclamation représente un hotspot et les actions pouvant être effectuées dessus. A droite, une zone vide, sans hotspot où la liste des actions apparaît après un certain temps d'appui sur l'écran »

Bien qu'ayant fait au plus simple dès le démarrage du projet (un simple touché de doigt pour se déplacer, une liste d'actions sur un point clé ou sur une zone libre...), le jeu Gobliiins étant très compliqué, nous avons réalisé quelques semaines plus tard que ce mode de jeu ne suffirait pas. Nous avons alors réfléchi à un mode de contrôle alternatif, basé sur notre recherche de l'existant. Il s'agit du mode curseur.

Le mode curseur reprend un peu le principe des jeux *Monkey Island* sur iPhone. En effet, ces derniers possèdent un curseur qui change en fonction des actions que le joueur choisit d'effectuer dans une liste.

Ici, les jeux Gobliiins possèdent tellement d'objets et de petits détails qu'il fallait aussi prévoir de viser un endroit inaccessible de la scène. C'est à cela que sert le mode « curseur ». Il permet d'ajouter un élément visuel de position au joueur, une extension du doigt à l'échelle des gobliiins. La gestion du curseur est donc effectuée comme un *trackpad*. Un décalage est calculé en fonction de la position du doigt sur l'écran, pour que le joueur puisse voir où positionner son curseur. Dans *Gobliiins*, le curseur possède également la tête du personnage sélectionné (voir figure 13 ci-après).



Figure 13 : « Illustration représentant le curseur du jeu comportant la tête du gobliiins actuellement joué »

Ces deux modes de contrôle intuitifs pour Gobliiins permettent une gestion complète de la souris d'antan sur ce type de jeu.

Concernant *Gobliiins 2* et *Goblins 3*, utilisant un même moteur de jeu et un même fonctionnement, un seul *gameplay* commun y a été intégré. Il s'agit tout simplement d'un mix entre le mode *touch* et le mode curseur de leur grand frère.

En effet, nous possédons ici un curseur qui reste présent dans la scène. Une nouvelle fois, dans ces deux nouveaux jeux, certains détails sont difficiles d'accès. Il fallait donc permettre au joueur de les atteindre. Concernant le système de jeu, celui-ci est complètement différent du premier volet. Dans *Gobliiins 2*, nous contrôlons deux personnages. Ceux-ci ne possèdent pas d'actions différentes. Ils ont tout deux un même

inventaire contenant des objets. Ils peuvent sélectionner un objet différent et l'utiliser en même temps. Pour le 3<sup>ème</sup>, il existe un seul personnage principal et des personnages secondaires qui vont avoir le même fonctionnement que nos acolytes de *Gobliins 2*.

Ici, le curseur, en plus d'indiquer précisément l'endroit où se jouera l'action, possède dans une petite bulle une information supplémentaire. S'il s'agit d'un endroit où le gobliins peut se déplacer, le curseur possèdera des traces de pas. S'il s'agit d'un *hotspot*, celui-ci possèdera un point d'exclamation, signalant ce dernier. Enfin, si le gobliins tient un objet dans sa main (donc non rangé dans l'inventaire), le curseur possèdera un sac à dos ouvert, signalant l'utilisation de l'objet sur la zone pointée. Vous pouvez voir un exemple de ces curseurs sur les figures 14 a, b et c suivantes.



Figure 14 a, b et c : « En haut à gauche, curseur possédant une trace de pas, pour le déplacement (action par défaut). En haut à droite, un point d'exclamation représente qu'un objet est utilisable (hotspot). En bas, le sac ouvert montre l'utilisation d'un objet sur la zone pointée en cas de validation »

Ainsi, dans ces deux jeux, il n'y a pas vraiment d'action différente entre les personnages. Il n'y en a qu'une seule, se déplacer ou utiliser un objet (si l'objet est placé dans la main). C'est ici que le mode *touch* intervient. Si le joueur touche un endroit de la scène avec son doigt et que le curseur est en mode « trace de pas » (soit qu'aucun objet de l'inventaire n'a été sélectionné), le personnage se déplacera à cet endroit. Si par contre, le sac à dos est ouvert, le personnage utilisera l'objet placé dans sa main à l'endroit du curseur. Le joueur peut également décider de viser un endroit précis de la carte. Pour cela, il suffit comme dans *Gobliins*, d'utiliser le curseur comme un *trackpad* en laissant son doigt enfoncé et en glissant

ce dernier vers l'endroit voulu. Par la suite, un simple touché n'importe où sur l'écran lancera l'action (déplacement/utilisation).

Voyons maintenant la gestion des *hotspots* dans les jeux.

#### 4.1.2 – Qu'est-ce qu'un *hotspot* ?

Comme déjà annoncé auparavant, un *hotspot* est un endroit clé du jeu. Il s'agit en général d'une zone de la scène du jeu sur laquelle le joueur peut interagir. La plupart du temps, il s'agit d'un objet que le gobliiins peut ramasser. D'autres fois, et notamment dans *Gobliiins*, il peut s'agir de simples zones interactives comportant des animations comme une porte qui s'ouvre.

