

Polybasite

Cahier des charges

STAMEGNA Clément

MICHON Guillaume



Sommaire

I.	Introduction	<i>page.3</i>
	➤ Objet	<i>page. 3</i>
	➤ Contexte	<i>page. 3</i>
	➤ Terminologie	<i>page. 3</i>
II.	Description de l'application	<i>page.4 - 7</i>
	➤ Description générale	<i>page. 4</i>
	➤ Description des besoins fonctionnels	<i>page. 5 - 6</i>
	➤ Analyse technique	<i>page. 7</i>
III.	Prestations attendues	<i>page.8 - 12</i>
	➤ Présentation des prestations attendues	<i>page. 8</i>
	➤ Exigences	<i>page. 8</i>
	➤ Suivi de réalisation - Client	<i>page. 8</i>
	➤ Annexes	<i>page. 9 - 12</i>

I. Introduction

Objet :

Réaliser un programme de jeu à composante jeu de la vie.

« Le jeu de la vie est un automate cellulaire imaginé par John Horton Conway en 1970 qui est probablement, au début du XXIe siècle, le plus connu de tous les automates cellulaires. Malgré des règles très simples, le jeu de la vie est Turing-complet. »

https://fr.wikipedia.org/wiki/Jeu_de_la_vie

Contexte :

Le projet est un projet de POO (programmations orientées objets) de 4eme Année de filière Informatique de l'école Polytech Marseille. Réaliser par STAMEGNA Clément et MICHON Guillaume en binôme pour le client Mr MAVROMATIS Sébastien.

Terminologie :

- **POO** : programmation orienté objet.
- **IHM** : interface homme - machine.
- **IA** : intelligence artificielle

II. Description de l'application

Description générale :

Le but de ce jeu autonome, permettre à plusieurs IA de s'affronter selon des règles et les mécaniques de jeu. Chaque IA possède un groupe de mineur avec une certaine puissance leur but est de miner des minerais. L'IA qui possède le plus de puissance ou le plus de terrain l'emporte lorsque le nombre de tours maximum est atteint ou si une IA éradique tous ses adversaires.

La partie commence, on affiche la grille de jeu en deux dimensions. Cette grille est remplie de minerai et possède un/plusieurs mineur(s) contrôlé par l'IA.

C'est alors au tour des IA d'agir selon une liste d'actions possibles pour leurs mineurs.

La partie se déroule en plusieurs tours de jeu, les IA définissent l'action de chaque mineur pour le tour.

La fin du jeu se décide au bout d'un nombre fixé de tour, ou bien lorsqu'une IA vainc ses adversaires.

Description des besoins fonctionnels:

Voici les différents points fonctionnels du jeu :

- la Map

La Map est une grille constituée de plusieurs cases. Elle représente la zone de jeu. Chaque case peut posséder ou un Minerai ou un Mineur ou rien.

- Les Mineurs

Le Mineur est une unité qui appartient à un camp, il peut bouger, miner. Pour miner le Mineur confronte sa puissance à celle du Minerai, s'il gagne, il conquiert la case sinon il meurt.

- Les Minerais

Les Minerais sont des unités neutres, elle ne peut pas agir elle peut seulement se faire miner. Chaque minerai possède une puissance fixée aléatoirement.

- La puissance

La puissance est une valeur que contient chaque entité (Mineur et Minerai), elle permet de déterminer si un Mineur gagne contre un Minerai, elle s'obtient à chaque tour.

- L'IA

L'IA est la logique qui va gérer les actions des mineurs sur la Map.

« *front office* »

- Le jeu devra afficher une première fenêtre « la Map », les éléments (Mineurs et Minerais) sur la grille ainsi que les interactions entre eux. « La Map » est une grille. Cette grille est composée de plusieurs cases, les mouvements se feront de case en case.
- Une deuxième fenêtre de jeu sera utilisée pour afficher des statistiques sur les scores des IA.

« *back office* »

- Le projet utilisera de l'Intelligence Artificielle qui sera utilisée pour jouer et définir les actions des mineurs à chaque tour.
- Les Mineurs peuvent interagir avec les autres mineurs en transférant leurs puissances ou bien avec le minerai en l'attaquant.
- Les Mineurs peuvent se déplacer de case en case.
- La puissance s'obtient à chaque tour, la puissance gagnée à chaque tour dépend de la case.

Analyse technique:

« *front office* »

- Pour la création de la Map on créera un élément Map qui sera une collection de cases. Chaque case pourra contenir une seule entité.
- Les mineurs de départs seront soit positionné aléatoirement, soit positionner par l'utilisateur.
- une seconde fenêtre servira à afficher les informations sur la partie ex : puissance de chaque IA, nombre de tours...

