Light | Dark

**Pour l’obtention du Diplôme Universitaire de Technologies - Informatique**

**Rapport de projet**

**Année universitaire 2013 - 2014**

Réalisé par :   
**Mickaël CAMPMAS – Léo CASTERA– Jean-Joseph MARTY – Théo KRISZT**

Sous la direction de :   
**Dr. Abdelkader GOUAICH**

Remerciements

Sommaire

[1 Cahier des Charges : Game Design Document - 2 -](#_Toc377487138)

[1.1 Introduction - 2 -](#_Toc377487139)

[1.2 Gameplay - 2 -](#_Toc377487140)

[1.3 Level Design - 2 -](#_Toc377487141)

[1.4 Environnement graphique - 2 -](#_Toc377487142)

[1.5 Environnement sonore - 2 -](#_Toc377487143)

[2 Rapport technique - 3 -](#_Toc377487144)

[2.1 Conception - 3 -](#_Toc377487145)

[2.2 Développement des principaux concepts - 4 -](#_Toc377487146)

[2.2.1 Outils et technologies utilisés - 4 -](#_Toc377487147)

[2.2.2 Déplacements - 4 -](#_Toc377487148)

[2.2.3 Collisions - 4 -](#_Toc377487149)

[2.2.4 Light Form - 4 -](#_Toc377487150)

[2.2.5 Shadow Form - 4 -](#_Toc377487151)

[2.3 Regroupement des différentes fonctionnalités - 4 -](#_Toc377487152)

[3 Manuel d’utilisation - 5 -](#_Toc377487153)

[3.1 Manuel d’installation - 5 -](#_Toc377487154)

[3.1.1 Installation sur PC - 5 -](#_Toc377487155)

[3.1.2 Installation sous Android - 5 -](#_Toc377487156)

[3.2 Manuel d’utilisation - 5 -](#_Toc377487157)

[3.2.1 Se déplacer - 5 -](#_Toc377487158)

[3.2.2 Changer de forme - 5 -](#_Toc377487159)

[3.2.3 Attaquer - 6 -](#_Toc377487160)

[3.2.4 Se glisser dans une ombre - 6 -](#_Toc377487161)

[4 Rapport d’activité - 7 -](#_Toc377487162)

[4.1 Méthode de développement - 7 -](#_Toc377487163)

[4.1.1 Mise en place de la méthode Agile Scrum - 7 -](#_Toc377487164)

[4.2 Planification - 7 -](#_Toc377487165)

[4.2.1 Planning prévu - 7 -](#_Toc377487166)

[4.2.2 Planning réels. - 8 -](#_Toc377487167)

[4.3 Méthode et outils de travail - 8 -](#_Toc377487168)

[4.3.1 Section modélisation - 8 -](#_Toc377487169)

[4.3.2 Section Codage - 8 -](#_Toc377487170)

[4.3.3 Section organisation. - 9 -](#_Toc377487171)

Table des figures

[Figure 1 - Use case Light Form - 3 -](file:///C:\Users\Dell\Desktop\Remerciements.docx#_Toc375047763)

[Figure 2 - Use Case Shadow Form - 3 -](#_Toc375047764)

Glossaire

**Gameplay**: Terme difficile à expliciter. Série de règles définissant la jouabilité d'un jeu, ce qui comprends les contrôles, l'intéraction entre le jeu et le joueur, les objectifs, le challenge, les récompenses...

**libGDX** : Terme difficile à expliciter. Série de règles définissant la jouabilité d'un jeu, ce qui comprends les contrôles, l'intéraction entre le jeu et le joueur, les objectifs, le challenge, les récompenses...

Introduction

Le but de ce projet est d’élaborer un jeu vidéo en deux dimensions, programmé en Java à l’aide de la librairie libGDX\*. Le jeu suivra des règles précises et essayera d’attirer le joueur et de lui proposer de se divertir tout en se concentrant.

Le gameplay\* du jeu proposera au joueur d’incarner un personnage dont les aptitudes seront influencées par le cycle jour/nuit à travers des phases de combat ou d’infiltration selon une orientation action/puzzle.

L’enjeu principal consistera à élaborer un gameplay\* innovant et indépendant du scénario et ce, malgré nos connaissances dans le domaine de la création de jeu vidéo encore peu développées. Nous avons dû suivre une méthode bien spécifique et propre à ce média. Il s’agit donc là, contrairement à la croyance populaire, de ne pas créer un jeu à partir d’un scénario mais d’élaborer un gameplay\* dont le scénario permet de justifier les actions entreprises dans ce dernier. Cette composante du jeu doit donc être créée avant et séparément du scénario.

