ÉCOLE NATIONALE D'INGÉNIEURS DE BREST

RÉVÉLATEUR D'INGÉNIEURS
DEPUIS 1961

Cahier des charges _ Projet informatique S2

Type	Cahier des charges
Nom du projet	Robot Rumble
Commentaire	Jeu de tir
Auteur	Carrée Ewan S2P-B
Version	3.0
Date	17/04/2019

Table des matières

1 - Objectifs

Contexte : Dans le cadre du projet informatique en Initiation au Projet Informatique (IPI), à l'ENIB, je souhaite réaliser un jeu de tir qui s'inspire d'un des modes du célèbre jeu mobile Brawl Stars. Ce jeu gratuit développé par Supercell sorti en juin 2017 et est très apprécié du jeune public.

Description générale : «Robot Rumble» est un jeu de survie pour un seul joueur. L'utilisateur peut se mesurer au robot dont le principe du jeu est de tuer le robot ou de survivre le plus longtemps possible.

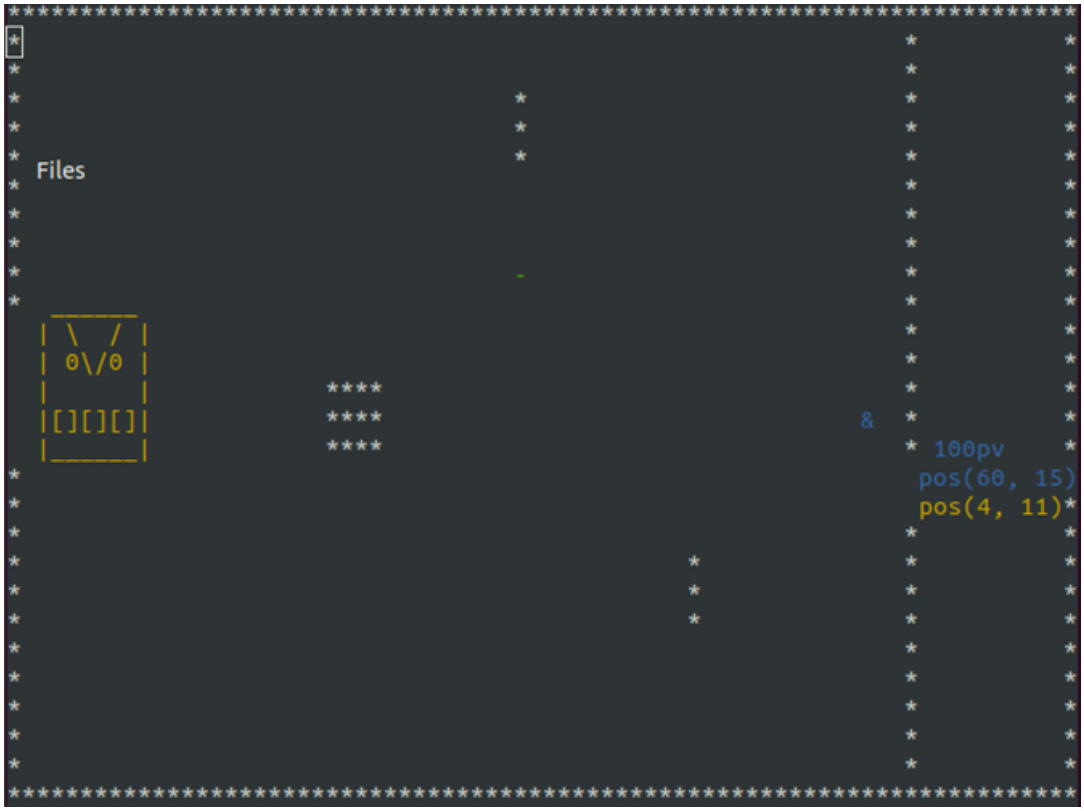
Ce document constitue donc le cahier des charges du jeu de tir : Robot Rumble.