Ses indications permettent d'aider le joueur afin de savoir quelles zones de la scène sont susceptibles d'être utilisables. Ces aides n'existaient pas toujours dans les jeux d'origine, et n'étaient pas très visuelles. Dans *Gobliiins*, des points d'exclamation bleus apparaissent à ces endroits. Dans *Gobliiins 2 et 3*, il s'agit de simples petits ronds, le nombre de zones utilisables étant beaucoup plus grand. Vous pourrez voir une illustration de ces *hotspots* sur les figures 15 a et b jointes.



Figure 15 a, b : « A gauche, vous pouvez voir les *hotspots* sur *Gobliiins* représentés par des points d'exclamation. A droite, sur *Gobliiins 2*, les *hotspots* sont représentés par des cercles »

Dans *Gobliiins*, le joueur peut faire apparaître ces *hotspots* de plusieurs façons. Avec le *gameplay touch*, c'est en glissant son doigt sur l'écran près d'un *hotspot* que celui-ci sera affiché avec un point d'exclamation. En mode curseur, c'est en glissant le curseur près d'un de ces endroits. Dans le 2<sup>ème</sup> et 3<sup>ème</sup> volet, c'est également en glissant le curseur dans les zones d'*hotspots* que ceux-ci apparaîtront. Un point d'exclamation dans le curseur sera également disponible. Le joueur n'a alors plus qu'à effectuer son action. Voyons maintenant les autres aides pour les joueurs.

#### 4.1.3 – Des aides aux joueurs ?

Dans les jeux Gobliiins, la difficulté élevée est, de plus, croissante. Les créateurs avaient pour cela fait paraître, en plus des jeux, des manuels contenant les astuces de chaque scène afin d'aider le joueur à les résoudre.

Etant dans une ère entièrement numérique, même pour les jeux vidéos, ces informations ont été intégrées directement dans le jeu. Elles se trouvent dans le menu *hint* (astuce), représenté par un point d'interrogation. Dans ce menu, et en fonction de la scène jouée, le joueur trouvera les étapes à effectuer en plusieurs points afin d'atteindre son but. Un exemple de cette aide est disponible sur la figure 16 suivante. Voyons maintenant la partie concernant les interfaces des jeux.

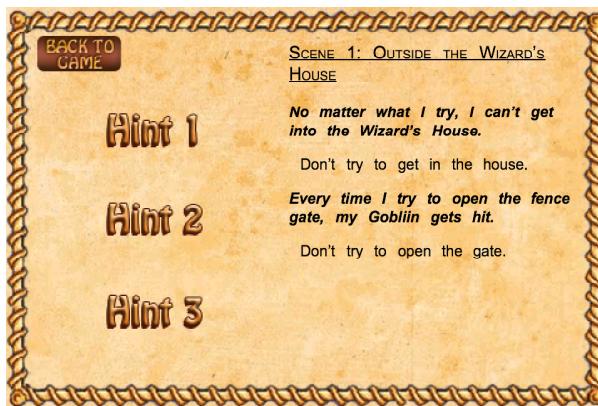


Figure 16 : « *Menu des astuces (hint book) consultable pendant la partie* »

### 4.2 – Interface

Qui dit jeu vidéo, dit menu. En effet, il faut pouvoir accéder à différents éléments. Le chargement d'une partie, le commencement d'une nouvelle, la gestion des options, les indications et aides etc. Voyons tout cela sans plus tarder.

#### 4.2.1 – Les menus

La gestion des menus et de l'interface en général, n'a pas été des plus aisée. L'iPhone et son **SDK** permettent une gestion de vue/contrôleurs. Cependant, des mécanismes et des outils de base sont déjà implémentés. Ils sont accessibles avec l'aide de *l'interface builder*, un outil interactif, fonctionnant comme le Qt Designer et bien d'autres, en glisser/déposer. Il est donc très rapide de créer un simple menu avec une zone de texte et trois boutons. Il faut juste savoir où aller et de quoi avoir besoin.

Vous pouvez à tout moment déclarer dans votre code source un objet comme étant un *IBOutlet*. En renseignant celui-ci par l'objet ajouté dans *l'interface builder*, vous aurez un lien sur ce même objet dans votre code source. Vous pourrez donc à tout moment le modifier.

Concernant les menus de *Gobliiins*, ces derniers ne sont pas très compliqués. Un écran permet de charger une partie. Dans le premier volet, la différence est que tant qu'un niveau n'a pas été débloqué, celui-ci n'est pas sélectionnable. Ainsi, on ne pourra pas charger n'importe quelle scène. Dans le 2 et 3, il s'agit de simples sauvegardes. L'écran de chargement du menu principal permet donc de charger une partie précédente.

Le menu des options permet de gérer le son de la musique et des bruitages. Pour *Gobliiins*, un bouton sélectionnant le mode de contrôle est également présent. Le menu d'aide correspond aux indications décrivant au mieux le jeu pour aider le joueur lors de sa première partie. Un tutoriel a d'ailleurs été créé lors du lancement d'une nouvelle partie, afin toujours, d'aider l'immersion du joueur dans le jeu et de l'aider dans sa quête. Enfin, le menu des crédits et le bouton « *more games* » permettent de donner les informations essentielles du produit. Vous retrouverez l'illustration de ces menus sur les figures de la partie *storyboard* de l'annexe situé à la fin de ce rapport.