Ce projet a pour but de nous initier aux mécanismes principaux de développement d’un jeu vidéo. Il nous montre aussi les aspects vitaux de la réalisation d’un projet en groupe.

Pour réaliser un gameplay\* complet et original, nous avons commencé par nous documenter sur des règles existant et méthodes de game designer\*. Aussi, nous nous sommes fortement inspirés de la mécanique principale des jeux de rôles, en particulier des premiers Zelda. Le concept peut se résumer aux déplacements dans les huit directions et une attaque. Contrairement à la majorité des jeux de rôles, nous n’avons pas intégré de moyens directs de défense. Cependant, notre personnage devait avoir une part importante de faiblesse afin d’accentuer la difficulté du jeu. Nous sommes donc orientés vers un gameplay\* en deux cœurs diamétralement opposé : l’un se basant sur l’action et l’autre se basant sur la réflexion.

Dans un premier temps nous parlerons du document de Game-Design, fruit de cette réflexion, qui explicite ce que nous avons décidé de réaliser. Ensuite, nous expliquerons quels sont les mécanismes qui régissent le jeu, les méthodes, techniques et technologies mises en œuvre afin de réaliser ce projet. Enfin, nous présenterons notre expérience désormais enrichie en tant que développeurs de jeu vidéo grâce à l’utilisation de méthodes et d’outils de travail spécifiques.

