

Poly Defense

CAHIER D'ANALYSE



Dylan GALMICHE et Maxence MOHR – IRM2016
POLYTECH MARSEILLE | 162 AVENUE DE LUMINY – 13009 MARSEILLE

Sommaire

Présentation du projet	1
Présentation générale.....	1
Présentation de l'équipe.....	1
Présentation du commanditaire	1
Description fonctionnelle.....	2
Objectif principal	2
Fonctionnement global de l'application.....	2
Exigences fonctionnelles.....	2
Fonctionnalité 1 : Démarrer une partie.....	4
Fonctionnalité 2 : Afficher l'aide.....	4
Fonctionnalité 3 : Afficher les crédits	4
Fonctionnalité 4 : Afficher les options.....	4
Contraintes.....	5
Contraintes de réalisation.....	5
Contraintes de performances.....	5
Descriptif interne du système.....	6
Scénario de base.....	6
Lancer une partie	6
Phase de préparation	7
Phase de défense	8
Fin de partie	8
Organisation et structures de données.....	9
Planning prévisionnel	10



Présentation du projet

Présentation générale

Ce document présente le projet à réaliser dans le cadre de nos études à Polytech Marseille en section informatique, réseaux et multimédia ; plus précisément dans le cours de programmation orientée objet.

Il est demandé à l'équipe de développer un jeu de *tower defense* original, ayant les caractéristiques suivantes :

- Sans labyrinthe
- Avec au moins 3 types de monstres différents (plus ou moins rapides, plus ou moins résistants, etc...)
- Avec au moins 3 types de tours différents (cadence de tir plus ou moins rapide, coups plus ou moins efficaces, etc...)

Présentation de l'équipe

Pour la réalisation de ce projet, l'équipe est composée de deux membres, élèves ingénieurs à Polytech Marseille :

- Dylan GALMICHE
- Maxence MOHR

Présentation du commanditaire

Notre commanditaire est Monsieur MAVROMATIS, professeur en charge de notre groupe de projet de programmation orientée objet.

Description fonctionnelle

Objectif principal

L'objectif principal de ce projet est de réaliser un jeu de tower defense fonctionnel dans le délai imparti.

Fonctionnement global de l'application

Au démarrage de l'application, il sera présenté à l'utilisateur un menu principal. Depuis ce menu, l'utilisateur pourra choisir le niveau de difficulté et la carte qu'il désire, et enfin lancer la partie.

Chaque carte aura un certain nombre de vagues de monstres, s'intensifiant au fur et à mesure de la progression du joueur sur cette carte.