« *back office* »

- Lors d'une attaque d'un Mineur sur un Minerai on vérifiera par une méthode que la puissance du mineur soit supérieure à celle du minerai. L'attaque sera une méthode qui prend en attribut un Minerai d'une case adjacente.
- Les Mineurs posséderont des méthodes pour transférer leurs puissances aux autres mineurs adjacents.
- Chaque case aura un attribut gain de puissance par tour.

Spécificité logicielle :

Le jeu sera réalisé et développé en langage C++ avec la bibliothèque SFML (Simple and Fast Multimedia Library).

III. Prestation attendues

Présentation des prestations attendues :

La prestation attendue comprend le jeu exécutable respectant les fonctionnalités, la documentation contenant des indications sur ce dernier, le document d'analyse, les rapports d'avancement, ainsi qu'un document final.

Exigences :

Le programme doit être un jeu de type jeux de la vie selon ces points.

- ne nécessite aucun joueur.
- Le « jeu » se déroule sur une grille à deux dimensions.

Il doit être développé dans un langage objet tel que C#, C++ ou bien Java.

Suivi de réalisation - Client :

Le projet pourra être suivi par le client via le lien Github :

<https://github.com/PolyDevTeam/Polybasite>

Ou bien par mail via les adresses ci-dessous.

Les contacts du projet sont :

- Les chefs de projet

STAMEGNA Clément - clementstamegna@etu.univ-amu.fr

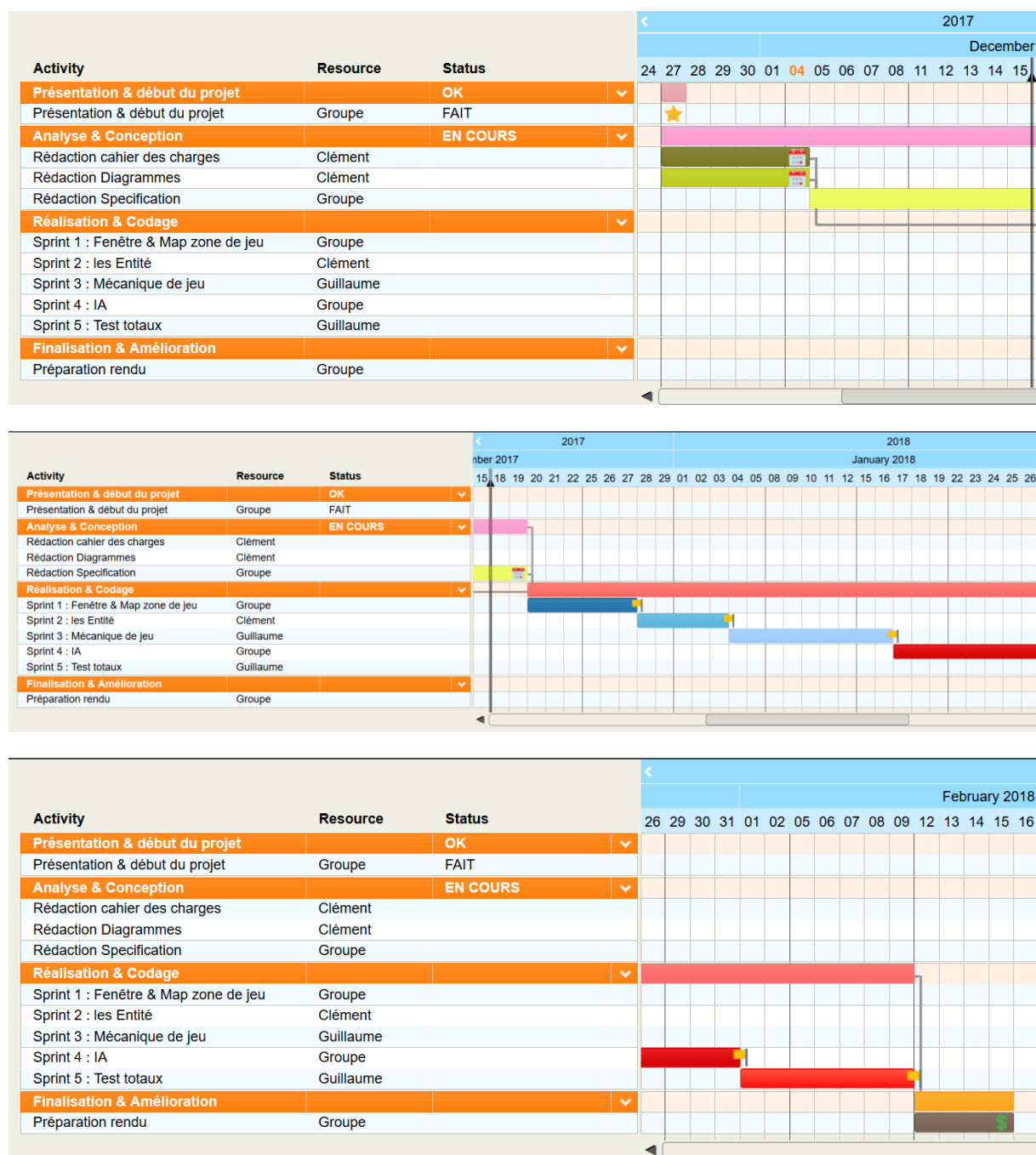
MICHON Guillaume - guillaumemichon@etu.univ-amu.fr