# Cahier des Charges : Game Design Document

## Introduction

## Gameplay

## Level Design

## Environnement graphique

## Environnement sonore

# Rapport technique

## Conception

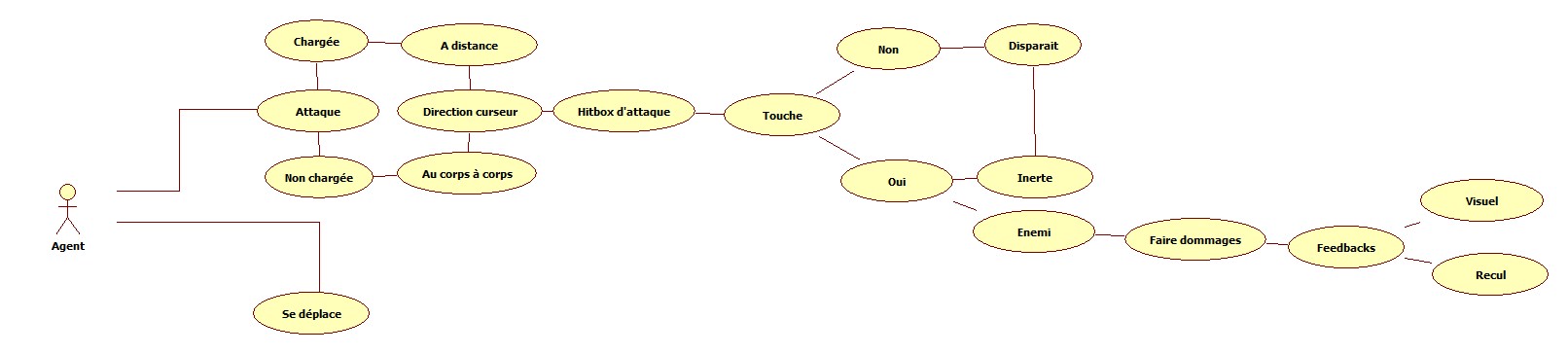


Figure - Use case Light Form

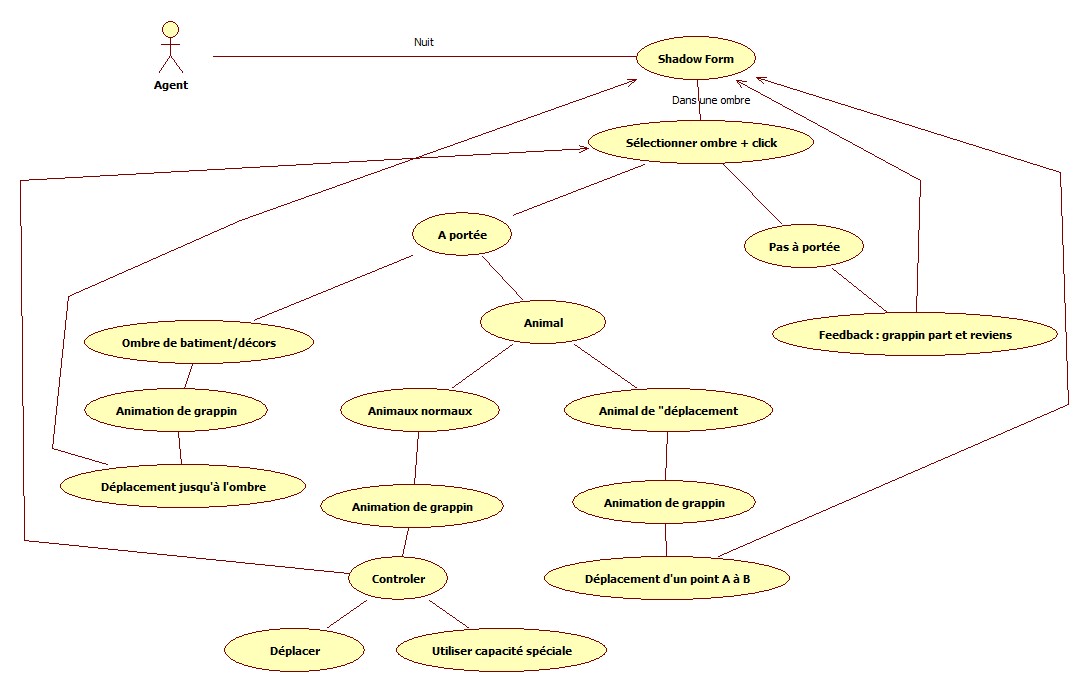


Figure 2 - Use Case Shadow Form

## Développement des principaux concepts

### Outils et technologies utilisés

#### StarUML

#### Java

#### libGDX

### Déplacements

### Collisions

### Light Form

### Shadow Form

## Regroupement des différentes fonctionnalités

# Manuel d’utilisation

## Manuel d’installation

### Installation sur PC

#### Environnement Java (JRE)

Une machine virtuelle Java est nécessaire pour lancer le projet. L’installateur est disponible sur le site officiel [www.java.com/fr/](http://www.java.com/fr/). Lancez et complétez l’installation pour disposer de la machine virtuelle Java qui permettra d’exécuter le jeu.

#### Récupération du jeu

La version java du projet est disponible en téléchargement à l’adresse [www.projet-lightdark.fr/](http://www.projet-lightdark.fr/). Téléchargez le fichier .jar et double-cliquez dessus. Le jeu va maintenant se lancer.

Si vous utilisez une console sous Linux, placez-vous dans le répertoire du projet *cd chemin/du/projet* puis exécutez le fichier avec la commande *java jar monfichier.jar*

### Installation sous Android

#### Récupération du jeu

Pour pouvoir installer le jeu sur la mémoire de votre appareil, la première étape consiste à transférer la version Android (.apk) du projet depuis [www.projet-lightdark.fr/](http://www.projet-lightdark.fr/) vers la mémoire de votre appareil.

#### Gestionnaire de fichiers

Ensuite, téléchargez et installez depuis le magasin virtuel de Google® un explorateur de fichiers comme Astro™ (application gratuite). Naviguez ensuite dans la mémoire de votre appareil jusqu’ à l’endroit où le projet a été placé. Sélectionnez le fichier et choisissez d’installer l’application. Cette dernière est désormais disponible sur votre appareil.

## Manuel d’utilisation

### Se déplacer

Le personnage peut se déplacer à l’aide des touches ZQSD du clavier ou, sur Android, à l’aide des flèches directionnelles affichées à l’écran.

Il est possible de se déplacer dans les quatre directions cardinales et en diagonale.

### Changer de forme

En cliquant ou en appuyant sur l’orbe, le joueur change temporairement de forme. Si l’orbe est grisé alors il n’est pas possible de changer de forme pour le moment.

L’orbe est utilisable uniquement si le joueur est placé sur une ombre. De plus, certaines zones peuvent empêcher l’orbe de fonctionner.

### Attaquer

Quand il est sous sa forme de lumière, le joueur peut attaquer en cliquant ou en appuyant dans la direction voulue. Le personnage porte alors un coup d’épée dans la direction pointée. Il peut ainsi combattre les monstres qui se dressent contre lui.

### Se glisser dans une ombre

Sous forme d’ombre, le joueur se déplacer d’ombre en ombre. Pour ce faire, il dispose d’un grappin qui lui permet de se déplacer sur l’ombre pointée si elle est bien à portée de son grappin. Sinon, le grappin revient à son lanceur.