#### 4.2.2 – Ingame

Le menu *ingame* est le menu que le joueur trouvera en jeu. Celui-ci est sélectionnable à partir du bouton menu dans l'interface du jeu en lui-même. Ici, plusieurs options sont proposées. La première est la reprise de la partie en cours. La deuxième selon les jeux correspond au relancement du niveau en cours ou de la sauvegarde/chargement d'une scène. Enfin, on peut modifier les options du jeu ou quitter et revenir au menu principal.

En effet, il n'y a aucun moyen de quitter un jeu à partir du menu dans une application iPhone. Cela est interdit. Pour les iPhone précédant l'iPhone 4, le bouton *Home* du milieu de l'appareil permettait de quitter le jeu. Depuis l'iPhone 4, celui-ci met le jeu en background. Ce dernier est donc en pause et tourne en fond de tâche, mais n'est pas quitté.

On trouve également en plus du bouton menu, d'autres boutons dans l'interface *ingame*. Le point d'interrogation, comme énoncé dans la rubrique précédente, permet d'afficher le menu des astuces. Dans *Gobliiins*, le bouton du milieu permet de changer de personnage. Tandis que dans le 2 et 3, le sac permet de sélectionner le menu des inventaires. Vous trouverez divers exemples de ces menus dans les figures qui ont déjà pu être présentées précédemment.

#### 4.2.3 – Les transitions et animations

Les transitions entre les menus sont également importantes pour des jeux sur ce type de périphérique. En effet, il faut pouvoir montrer au joueur une certaine dynamique du jeu. Cela est souvent transmis par les animations et les transitions.

Nous avons ainsi travaillé, avec le graphiste de l'équipe, pour créer, dès le lancement du jeu de simples animations. Cela permet ainsi d'attirer l'œil du joueur dans un environnement visuel réussi. La dynamique du jeu est alors donnée, il ne reste plus qu'à lancer le jeu.

Concernant leurs implémentations, de nombreux exemples existent sur le net et dans la documentation du SDK de l'iPhone. Pour les jeux Gobliiins, de simples *fade in/out*, translations et rotations, changement d'images et de composantes couleurs ont été effectués. Les figures 17 a et b ci-contre, bien que non animées sur papier, présentes l'endroit où trouver ces animations.

Je vais maintenant vous expliquer la gestion des mécanismes de base essentiels pour les applications iPhone.



Figure 17 a et b : « Menu principale des deux premiers jeux. Sur Gobliiins, l'animation s'effectue au niveau des nuages. Sur Gobliiins 2, l'animation apparaît dans le titre et sur l'œil droit du personnage »

#### 4.3 – Apple iPhone OS

L'iPhone est un périphérique complet et complexe. On peut faire de la photo, de la vidéo, jouer de la musique, lire ses *emails*, jouer et même téléphoner. Il est donc très important de gérer toutes ces particularités dans les applications développées.

Lors du lancement du jeu *Gobliiins*, si le joueur a la musique de son appareil en marche, un *popup* apparaîtra demandant au joueur si celui-ci souhaite conserver sa musique ou

l'arrêter pour laisser place à la musique du jeu. Cette alerte était importante à intégrer pour éviter au joueur d'arrêter subitement sa musique si celui-ci préfère la conserver.

Les autres contraintes étaient liées à la gestion de sauvegarde et reprise du jeu. En effet, lorsqu'un joueur reçoit un appel, ou si celui-ci met son iPhone en pause (bouton *sleep mode* de l'appareil), il faut effectuer une action. La première est de sauvegarder les informations nécessaires à la reprise future du jeu. Dans le cas des iPhone précédant la version 4, le jeu se quittait généralement. Une reprise de la partie en cours était alors proposée au joueur lors du redémarrage. Cependant, depuis l'iPhone 4, la gestion de *background* est apparue. L'application ne s'arrête donc pas, mais elle est mise en pause. Il faut donc toujours sauvegarder les informations essentielles en cas de sortie du mode jeu. Cependant, lors de sa reprise, il faut s'assurer de reprendre les bons mécanismes du jeu, sans que celui-ci ne s'arrête subitement.

En plus du jeu à implémenter et adapter, il fallait donc faire attention à l'utilisation de ce matériel. Des mécanismes utiles étaient à intégrer pour conserver l'essence même de ce périphérique. Vous pouvez voir des exemples de ceux-ci sur les figures de l'annexe disponible en fin de rapport.

#### 4.4 – Problèmes rencontrés et solutions apportées

Je vais maintenant vous présenter les divers problèmes rencontrés et les solutions qui ont donc été apportées.

Le premier thème serait sans nul doute la gestion des différentes versions de l'OS (*Operating System*) de l'iPhone. En effet, chaque utilisateur d'iPhone ne possède pas nécessairement la même version. De plus, ce périphérique en est maintenant arrivé à sa 4<sup>ème</sup> évolution. Il existe l'iPhone 2G, l'iPhone 3G, le 3GS et enfin le 4. Sans parler des iPod Touch 1<sup>ère</sup>, 2<sup>ème</sup> et 3<sup>ème</sup> génération.