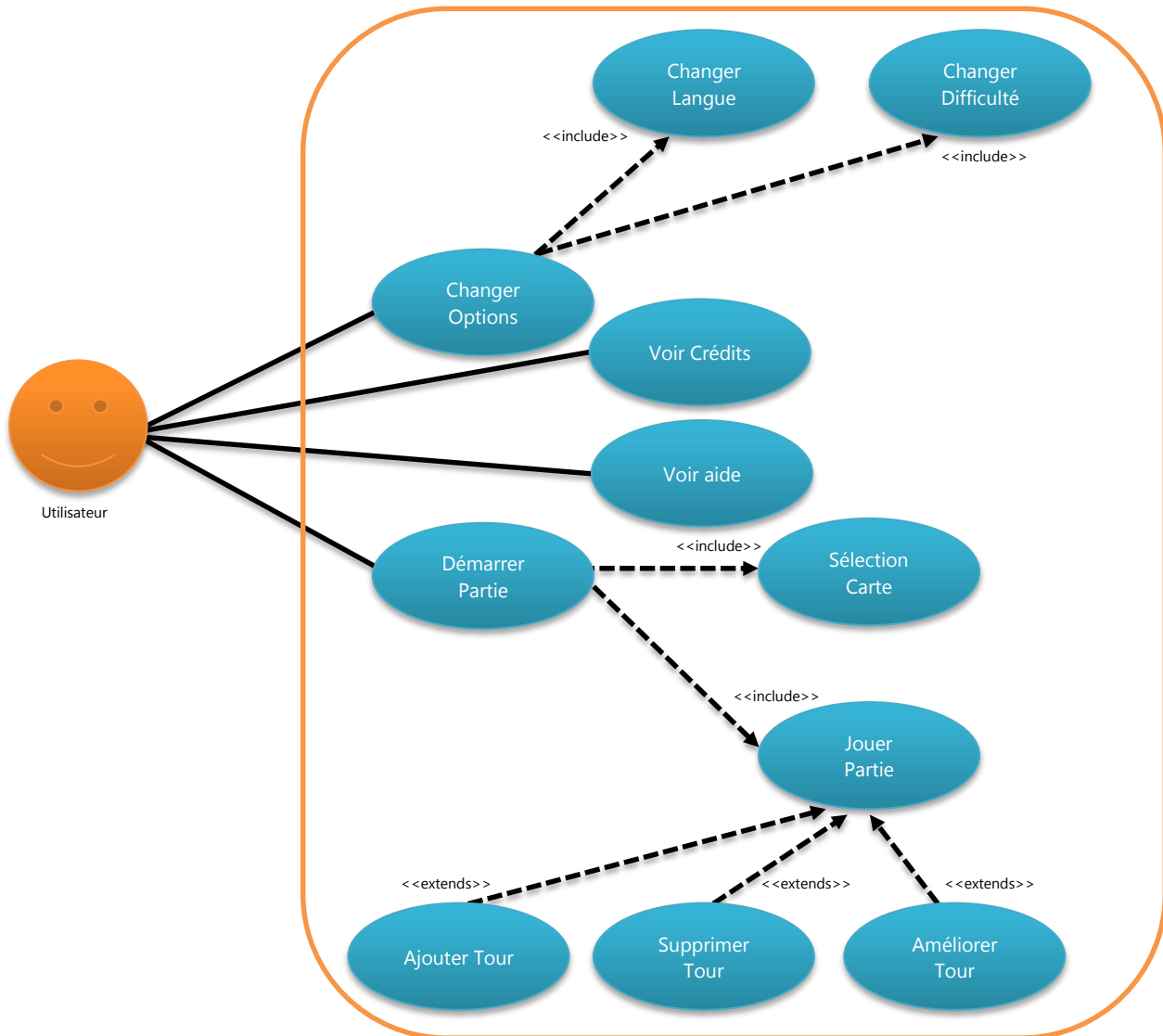
Seront également présents : un petit manuel d'aide et un bouton pour quitter l'application.

Une fois la partie lancée, le joueur pourra poser des tours, les mettre à jour ou les revendre pendant une phase dite phase de préparation. Une fois que le joueur aura fini, un bouton permettra de lancer la vague de monstre suivante ; et ainsi de suite jusqu'à la victoire ou la défaite du joueur.

Exigences fonctionnelles

Il est possible, au vu de la nature de l'application demandée, de réaliser un diagramme de cas d'utilisation.

Celui-ci est présent à la page suivante.



On discerne ainsi un seul acteur qui va interagir avec l'application : l'utilisateur (ou joueur). Il sera le déclencheur des principaux événements liés à l'application.

Les jeux de Tower Defense sont assez connus et répandus aujourd'hui, et il s'agit de respecter les conventions et incontournables du genre. Une carte durera un certain nombre de vagues. Entre chacune de ces vagues, le joueur pourra construire, vendre ou améliorer des tours.

Une fois toutes les vagues achevés, le joueur gagne. S'il se laisse submerger et que trop d'ennemis parviennent à s'échapper, le joueur perd.

Dans les options, il sera possible de configurer la langue (le support de l'anglais et du français à minima) et configurer la difficulté du jeu.

Il sera possible, pour les novices, de découvrir le gameplay avec l'aide : seront expliqués les bases des jeux de Tower Defense, ainsi qu'une aide à l'interface.

Enfin, un écran des crédits affichera les informations concernant les développeurs et la licence d'utilisation du logiciel.