Il est également possible de prendre le contrôle d’animaux en se glissant dans leur ombre. Il est ainsi possible de contrôler les mouvements de l’animal : se déplacer et utiliser son action spéciale. Cette action spéciale est propre à chaque animal et se déclenche en gardant le clic (ou le doigt) enfoncé lorsque l’on contrôle l’animal.

# Rapport d’activité

## Méthode de développement

Pour le développement de notre projet, nous avons adopté, à l'initiative de notre tuteur, M Abdelkader Gouaich, la méthode Agile Scrum.

### Mise en place de la méthode Agile Scrum

#### Méthode Scrum : Définition

La méthode Scrum est une méthode Agile pour gérer les projets.

Le travail à réaliser est découpé en différentes parties, nommées sprints, qui couvrent chacun un laps de temps (dans notre cas, de une à trois semaines).

Au début de chaque sprint, nous nous retrouvions avec notre tuteur pour faire le point sur le sprint précédent, et fixer les objectifs pour le sprint courant.

Nous nous organisions alors pour nous répartir librement les différentes tâches entre nous, selon le principe d’auto-organisation de cette méthode.

Enfin nous nous réunissions fréquemment durant les sprints au cours de réunions de travail ou de réflexions, que nous avons nommé séances de brainstorming.

#### Bilan d'utilisation

Nous avons rapidement réussi à mettre en place cette technique.

L’utilisation de la méthode Agile Scrum nous a permis, rétrospectivement, de structurer et d’organiser l’avancement de notre projet, combiné à l’utilisation de Trello.

Cependant nous avons aménagé et modifié quelque peu cette méthode, pour l’adapter avec nos autres contraintes scolaires.

////////////////////////

Réunions avec Gouaich

Brainstorming

#### Conclusion de l'utilisation de la méthode Agile Scrum

///////////////////////////////////////////

## Planification

### Planning prévu

Comme demandé par les consignes, nous avons réalisé un planning initial, sous forme de diagramme de Gantt, couvrant l’intégralité de la durée du projet, en évaluant à priori les taches à réaliser et leur durée.

Ce planning global a au final été peu suivi, pour plusieurs raisons :

Nous ne savions pas comment créer un jeu vidéo, ni d’idée précise sur les tâches à réaliser.

En particulier concernant la rédaction du Game design document, nous avions estimé une durée d’une semaine, et au final, cette tâche, bien plus complexe que prévue, nous a occupés près de six sprints.

D’autre part, les autres travaux que nous ont confiés l’IUT nous ont malgré nous pris beaucoup de temps, qui a d’autant retardé l’avancement du projet.

//(Mettre ici ou en annexe le planning initial prévu qui est sur le wiki)

### Planning réels.

#### Des plannings sprints par sprints

Nous avons, en plus du planning global, réalisé pour chaque sprint des plannings des taches à l’aide de diagrammes de Gantt. Ces planning, plus en adéquation avec la réalité de l’avancement de notre projet, ont été plus réalistes et mieux suivis.

#### Les contraintes

Ces plannings, bien que plus réalistes, ne comprenaient pas nos contraintes, notamment les vacances ou bien les examens, ils n’ont donc pas été suivis parfaitement.

#### Conclusion sur les plannings

Bien que nous ayons réalisés les plannings, ils n’ont été finalement que moyennement suivi, du fait des différents facteurs cités plus haut.

A posteriori, l’utilisation de ces plannings ne nous a apporté que peu de choses, comparé à l’utilisation des outils Wiki et Trello.

## Méthode et outils de travail

### Section modélisation

#### Modélisation UML

Pour la modélisation nous avons choisi d'utiliser surtout le logiciel Modelio, mais aussi parfois le logiciel StarUML, qui sont des logiciels gratuits de modélisation UML.

#### Autres diagrammes

Pour réaliser quelques diagrammes explicatifs sous forme de cartes mentales, à destination du wiki, nous avons également utilisé occasionnellement le logiciel Xmind.

Enfin nous avons réalisé quelques schémas à la main, que nous avons ensuite scannés.

### Section Codage

Pour le codage, nous avons utilisé le langage de programmation JAVA, la plateforme de développement éclipse, et libGDX, avec SVN sur un serveur Google.