Les problèmes liés aux versions ont donc été nombreuses. Avant l'iPhone 4, comme déjà expliqué, le périphérique fermait l'application avec l'appui sur le bouton *Home*. Depuis le 4, l'application est mise en *stand-by*. Il a fallu donc gérer la gestion de sauvegarde des informations différemment. Il fallait également prévoir dans le code, quoi faire pour telle ou telle version de l'iPhone, ajoutant donc une analyse et réflexion supplémentaire.

Un autre problème, lié également au matériel, a été la gestion de mémoire. En effet, travaillant sur des plate-formes mobile, la gestion de mémoire est délicate, ces appareils ne disposant pas d'autant de mémoire qu'un ordinateur. Une gestion et une optimisation est donc souvent de rigueur. Ici, la difficulté a été au niveau de la gestion d'apparition de certains

éléments du jeu. Quand le périphérique commence à avoir moins de mémoire, il envoie un signal à l'application appelé *Memory Warning*. Lorsque l'application reçoit ce signal, elle doit commencer à vider les éléments qu'elle n'utilise plus afin d'éviter à l'application de se fermer subitement. Cette gestion doit être faite par le développeur, qui doit donc nettoyer et supprimer correctement les éléments qui ne sont plus utilisés ou affichés en premier plan.

Concernant les menus, une certaine difficulté a été éprouvée à la conception des animations. En effet, il fallait gérer les différents cas d'arrêt et de reprise de ces animations lors des divers signaux reçus résultant d'une action de l'utilisateur (appel, *sleep mode...*). De plus, lié aux problèmes de mémoire, la gestion de nettoyage des menus était importante.

Concernant les jeux proprement dit, la difficulté réelle de ce projet s'est posé sur la question du *gameplay*. En effet, il fallait trouver la technique la plus simple possible et la plus intuitive pour effectuer les actions d'origine du jeu. Les solutions qui ont été proposées me sont apparus comme étant les meilleurs pour ce genre de jeu. L'avenir dira si ce choix a été judicieux ou non. Un autre point difficile de ce projet a été la compréhension et l'utilisation de ScummVM. En effet, le moteur générant le jeu étant vaste et complexe, il m'aura fallu de longues heures de recherche et de lecture de code pour pouvoir trouver où intervenaient les différents appels aux routines du jeu. C'est en traçant ces appels que j'ai pu déterminer par la suite la disposition des *hotspots*, et effectuer les différentes gestions des actions et de sauvegarde/chargement présentes dans le jeu.

## 5 – Gobliiins Trilogy iPhone (Gestion de projet)

### 5.1 – Gestion personnelle du projet

En plus du développement de trois applications de jeux sur iPhone, mon travail durant ce stage a été assez varié. En effet, pendant mon stage, j'étais le seul développeur à travailler sur ce projet. J'ai donc dû prendre mes responsabilités pour le mener à bien en temps et en heure.

Pour cela, je me suis tout d'abord fixé des plannings avec l'aide de *Google Docs*. Cet outil est assez pratique pour de petites entreprises comme **DotEmu**, car il permet de créer et de partager des documents pour ses collègues de manière très rapide.

Je possédais quant à moi mes propres documents pour indiquer mon avancement dans le développement du projet. Il fallait donc faire preuve de rigueur afin d'être efficace et de ne pas prendre de retard. Les notions de *deadlines* sont alors ressorties. Je me posais mes propres *deadlines* personnelles dans le développement comme dans la gestion du projet en général, regroupant tous les corps de métier (graphisme notamment).

En milieu de stage, et voyant les trois jeux évolués à bonne vitesse, mes patrons ont alors décidé de faire publier les jeux par une entreprise tierce, très présente sur le marché de jeux iPhone : Bulkypix. Nous avons alors eu une réunion avec l'un de ses fondateurs pour lui présenter les jeux et discuter de la marche à suivre.

Après divers échanges de courriels, la machine était lancée, et la gestion du projet sur lequel je travaillais déjà pour ne pas prendre de retard s'est intensifié.

### 5.2 – Collaboration

La gestion de projet, en plus de son propre travail, repose sur la collaboration avec les divers protagonistes y participant. Dans un jeu vidéo, les corps de métier sont souvent très nombreux. Ici, étant une petite équipe, le projet ne réunissait que quelques personnes. Il y avait Xavier Liard, qui s'occupait de la partie commerciale et négociation avec les ayant-droits et l'éditeur, les graphistes pour habiller les menus et les divers éléments du jeu et enfin moi au développement et en tant que chef de projet pour récupérer tous les retours et synchroniser l'avancement.

De plus, lorsque l'on crée un jeu ou une application, que celle-ci soit créée de toute pièce ou en partie (portage par exemple), il faut pouvoir avoir des avis sur certains aspects. Ainsi, lorsque je développais le *gameplay* ou certaines animations, je demandais souvent à mes collègues leurs avis. Ceux-ci me permettaient d'avancer et surtout de garder tout le monde dans la boucle, étant un projet de l'entreprise. J'essayais également de présenter les changements de graphismes réalisés par mes collègues pour avoir un retour et ainsi pouvoir trouver le meilleur aspect visuel possible.