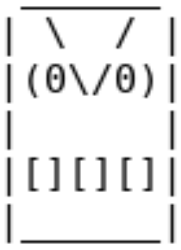
2 - L'expression fonctionnelle du besoin

Règles du jeu : Le joueur doit se déplacer sur la carte tout en évitant le robot afin de ne pas prendre des dégâts car le robot essaye de le tuer. Il doit aussi éviter les murs qui lui inflige des dégâts lorsqu'il entre en contact avec.
Ce joueur possède une arme qui lui permet d'infliger aléatoirement des dégâts allant de 5 à 10 points de vie au robot. Cette arme est une arme à feu qui prend la forme que l'imagination de l'utilisateur lui donne. Pour faciliter le jeu, cette arme possède une réserve de munitions illimitée qui lui permet de tirer sans penser à comment gérer ses réserves. Cette arme tire dans une seule direction sur l'axe horizontal.
Au fur et à mesure des dégâts que le robot prend, celui-ci s'énerve et ses capacités se décuplent, il gagne en vitesse ce qui rend la tâche de l'utilisateur plus difficile car le robot se déplace très vite et il est plus difficile de lui tirer dessus. Le robot gagne aussi en puissance et ses coups affligent plus de dégâts au joueur. Il y aura trois paliers d'énervement allant de 100pv à 60pv, de 60pv à 10pv et de 10pv à 0pv.
Les éléments qui se déplacent comme le robot qui se déplace aléatoirement sur l'axe vertical et le joueur qui se déplace en fonction des volontés du joueur doivent rester dans la zone du jeu délimitée par des asterix.
Un chronomètre pourrait être ajouté afin d'observer les performances. Le joueur peut s'amuser à essayer de battre ce temps la partie d'après.
Sur la carte, nous pouvons retrouver des obstacles qui prennent différentes formes et créés aussi avec des asterix. Ces obstacles permettent au joueur de se protéger afin d'éviter de prendre des coups avec le robot. Ils peuvent s'avérer utiles lorsque le robot atteint sa vitesse maximale.
La forme des éléments du jeu peut être sujette à des modifications telle que le graphisme du robot afin de mieux s'adapter au code.

L’interface utilisateur :
Visuel : Pendant la partie, l'utilisateur voit la zone dans laquelle il peut jouer, son personnage et le robot. Sur la partie droite, on peut aussi retrouver les informations de la partie comme les points de vie par exemple. L’ensemble est en mouvement en fonction du temps.
Une mention de fin de partie doit afficher le temps final avec la mention «Gagné» si le joueur est parvenu à tuer le robot ou «Perdu» si ce joueur s’est fait tuer par le robot.

Voici un exemple de ce que le joueur verra :



