

***PLANTS***

***VS***

***ZOMBIES***



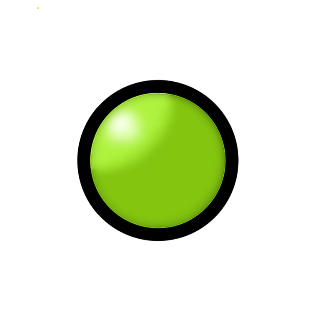


Table des matières

João Victor

DA SILVA JUSSANI

CID2A

P\_OO 320

[1 Analyse préliminaire 3](#_Toc176348919)

[1.1 Introduction 3](#_Toc176348920)

[1.2 Objectifs 3](#_Toc176348921)

[1.3 Gestion de projet 4](#_Toc176348922)

[2 Analyse / Conception 4](#_Toc176348923)

[2.1 Gameplay 4](#_Toc176348924)

[2.2 Concept 4](#_Toc176348925)

[2.3 Analyse fonctionnelle 4](#_Toc176348926)

[2.4 Stratégie de test 4](#_Toc176348927)

[3 Réalisation 4](#_Toc176348928)

[3.1 Points de design spécifiques 4](#_Toc176348929)

[*3.1.1* *…* 5](#_Toc176348930)

[*3.1.2* *…* 5](#_Toc176348931)

[*3.1.3* *…* 5](#_Toc176348932)

[3.2 Déroulement 5](#_Toc176348933)

[3.3 Mise en place de l’environnement de travail 5](#_Toc176348934)

[3.4 Description des tests effectués 6](#_Toc176348935)

[3.5 Erreurs restantes 6](#_Toc176348936)

[4 Conclusions 6](#_Toc176348937)

[5 Annexes 7](#_Toc176348938)

[5.1 Manuel de référence 7](#_Toc176348939)

[5.2 Journal de travail 7](#_Toc176348940)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS :*  
*Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

# Analyse préliminaire

## Introduction

Le projet est une inspiration de Plants vs Zombies où dans le jeu il y a des zombies qui vienne de droite à gauche avec un seul but : manger tous ce qu’il y a sur son chemin ! Notre but ? Avoir une bonne stratégie de gestion de point et de placement des plantes pour mettre les plantes bouclier, les plantes qui génère des points et celles qui vont tirer sur les zombies. J’ai choisi ce jeu car c’est un jeu de shoot, c’est une vue de dessus mais au lieu que ça soit de bas en haut c’est de droite à gauche. Ce projet ira m’apporter beaucoup de connaissance dans l’orientation objet de C# et une motivation plus grande à apprendre car c’est un jeu que j’apprécie beaucoup.

## Objectifs

L’objectif de se projet se base dans l’apprentissage de l’orienté objet, savoir différencier des classes avec des objets, savoir comment faire une encapsulation de variable, les exceptions, savoir comment fonctionne l’héritage, faire des listes, savoir la fonctionnalité des classes statiques et savoir comment faire des tests unitaires

# Gestion de projet

La méthode de projet utilisé est SCRUM. Avec cette méthode j’ai planifié toutes les tâches à faire avant de commencer à travailler, j’ai préparé des tests pour que je sache quand les tâches sont terminées. Tous les matins avant de commencer le travail, je regarde les tâches que j’ai réalisé la fois d’avant, je regarde les tâches que je compte faire le jour et si j’ai un problème, le résoudre en essayant ou en cherchant sur internet.

# Analyse / Conception

## Gameplay

* Le joueur :
  + Le but du joueur c’est d’administrer des plantes et des points pour que les plantes puissent tirer sur des zombies qui veulent les tuer
* Les ennemis (les zombies) :
  + Leur objectif est de manger les plantes sur le chemin pour atteindre l’arrivé
* Les déplacements :
  + Les déplacements sont uniquement des zombies et que leurs déplacements sont de droite à gauche pour arriver à la ligne d’arrivée.
* Les niveaux :
  + Mon envie pour les niveaux est de faire une quinzaine de niveau où la quantité de zombies augmentent et avec plus de vies.
* Le tir :
  + Le tir est une partie super importante. Pour placer des plantes sur la map, il faut collecter une certaine quantité de points soleil pour le faire. La plante qui génère des points coûte 50 points et elle donne des soleils après un certain temps, le tireur principal coûte 100 points soleil. Le tir est fait après un certain temps le personnage ira tirer jusqu’à que le zombie le tue.
* La gestion des vies
  + Dans le jeu de base, les vies sont les lignes où les zombies rentrent et s’ils arrivent au bout du jardin, une tondeuse passe par-dessus tous ce qui est dans la ligne. On a le droit a une tondeuse par ligne mais dans mon jeu il aura 3 vies, si les trois cœurs sont perdus, la personne a perdu et aura le choix de recommencer la partie ou de repartir sur le menu principal du jeu.
* Gain de points
  + Le gain de points est assez simple. Au début on commence avec 0 points mais après un certain temps aléatoire il y a aura des petits soleils qui vont tomber du ciel et en les collectant on gagne 25 points.

## Concept

* *Diagramme de classe*
* *Diagramme(s) d’état*

## Analyse fonctionnelle

***Reprendre le contenu des User Stories d’IceScrum : Story + tests d’acceptance (avec IceTools) + maquettes***

## Stratégie de test

***Décrire quels sont les MOYENS utilisés pour faire les tests, ne pas décrire les tests à effectuer !!!***

***Décrire l’environnement dans lequel se fait la sprint review***

*Décrire la stratégie globale de test :*

* *types de des tests et ordre dans lequel ils seront effectués.*
* *les moyens à mettre en œuvre.*
* *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
* *données de test à prévoir (données réelles ?)* ***et comment elles seront mises en place****.*
* *les testeurs extérieurs éventuels.*

# Réalisation

## Points de design spécifiques

***Ce chapitre est constitué de plusieurs sous-chapitre.***

***Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s’expliquer par de simples commentaires dans le code.***

***Il s’agit d’explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.***

***NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n’EXPLIQUEZ PAS ICI CE QU’EST LE PATTERN MVC.***

***Exemple (simplifié à l’extrême) : Protection contre des formulaires mal intentionnés ou modifiés***

* ***Au moment de générer le formulaire, le script php :***
  + ***Concatène les noms de tous les champs contenus dans le formulaire***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Ajoute un input nommé « CSRF » de type hidden dans le form***
* ***A la réception du POST du fromulaire***
  + ***Concatène les noms des indices de $\_POST***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Vérifie que la valeur du champ CSRF correspond***

### *…*

### *…*

### *…*

## Déroulement

***Résumer comment s’est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, …***

## Mise en place de l’environnement de travail

* ***Comment accéder au code source***
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*

***Ce chapitre décrit précisément comment un employé qualifié peut recréer l’environnement dans lequel vous avez effectué ce travail***

## Description des tests effectués

***Reprendre les tests d’acceptance d’IceScrum au moyen de la feuille ad hoc d’IceTools***

## Erreurs restantes

*S'il reste encore des erreurs :*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

***Reporter la*** [***dette technique***](https://www.premaccess.com/qu-est-ce-que-dette-technique-comment-la-maitriser/#:~:text=La%20dette%20technique%20survient%20quand,de%20plus%20en%20plus%20fr%C3%A9quents.) ***connue. S’appuyer sur la pratique des // TODO***

# Conclusions

*Développez en tous cas les points suivants :*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Manuel de référence

***Issu de la génération automatique à partir des commentaires***

## Journal de travail