- Le Client

MAVROMATIS Sébastien - sebastienmavromatis@univ-amu.fr

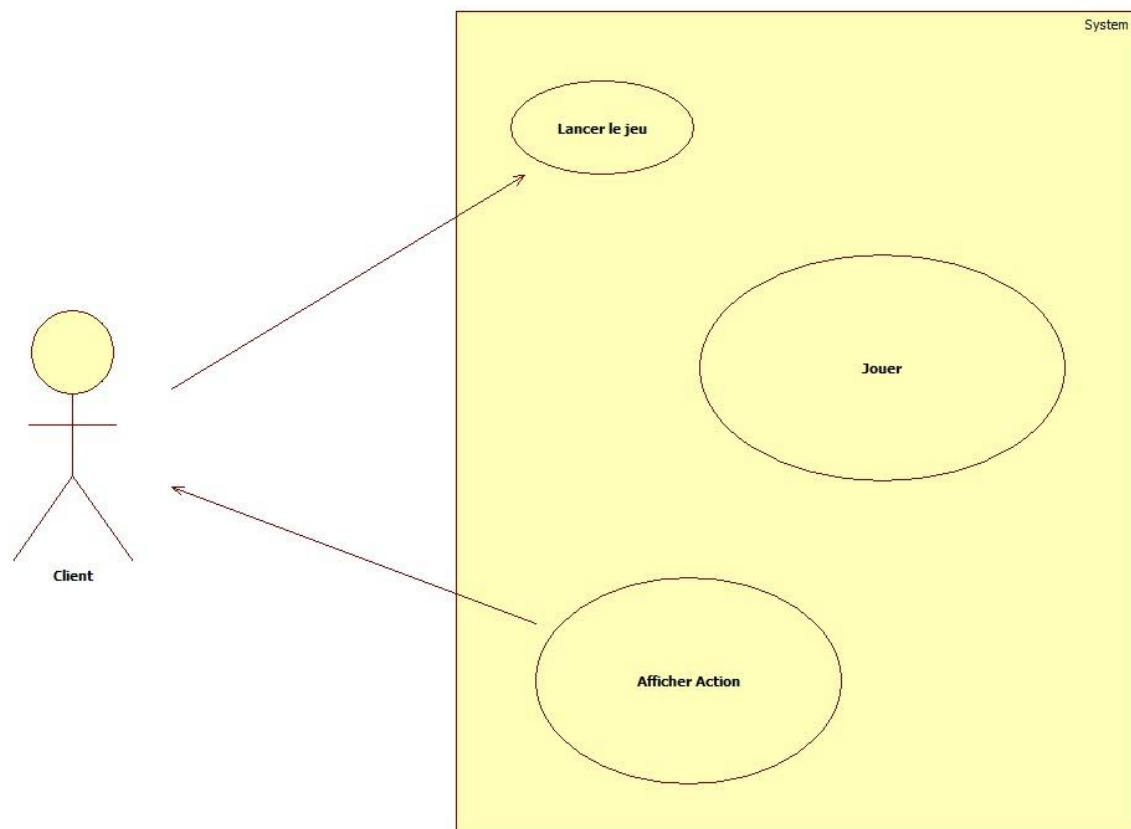
Annexes :

Planning prévisionnel :