Enfin, la gestion de la collaboration avec un organisme extérieur était une première pour moi. En effet, Bulkypix (figure 18) s'occupant de l'édition des jeux, il fallait pouvoir leur fournir divers éléments (traduction, visuels...) pour compléter le jeu et arriver à un produit fini. Cette gestion était donc une responsabilité importante pour l'avenir de ce projet.



Figure 18 : « Logo de l'éditeur des Gobliins : Bulkypix »

### 5.3 – Difficultés rencontrées et solutions apportées

Les difficultés rencontrées lors de cette gestion de projet ont été surtout liées au manque d'expérience. En effet, même si nous apprenons à faire des projets et prendre certaines responsabilités lors de notre cursus, rien n'est comparable à celles que l'on doit prendre en entreprise.

La première notion est celle de *deadline*. Il faut pouvoir jauger et juger de son travail et du temps que l'on devra mettre pour réaliser ou résoudre tel ou tel problème. Il faut donc être plus que rigoureux dans les étapes de conception et de développement pour ne pas être confronté à un problème de priorité par la suite.

Il faut également pouvoir être disponible pour les autres corps de métier et gérer au mieux les éléments dont nous pouvons avoir besoin. Par exemple, concernant le graphisme,

---

il fallait que je puisse gérer à la fois les éléments visuels dont j'avais besoin et leur implémentation pour que les deux puissent correspondre (animation des menus...)

Enfin, la synchronisation de tous les corps de métier avec cette notion de date limite reste la difficulté la plus grande dans la gestion d'un projet, mais je pense m'en être sorti correctement tout au long de celui-ci.

# Conclusion

## 1 – Bilan du travail réalisé

Concernant le travail réalisé, celui-ci a été très varié tout au long de ce stage.

Au début, après avoir pris en main le langage et l'environnement à l'aide de petits exercices, j'ai pu être confronté à un grand projet qui allait me prendre six mois.

La première difficulté a été de maîtriser le code de ScummVM afin de comprendre toutes les finesses de cette machine virtuelle. En effet, sa compréhension était un élément clé de ce projet, puisque c'est à partir de ce moteur que j'ai pu trouver les différents éléments constituant les jeux (gestion des *hotspots*, des actions, des niveaux...).

La deuxième partie de mon travail a été la conception des applications de A à Z. Il a alors fallu faire preuve de créativité et également de simplicité pour tenter de porter ces produits le mieux possible afin de garder l'esprit d'origine. Cela allait de la jouabilité, aux diverses aides proposées aux joueurs, mais passait également par la créativité des graphismes pour les menus et les animations de ces derniers.

Une partie importante de mon travail concernait également les divers retours de mes collègues. En effet, il fallait pouvoir résoudre certains bogues pour obtenir une application finale fonctionnelle. Une difficulté, lors du développement de ces applications, était liée aux différentes versions de l'appareil. En effet, il faut pouvoir prévoir tous les cas d'interactions possibles, et donc ainsi, tous les bogues et problèmes directement ou indirectement associés.

Enfin, mon travail n'a pas été que de développer et créer ses applications. Il a fallu les coordonner, les gérer, collaborer avec les équipes et faire en sorte que son avancement se passe correctement. Ce stage m'aura donc ouvert les yeux sur les responsabilités que l'on peut avoir, que ce soit au niveau du développement comme de la gestion de projet.

## 2 – Bilan personnel

En conclusion, je vais maintenant vous présenter mon bilan personnel à l'issue de ce stage.

Bien que nous ayons été habitué à faire des projets et à effectuer des stages pendant notre cursus, rien n'était comparable à celui-ci. En effet, durant six mois j'ai été confronté à un même projet, et ainsi j'ai pu m'habituer aux techniques utilisées et également m'intégrer au sein de l'équipe. De plus, ayant toujours voulu travailler dans le jeu vidéo, cette première expérience dans ce domaine m'a été très bénéfique.

Le fait de retrouver certains jeux de mon enfance et de pouvoir en discuter avec une équipe de jeunes passionnés a également été une expérience motivante. J'ai ainsi pu travailler avec les principaux corps de métier qui peuvent être présents dans leur conception ou leur adaptation. J'ai également eu de grandes responsabilités pour effectuer ce projet, de son développement à la gestion et à la collaboration des autres équipes pour le mener à bien.

Pour conclure, ce stage m'a donc donné une expérience inestimable sur plusieurs points. En plus de tous ceux cités précédemment, j'ai également pu développer ma curiosité sur le fonctionnement d'une entreprise de ce domaine. Ce stage m'a finalement donné une grande maturité face à ma vie professionnelle future ainsi qu'une grande rigueur pour pouvoir continuer dans cette voie.