Les fonctionnalités accessibles à l'utilisateur seront donc les suivantes :

Fonctionnalités
Démarrer une partie
Afficher l'aide
Afficher les crédits
Changer les options

Fonctionnalité 1 : Démarrer une partie

Cette fonctionnalité permettra au joueur de lancer une partie, et gèrera toute la partie. Les monstres, les vagues, la manipulation de tours, etc.

Fonctionnalité 2 : Afficher l'aide

Cette fonctionnalité, comme expliqué plus haut, permettra d'afficher quelques images et conseils pour le joueur, sur la façon de jouer, l'interface.

Fonctionnalité 3 : Afficher les crédits

Cette fonctionnalité affiche un écran d'information sur les développeurs et la licence.

Fonctionnalité 4 : Afficher les options

Cette fonctionnalité permettra au joueur de modifier quelques options, comme la difficulté ou la langue.

Contraintes

Contraintes de réalisation

Le jeu de Tower Defense devra être conçu en Java. Il ne devra pas consommer les ressources matérielles de manière excessive. Il devra être compatible avec Java 1.8 et les machines le supportant.

Contraintes de performances

Le jeu devra être le plus fluide possible sur différentes machines.

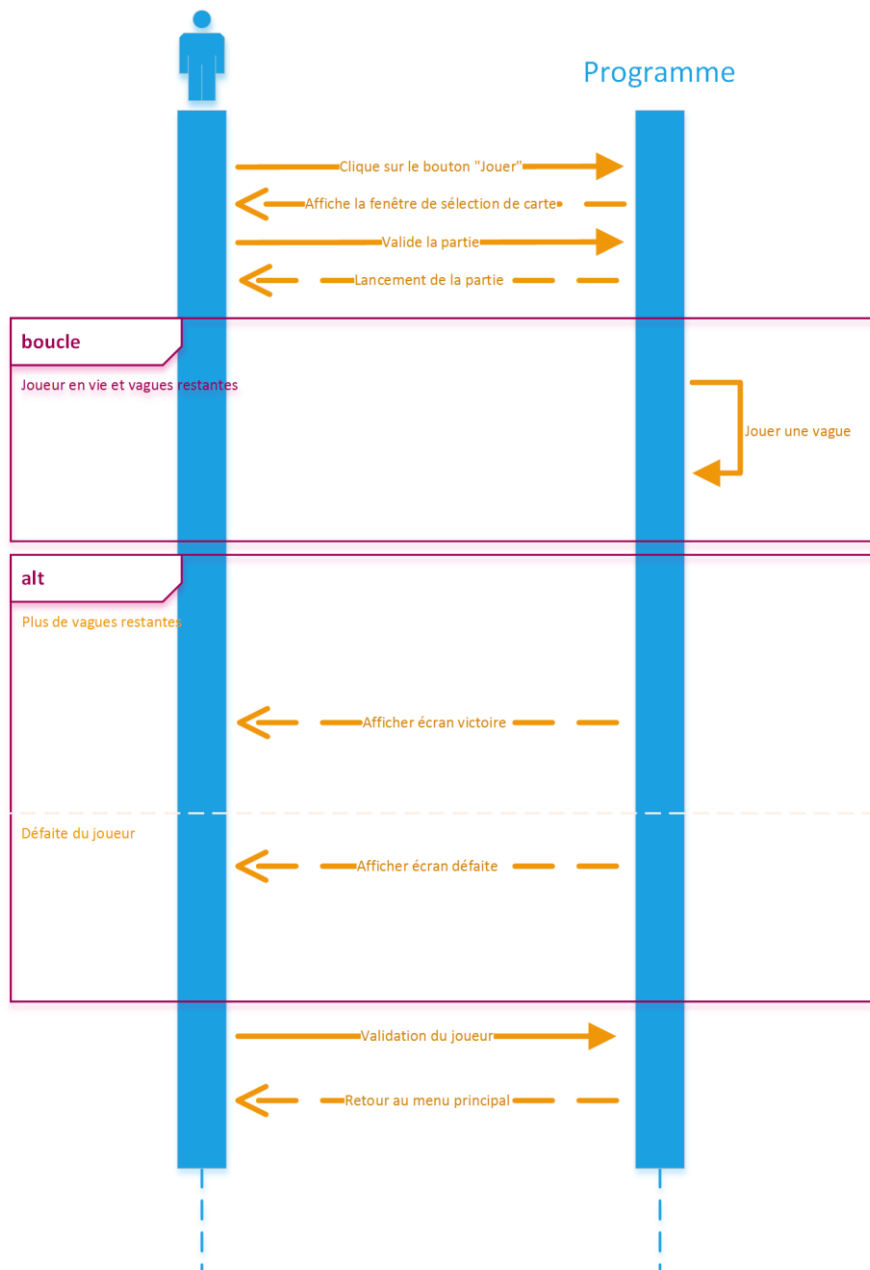
Descriptif interne du système

Scénario de base

Lancer une partie

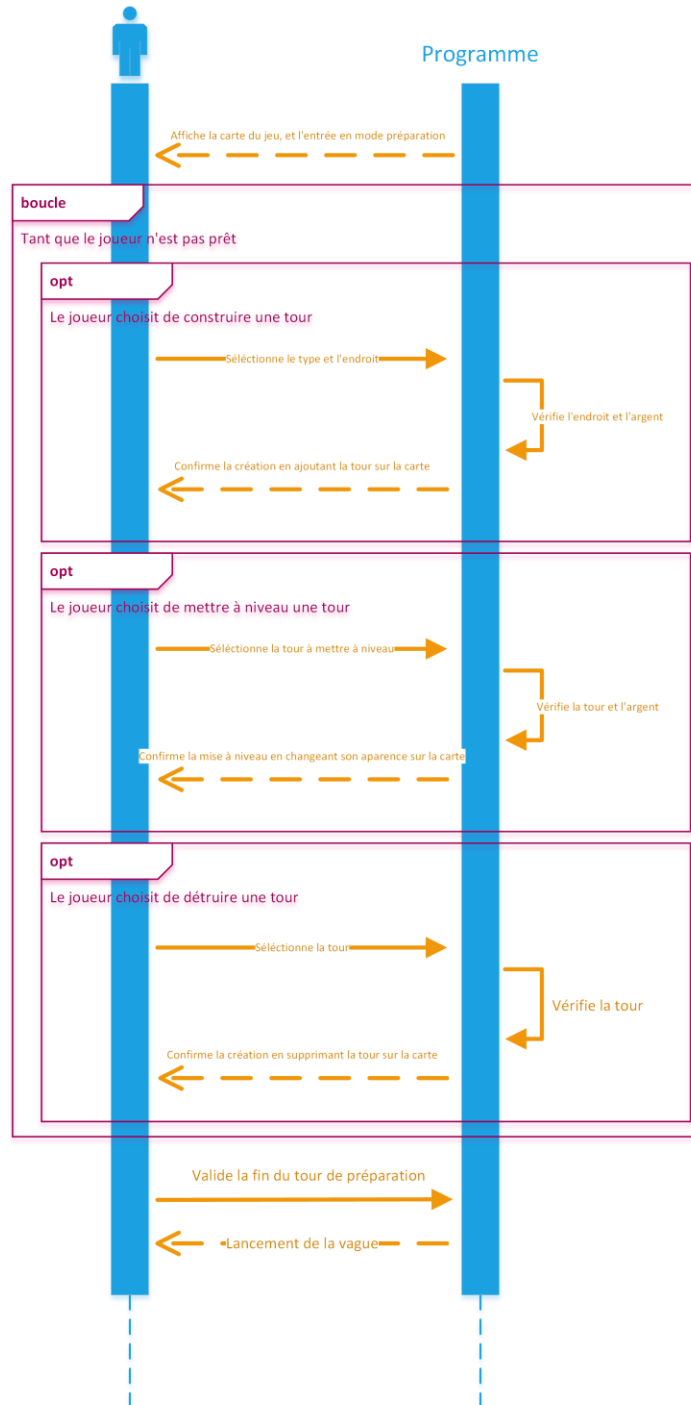
Pour lancer une partie, le joueur cliquera sur un bouton « Jouer ». Une fenêtre permettra de choisir la carte, et le joueur devra valider le démarrage de la partie.