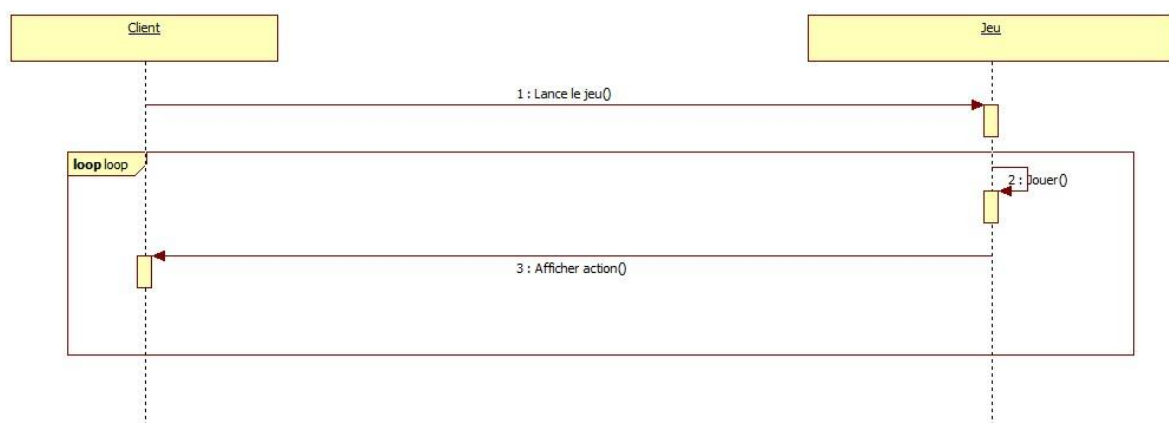


Diagrammes :

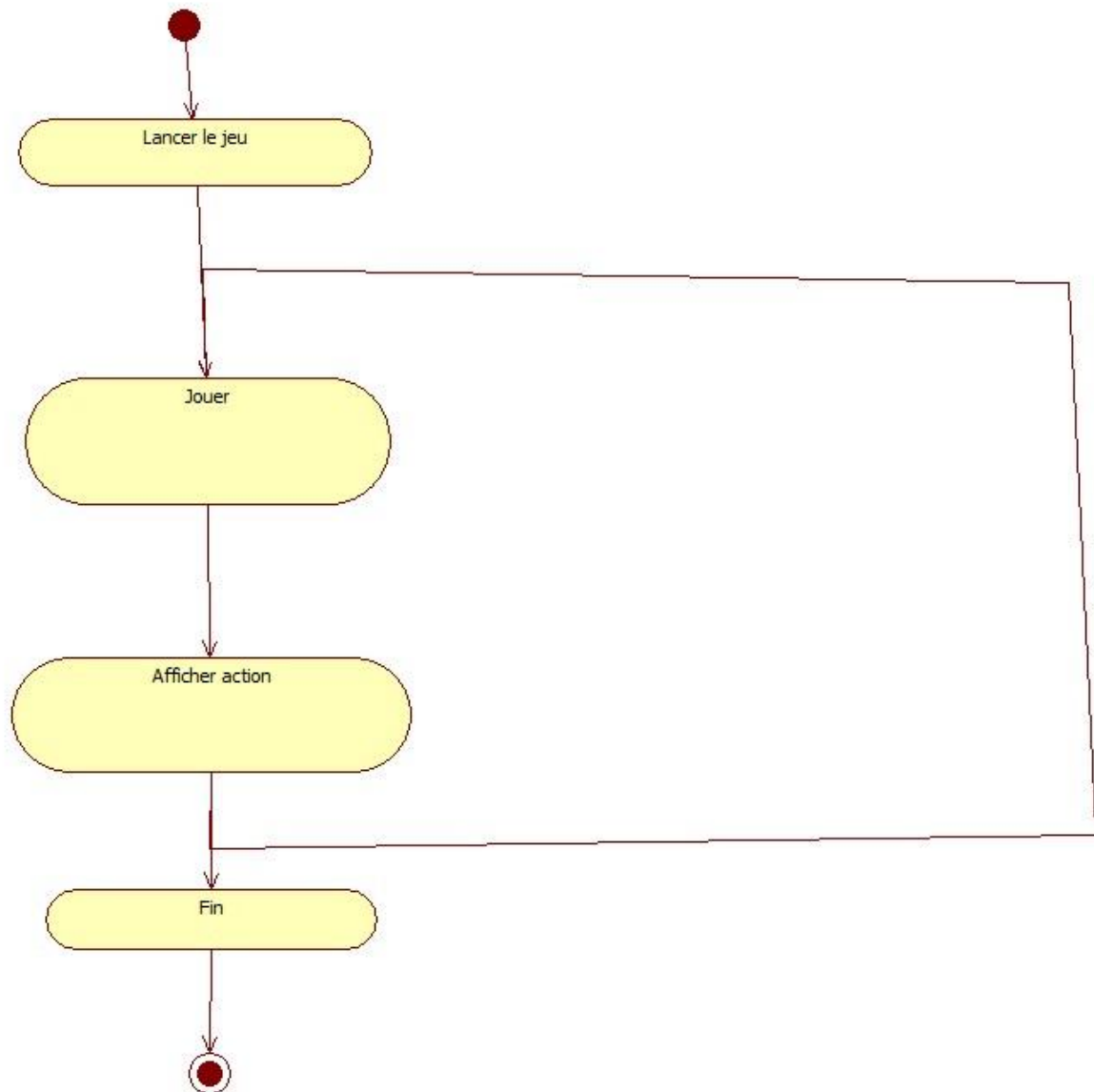
- Diagramme de cas d'utilisation



- Diagramme de Séquence



- Diagramme d'Activité



- Diagramme de Classes