#### JAVA

Nous avons dû utiliser, dans le cadre de notre projet, libGDX qui est une librairie écrite en java. Ainsi, le choix du langage de programmation s’est imposé : Java

#### Eclipse

Eclipse est un IDE axée développement Java

C’est un logiciel que nous avions l’habitude d’utiliser dans le cadre de notre formation, nous avons donc naturellement continué avec ce logiciel.

#### LibGDX

LibGDX est une librairie Java, axée sur la création et le développement de jeu vidéo.

LibGDX est une librairie que nous ne connaissions pas, et qui nous a été imposé par le projet, et qui s’est révélé être d’une grande aide pour le développement du jeu vidéo, en nous offrant de multiples méthodes déjà réalisée et en nous fournissant un squelette de code type.

#### SVN

SVN est un outil de gestion de version, qui permet de gérer le développement d’un projet à plusieurs membres.

Nous avons choisis d’utiliser SVN, plutôt que GIT, son concurrent, car c’est un outil sur lequel nous avons été formés dans le cadre de notre formation.

SVN, malgré quelques erreurs, nous a permis de travailler en commun sur une ou parfois plusieurs copies du programme,

Nous avons choisi de stocker notre projet sur un serveur Google, de par sa simplicité d’utilisation et sa gratuité.

#### Utilisation coordonnée

Une séance usuelle de travail sur le code du jeu suivait les étapes suivantes :

-Définition des objectifs de la séance sur Trello

-Récupération de la dernière version du dépôt SVN

-Développement de la fonctionnalité fixée plus tôt

-Mise à jour de la copie de travail puis enregistrement des modifications sur le dépôt

### Section organisation.

#### Wiki

Nous avons réalisé un wiki, pour permettre aux observateurs de suivre l’évolution de notre projet, tel que demandé dans les consignes du projet.

Nous avons préféré l’utilisation d’un wiki contre celle d’un blog, pour plus de simplicité : nous sommes en effets plus habituées à utiliser un wiki qu’un blog, et de plus, l’utilisation d’un wiki nous a semblé plus intéressante.

Nous avons veillé à tenir ce wiki à jour, d’autant plus qu’il s’est rapidement révélé utile pour savoir entre nous l’avancement global du projet, de savoir où nous en étions.

Permet de suivre l'évolution du projet

Permet de stocker les comptes rendus des réunions et de savoir l'avancement général du projet.

//////////////////////////////////////utile

#### Trello

Trello est un outil web gratuit permettant d’organiser et de gérer des projets. Les tâches, qui peuvent être ajoutés par chacun des membres, peuvent être ajoutés dans trois colonnes : à faire, en cours, fait, et sont déplaçable de l’une à l’autre. Elles peuvent par ailleurs se voir attribuer des membres.

Bien utilisé, Trello nous a permis de répondre aux questions : « Que faut-il faire ? », « Qui fait quoi ? », « Ou en est-on? », et nous a permis de nous organiser facilement au niveau de chaque tâche.

Cet outil que nous avons tenté d’associer à chacune de nos séances de travail, en particulier lors du codage ou les petites tâches se sont démultipliées, a pu être d’une grande aide.

#### Communication

Au niveau du travail, nous avons travaillé en groupe et individuellement.

Concernant le travail de groupe, nous avons privilégié tant que possible la rencontre de visu, de préférence dans les locaux de l’iut de Montpellier.

Lorsque nous travaillions ensemble mais à distance, nous avons utilisé Teamspeak comme logiciel de communication.

#### Rapport

Enfin, pour l’écriture du rapport, nous avons choisi d’utiliser Microsoft Word 2013, pour nous simplifier la mise en page et l’écriture du rapport en utilisant une plateforme moderne et identique pour chaque des membres du groupe.

Conclusion

Bibliographie / Sitographie

**Aucune source spécifiée dans le document actif.**

Annexes

**Résumé**

Le projet Light/Dark vise à créer un jeu vidéo en deux dimensions en passant par toutes les étapes de création d’un jeu, du document de Game-Design jusqu’à la réalisation d’un modèle pleinement fonctionnel pouvant être présenté puis joué par des joueurs réguliers.

Il illustre bien les différentes contraintes liées à la réalisation d’un jeu vidéo orienté puzzle/aventure : analyse, ergonomie, mécanismes, environnement visuel et sonore, mais surtout la relation particulière entre le scénario et le gameplay.

Ce jeu a été développé en utilisant le langage de programmation Java et la librairie libGDX.

**Mots clés**

Jeu vidéo, 2D, Game-Design Document, Gameplay, scénario, Boucle OCR, Java, libGDX, Agile, SCRUM.