Pour chaque vague (et tant qu'il y en a), il y aura une phase de préparation et une phase de défense et ce jusqu'à la fin de la phase de défense de la dernière carte (ou la défaite du joueur).



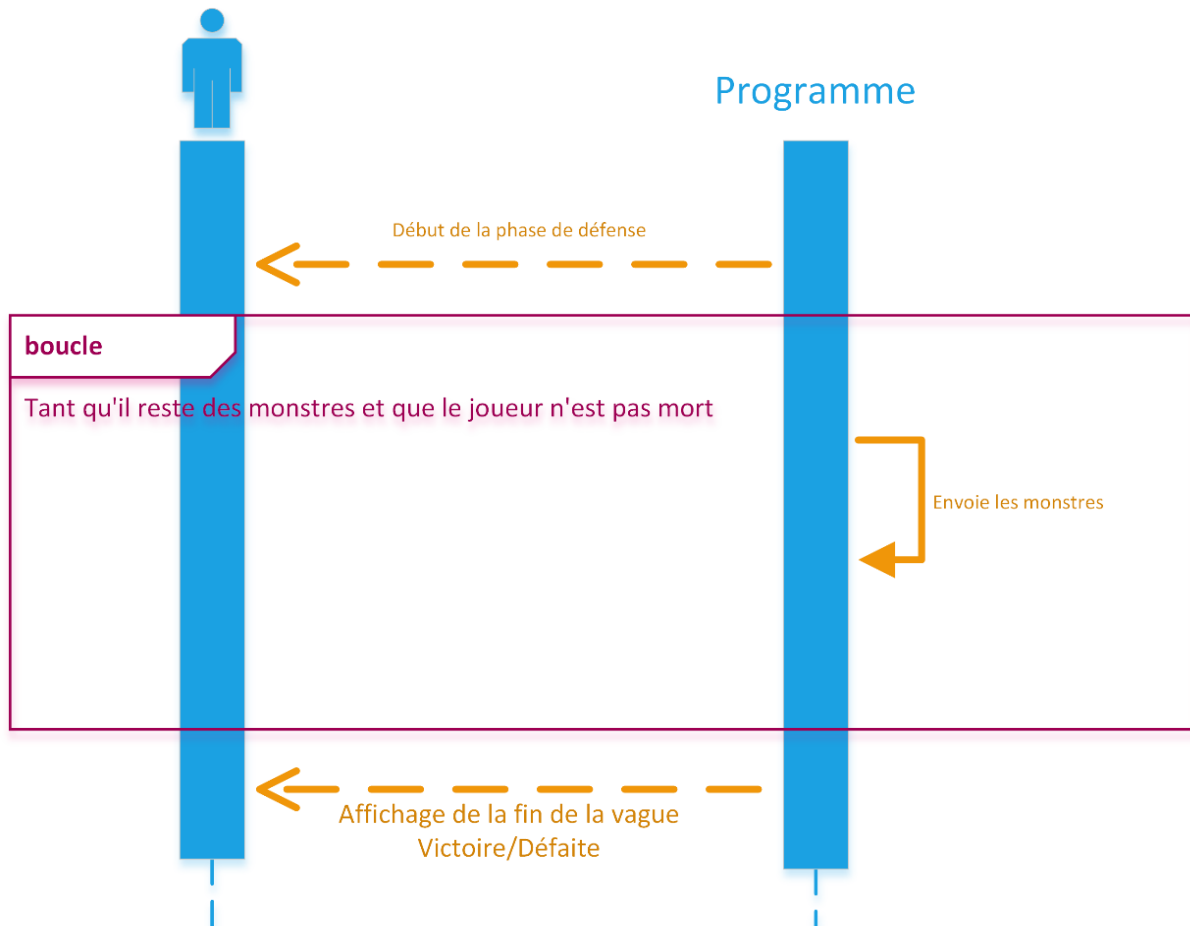
Phase de préparation

Dans cette phase, le joueur peut construire des tours, les mettre à jour, ou les détruire. Le joueur valide sa configuration de tour en cliquant sur valider.