# Bibliographie

- La boutique DotEmu : [www.dotemu.com](http://www.dotemu.com)
- Le site Corporate DotEmu : <http://corporate.dotemu.com/>
- Développer Apple SDK documentation : <http://developer.apple.com/iphone>
- La machine virtuelle ScummVM : <http://www.scummvm.org/>
- Site de l'éditeur des Gobliins sur iPhone : [www.bulkypix.com](http://www.bulkypix.com)
- Site d'Abandonware France : <http://www.abandonware-france.org>
- Wikipedia : [www.wikipedia.org](http://www.wikipedia.org)
- App Store sur iPhone (jeux *Monkey Island*, *Broken Sword...*)

## Annexe : Game Design Document de *Gobliiins*

## GDD Gobliins

### Introduction (short & catchy description of the game)

Gobliins is a puzzle / adventure game.

In Gobliins the player controls a team of three different goblins, each with a unique skill. The player has to solve a different puzzle at every level by using and combining the skills of the goblins he has at its disposal.

### Target Audience (who are the person susceptible to buy gobliins)

ESRB/PEGI (9+)

The persons susceptible to buy Gobliins are the players who have already played gobliins years ago and are eager to play it again (nostalgic) and those who love puzzle games.

### Platform (on which platform the game will be available)

iPhone 2G, 3G, 3GS, iPod Touch 1<sup>st</sup>, 2<sup>nd</sup> & 3<sup>rd</sup> Gen + iPad

On the iPad, the display is up scaled to take all of the screen resolution.

### Key features (what are the main features of the game)

- Puzzle
- Hours of gameplay
- Nice graphics & sounds
- Funny characters & situations
- 22 puzzle to solve.
- Resume the game where you left it.
- Start a game from any level unlocked.
- Gameplay specially adapted for the iPhone/iPod/iPad.
- 

### Story Line / Synopsis (what the game is about)

A voodoo doll has turned King Angoulafre mad. You control a team of three Goblins - Oups, Ignatius and Asgard (Hooter, Dwayne, and BoBo in the US release), who must find four magic components to track down Niak the Wizard, who is the only one who can put things right. They all have different skills - one can cast powerful but erratic spells, another can collect and use objects, and the last can hit things. You switch between them to solve puzzles in this point-and-click game.

### Objectives (goal of the game)

Complete each level to find the magic elements needed to cure the king.

### Game mechanics (rules of the game)

The games mix elements of adventure and puzzle gaming. In essence, the player must find the solution to each area (or level) in order to progress to the next.

## Game elements (what are the elements of the game)

The game takes in 22 landscapes/levels, each of which has puzzles that must be cleared. Your characters have limited energy, which can be sapped by traps.

The first goblin, Asgard /BoBo, is a warrior who is more brawn than brain, and whose skill is punching, causing various things to break or fall over. He is also the only one who can climb other objects than stairs and ladders.

The second, Ignatius/Hooter, is a magician who can 'zap' things with magic with a wide variety of usually unexpected effects, which include objects moving, growing, or becoming alive.

The third, Oups/Dwayne, is a technician and is the only one of the three that can pick up and use items.

## Interaction / Controls (how the player play the game)

Controls/Interactions with the goblins & decor have been re-designed for touch mode.



"The warrior punch something"



"The magician cast a spell"



"The technician picks the corn and then uses it"

You can select another gobliin by clicking on the crystal ball in the bottom bar. It will be cyclic, so every time you click on this button, a different gobliin will be selected.

You have 2 ways to move a goblin thanks to the touch screen. By clicking quickly on an empty area, your gobliin will move at its destination. If your touch is a long one, by releasing your finger, you will see all the actions possible by the gobliin on the empty area you selected. Then, you can use the cursor mode (available in the settings) to have precision and point unattainable areas.

A goblin can also interact with his environment. 2 categories of hotspot exist.

The first hotspot is about objects that can be useful for the gobliins. They can be picked or used in order to help the gobliins to resolve the current scene. The second type concerns the static objects. The technician can't pick them but they can be used to launch animations.

The gobliins actions are different. The magician and the warrior have just one action they can do on a hotspot. The magician can cast a spell and the warrior can punch it (cf screenshots above). However, if an empty area is selected, they will both just move to it.

It's a little bit more complicated for the technician. On the first type of hotspot (useful object), he can do 2 actions: he can use an object he has, or he can pick the object on the hotspot selected. On the second one (static object), he can only do the action "use" (cf screenshots above). Finally, on an empty area, the technician can do one of the 3 actions (move, use, pick).



"Here you can see all the actions possible by the technician an a hotspot (the door). By moving your finger on the screen, you will see exclamations marks corresponding of the different hotspots in the scene. If you enter in a specified hotspot, the actions your goblin can made will appear too."



"Here, you can see all the actions possible by the technician on a free area. This situation appears when you make a long touch on an empty area. When you'll release your finger from the screen, these actions will appear."

### Game progression (how is made the progression in the game)

The game is composed of 22 levels. Levels take more time to complete as the players progress through the game.

When the player finishes a level, this level is unlocked. When the player starts a new game he can start playing from the first level or any level he has unlocked.

While solving a puzzle several actions may harm the gobliin involved, including leaving them in an unsafe place, using incorrect items, or interacting with items with the wrong characters, thus reducing their shared life meter. When the meter runs out, the game ends in defeat (though all levels can be completed without loss of life) and the player will have to restart the level from the beginning.

