Cahier des charges

Tower Defense

Nathan Calvarin et Maxime Delin

28 mars 2014

Table des matières

1 Objectifs **2**
1.1 Description générale . . . 2
1.2 Contexte . . . 2

2 Expression du besoin **2**
2.1 Règles du jeu . . . 2
2.2 L'interface utilisateur . . . 2
2.2.1 Visuel . . . 2
2.2.2 Interaction . . . 2
2.3 Manuel utilisateur . . . 2
2.4 Contraintes techniques . . . 3
2.5 Scénario d'utilisation . . . 3
2.5.1 S0 : Scénario principal . . . 3
2.5.2 S1 : Débuter une manche . . . 3
2.5.3 S2 : Améliorer tour . . . 4

3 Analyse du besoin : **4**
3.1 Fonctionnalités . . . 4
3.2 Critères de validité et de qualité . . . 5
3.2.1 Validation . . . 5
3.2.2 Qualité . . . 5
3.2.3 Importance des fonctionnalités . . . 5

4 Livrables **5**
4.1 Échéancier . . . 5
4.2 Description des livrables . . . 6
4.2.1 CDC : Cahier des charges . . . 6
4.2.2 C1 : Document de conception v1.0 . . . 6
4.2.3 P1 : Prototype P1 . . . 6
4.2.4 P2 : Prototype P2 . . . 6
4.2.5 VF : Version finale . . . 6

1 Objectifs

1.1 Description générale

Dans le cadre de notre cours de Méthode de Développement nous souhaitons recréer un jeu dans l'esprit d'un *Tower Defense*.

1.2 Contexte

2 Expression du besoin

Tower Defense est un jeu pour un joueur dans lequel celui-ci doit disposer des tours dans le but de tuer des vagues de monstres. A chaque monstre éliminé revient au joueur une somme d'argent qui lui permet de racheter de nouvelles tours.

2.1 Règles du jeu

- Le but est d'éliminer les vagues de monstres à l'aide de tours disposée par l'utilisateur.
- Il y a un joueur : l'utilisateur.
- L'ordinateur envoie des monstres par vague qui ont pour but de traverser la *map*.
- Le jeu débute avec une *map* vide.
- Le joueur dispose d'une somme d'argent de départ et un certain nombre de vies.
- A chaque fois qu'un monstre atteint le bout de la *map* le joueur perd une vie.
- Lorsqu'une tour est placée, le joueur peut choisir de l'améliorer.

2.2 L'interface utilisateur

2.2.1 Visuel

Pendant la partie, l'utilisateur voit la *map*, son nombre de vie, son nombre argent, les vagues à venir, les tours qu'il peut choisir de disposer.
Un écran de sortie affiche la mention *Gagné* ou *Perdu*, et le score du joueur,

2.2.2 Interaction

L'utilisateur interagit uniquement avec le clavier : il se sert des flèches du clavier et des touches *f*, *g*, *p* et *espace*.

2.3 Manuel utilisateur

- **Lancer le jeu** : $python towerdefense.py
- **Saisir son nom** : taper son nom puis sur *Entrée*
- **Choisir un niveau** : avec les flèches du clavier.
- **Jouer** : séletionner type de tour à l'aide des touches du clavier, poser tour en appuyant sur *espace*, sélectionner tour déjà poser pour l'améliorer
- **Types de tour** :

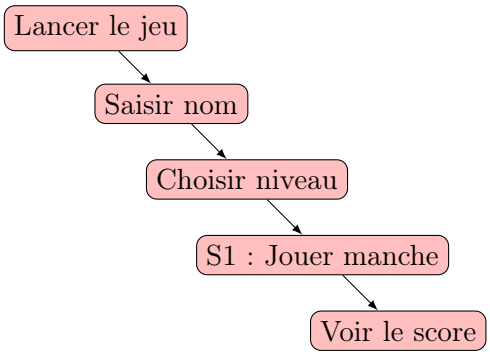
– **Tour feu** : l'utilisateur appuie sur la touche f pour sélectionner ce type de tour.
– **Tour glace** : l'utilisateur appuie sur la touche g pour sélectionner ce type de tour.
– **Tour poison** : l'utilisateur appuie sur la touche p pour sélectionner ce type de tour.