Phase de défense

Une fois la phase de préparation terminée, la vague de monstre arrive. Le joueur observe donc ses tours en action.



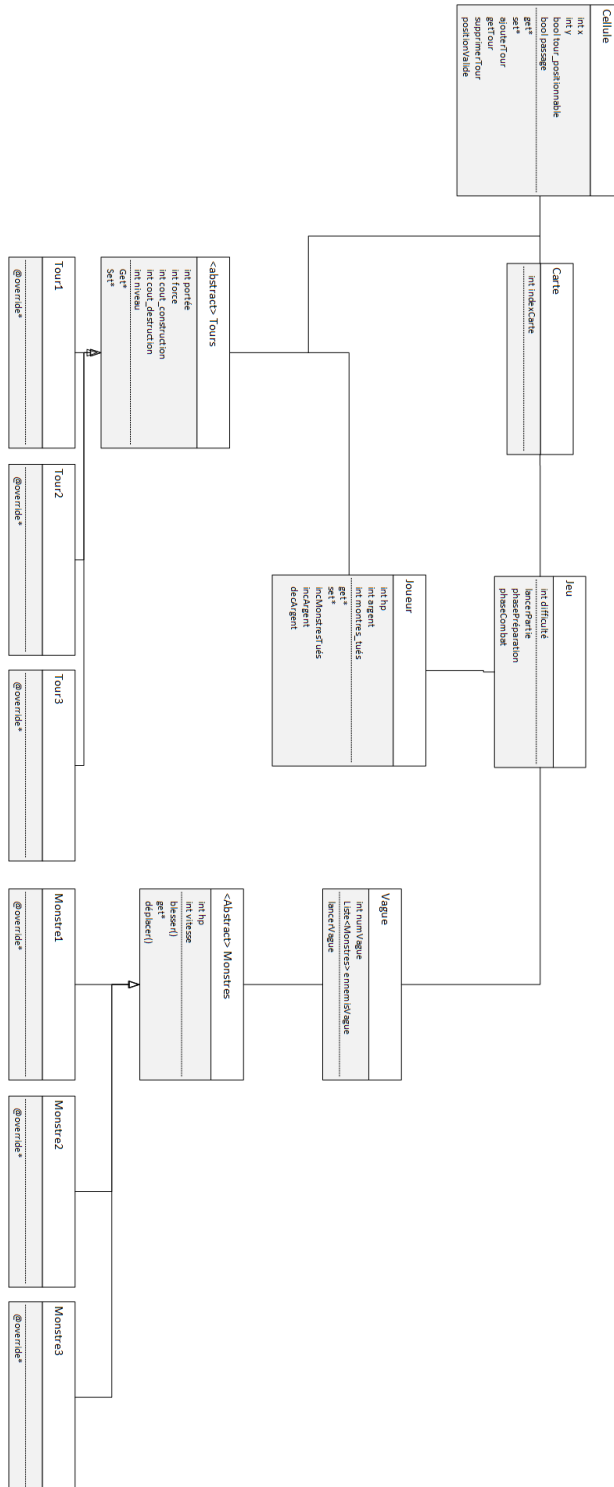
Fin de partie

En cas de victoire, le joueur se verra afficher une page de victoire.

En cas de défaite, il sera demandé au joueur s'il vaut rejouer sur cette carte ou revenir au menu principal.

Organisation et structures de données

Nota bene : ce schéma est actuellement une ébauche, des modifications et des précisions seront certainement introduites en phase de production.



Planning prévisionnel