If a “game over” situation appears, the player will have the choice to start again or to quit. If he quits, he will be redirected to the main menu of the game. If he chooses to start again, the current level will be loaded from the beginning and the life bar will be set to the maximum.

## Storyboard

There will be fade-in/out from the beginning to the main menu.



Screen 1: Popup to keep the music. OK: audio is off, iPod music is on. Cancel: audio is on (automatically kill the iPod music).



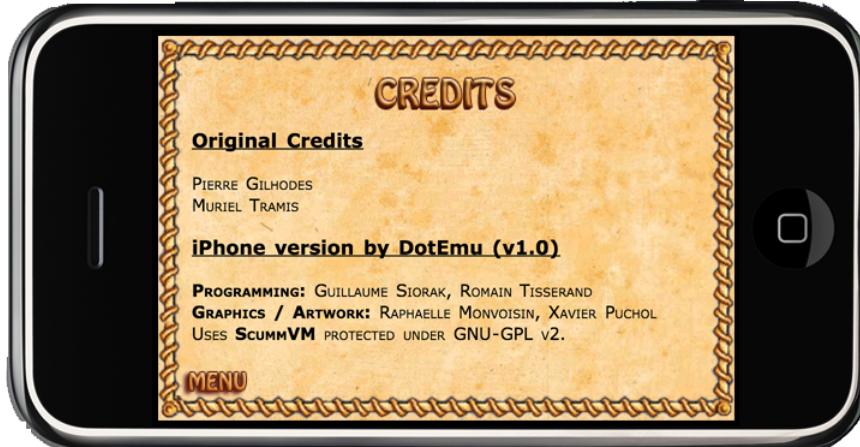
Screen 2: Popup to resume the game. OK: game is resumed to the last unlocked level. Cancel: application continues to the main menu.



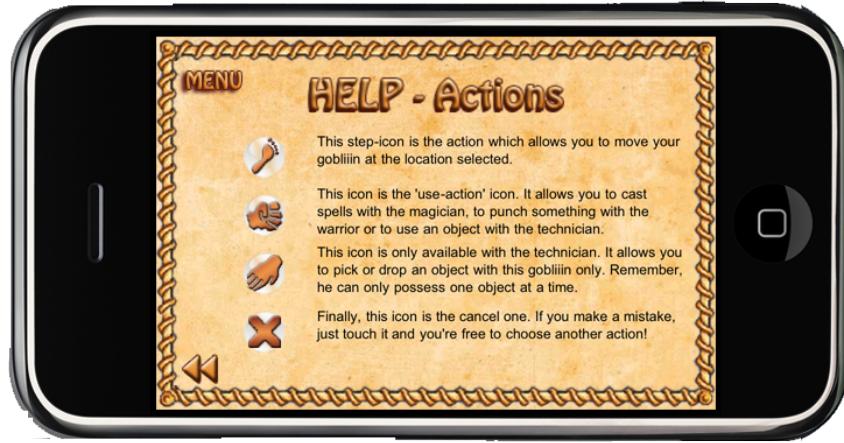
Screen 3: Logos (DotEmu + Bulkypix)



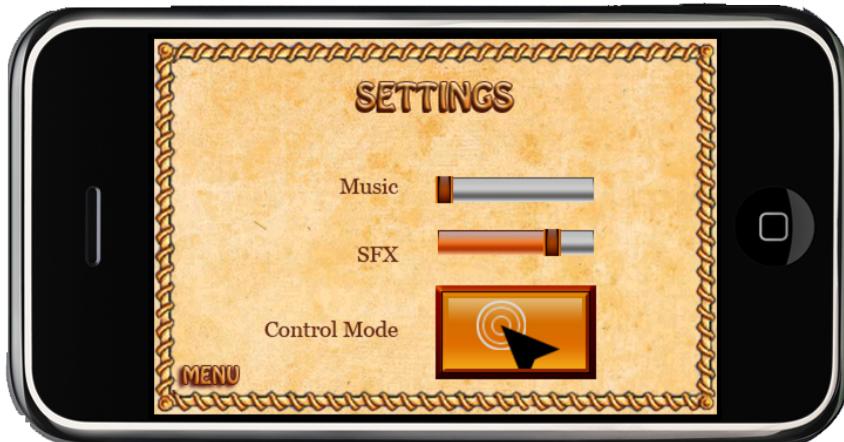
Screen 4: Main menu of Gobliins.



Screen 5: Credits will display the credits of the original game. A button "menu" is available to get back to the main menu.



Screen 6: Help will show different slides to help the player play the game. A button "menu" is available to get back to the main menu.



Screen 7: Settings will allow the player to set the volume of the music and the SFX. In Gobliiins, there is also a button to change the control mode. Cursor mode by default. A button "menu" is available to get back to the main menu.



Screen 8: Load is the screen to load your progression of the game. An icon will show if the level selected is unlocked. The player can only choose a level unlocked before. A button “menu” is available to get back to the main menu.