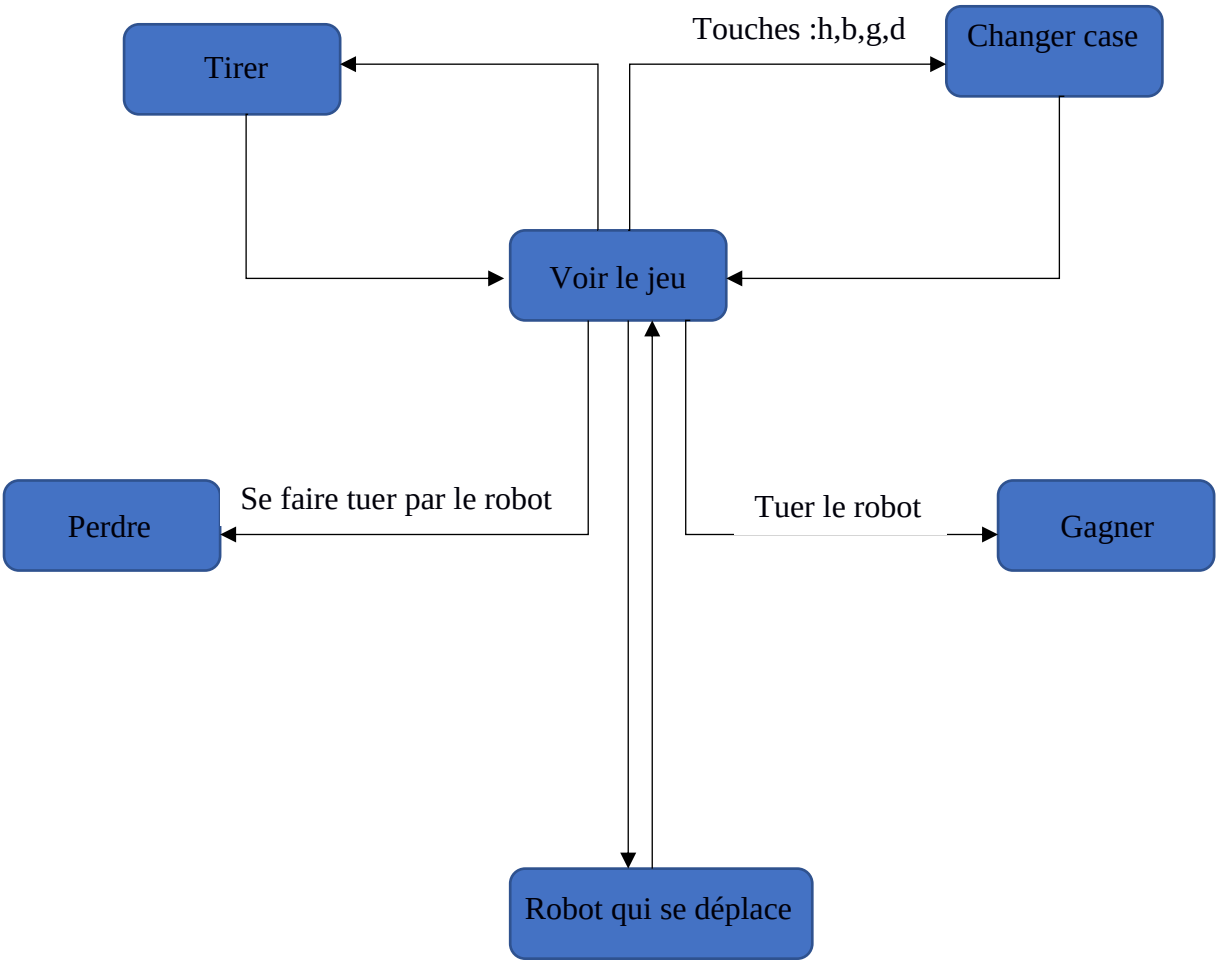


Interactions : L'utilisateur interagit uniquement avec le clavier :
Les touches «h», «b», «g», «d» permettent de se déplacer (haut, bas, gauche, droite).

Manuel utilisateur :
Lancer le jeu : *python Main.py* (dans un terminal)
Le terminal doit être mit en mode plein écran pour pouvoir jouer au jeu.
Pour quitter le jeu, l’utilisateur doit appuyer sur la toucher *échap* du clavier.
Jouer : h,b,g,d pour se déplacer

Contraintes techniques : Le jeu doit pouvoir être joué dans le terminal Linux. Il doit aussi fonctionner sur les ordinateurs de l’ENIB. Ce jeu doit être codé en langage python 2.7
Enfin, le jeu doit obligatoirement répondre à la condition générale imposée qui dit que le jeu doit être un jeu de tire. Il doit donc obligatoirement comporter une fonction qui permet de tirer.

Scénario d'utilisation :



3 – Analyse du besoin

Fonctionnalités :

F1 : jouer une partie
- F1.1 : afficher le jeu : grille, informations
- F1.2 : se déplacer sur la grille
- F1.3 : tirer
- F1.4 : présence d'un chronomètre
- F1.5 : présence et gestion d'obstacles dans la zone
- F1.6 : évolution du robot

F2 : finir une partie
- F2.1 : mention de fin de partie

Critères de validité et de qualité :

Validation : Le projet sera considéré comme validé si le joueur peut jouer une partie en utilisant toutes les fonctionnalités présentées précédemment. Il doit pouvoir se déplacer comme il le souhaite et tirer. La fin de la partie doit être claire et montrer qui a gagné.

Qualité : L'interface devra être épurée pour ne pas déranger l'utilisateur.
Le code doit s'exécuter correctement et doit permettre de contrer le maximum de problèmes que le joueur peut rencontrer comme des bugs ou des éléments qui fonctionnent anormalement. Le code doit être en accord avec les méthodes apprises durant les différentes UC de cours d'IPI que nous avons eu depuis le début de l'année. Il doit prendre en considération toutes les techniques vues avec les différents objectifs et doit se baser sur un codage typique des évaluations que nous avons dû réaliser.

Importance des fonctionnalités :

0 : indispensable
1 : forte valeur ajoutée au projet
2 : optionnelle

F1 : jouer une partie	0
F1.1 : afficher le jeu : grille, informations	0
F1.2 : se déplacer sur la grille	0
F1.3 : tirer	0
F1.4 : présence d'un chronomètre	2
F1.5 : présence et gestion d'obstacles dans la zone	1
F1.6 : évolution du robot	2
F2 : finir une partie	0
F2.1 : mention de fin de partie	1

4 - Livrables

Echéancier : les UC seront optimisées afin de réaliser le projet dans les meilleures conditions. Tout d'abord il y aura la conception de ce cahier des charges, puis celui de conception. Le tout en parallèle du codage qui se réalisera en même temps et aussi en dehors des cours.

Description des livrables :

- Cahier des charges
- Cahier de conception
- Prototypes (variant en fonction des différentes UC)
- Version finale contenant tous les points précédents.