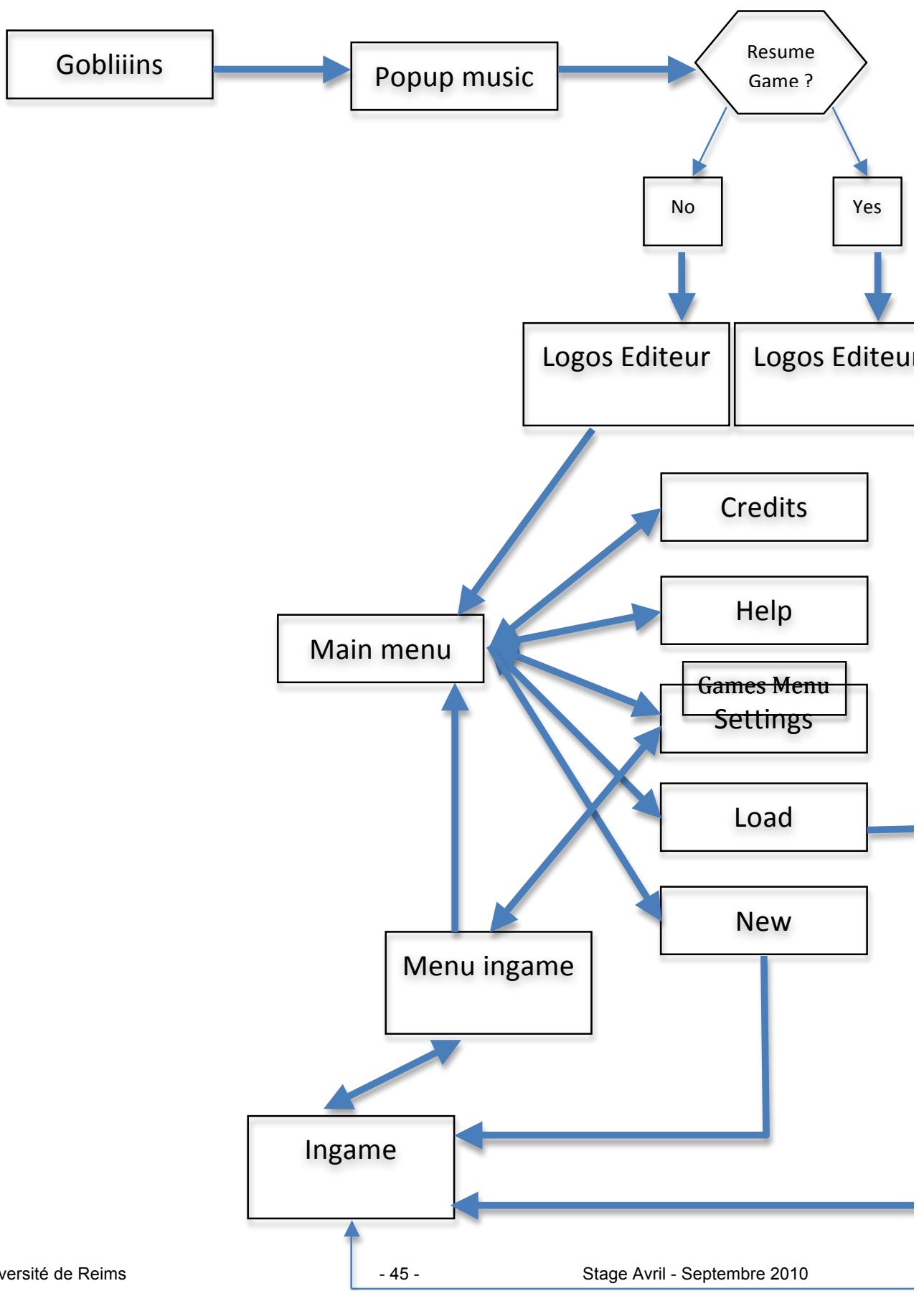


Screen 9 & 10: New will launch the game from the beginning (animation + 1<sup>st</sup> level).



Screen 11: Menu in game. Resume will go back to the game. Start again will restart the current level. Settings will show the settings menu (screen 7). Exit will go back to the main menu (screen 4).

## Flow chart



## Visual style (how can we define the visual style of the game)

The game has a unique cartoon style.

The visual style has well aged.

## Audio style (how can we define the audio style of the game)

The game features floppy/CD audio.

The CD version offers dialog and a soundtrack of a better quality.

The floppy one offers funny gibberish when the goblins speak.

The game soundtrack offers a unique immersion into the world of gobliins.

## Platform specific

(device locked, device entering sleep mode, taking/declining a call, alarm/calendar/notifications, powering off the device, etc ...).

- If the device received a call during a game, the current level is saved. When the player launches the game the next time, he'll be asked if he wants to resume his game or start a new one.
- When the game starts, if music is actually playing from the iPod, the player is asked if he wants to keep playing his music instead of the original game's music.
- With iPhone 4, when the player goes in background, the in-game menu appears and the game is paused. When the player goes back in game, everything resumes correctly.

## Bibliography (resources used to make this GDD + external resources)

<http://en.wikipedia.org/wiki/Gobliins>