2.4 Contraintes techniques

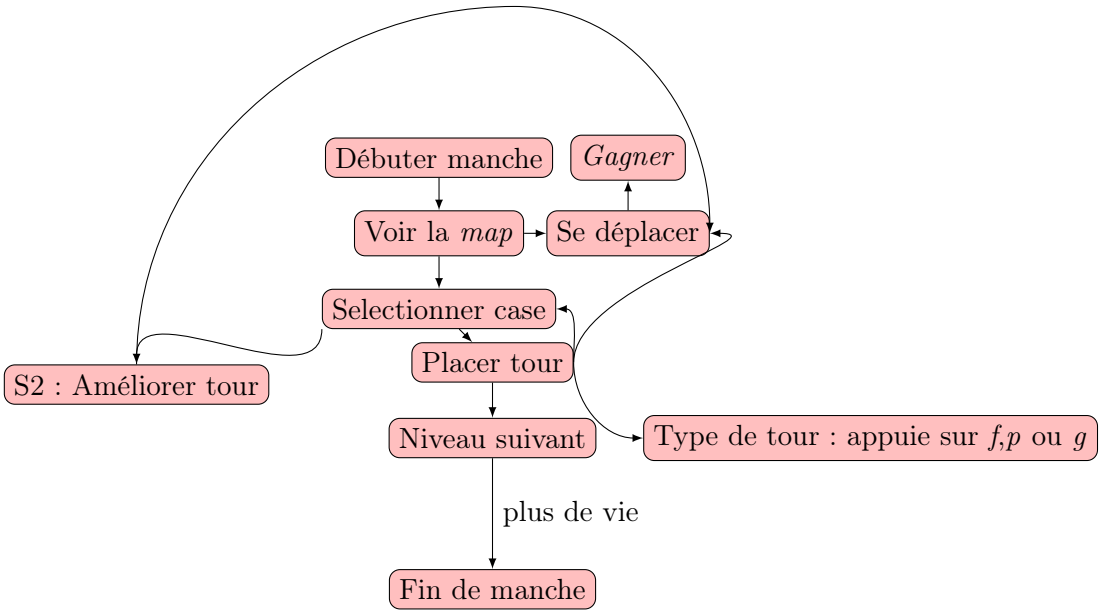
- Le logiciel est associé à un cours, il doit donc fonctionner sur les machines de TP de l'ENIB pour que es élèves puissent le tester.
- Le langage utilisé en cours est Python. Le développement devra donc se faire en python.
- Les notions de programmation orientée objet n'ayant pas encore été abordées, le programme devra essentiellement s'appuyer sur le paradigme de la programmation procédurale.
- Le logiciel devra être réalisé en conformité avec les pratiques préconisées en cours de MDD : barrière d'abstraction, modularité, unicode, etc...
- L'interface sera réalisée en mode texte dans un terminal.

2.5 Scénario d'utilisation

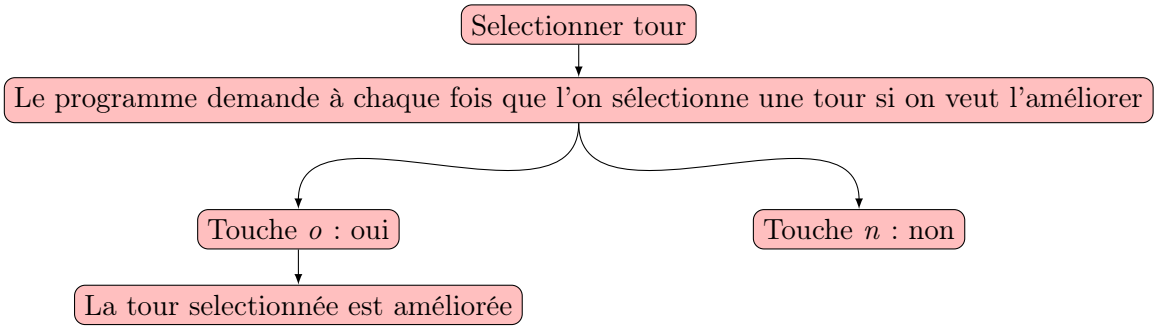
2.5.1 S0 : Scénario principal



2.5.2 S1 : Débuter une manche



2.5.3 S2 : Améliorer tour



3 Analyse du besoin :

3.1 Fonctionnalités

- F1 : Nommer le joueur
- F2 : Choisir le niveau
- F3 : Jouer une partie
  - F3.1 : Joueur un niveau
    - F3.1.1 Afficher le jeu
      - map
      - nom
      - niveau
      - score
      - case sélectionnée
      - nombre de monstres restants
      - différentes tours disponibles
      - argent
    - F3.1.2 Sélectionner une tour
    - F3.1.3 Se déplacer dans la map
    - F3.1.4 Placer une tour
    - F3.1.5 Améliorer une tour
    - F3.1.6 Finir manche
  - F3.2 Finir partie
    - F3.2.1 Afficher le résultat
    - F3.2.2 Quitter

3.2 Critères de validité et de qualité

3.2.1 Validation

Le logiciel sera validé de la manière suivante :

- Le code doit s'exécuter correctement en suivant les instructions livrées avec le logiciel.
- L'utilisation du logiciel permettra de constater que les fonctionnalités ont été bien implémentées.

3.2.2 Qualité

Différents critères permettront d'évaluer la qualité du jeu :

- La jouabilité : l'interface devra être suffisamment ergonomique pour permettre au joueur d'enchainer rapidement les manches.
- La robustesse.
- Le respect des méthodes de conception et de codage données en cours de MDD.

3.2.3 Importance des fonctionnalités

0 : Indispensable
1 : Forte valeur ajouté au projet
2 : Optionnelle

F1 : Nommer le joueur	2
F2 : Choisir le niveau	1
F3 : Jouer une partie	0
F3.1 : Jouer un niveau	0
F3.1.1 : Afficher le jeu	0
F3.1.2 : Sélectionner une tour	0
F3.1.3 : Se déplacer sur la map	0
F3.1.4 : Placer une tour	0
F3.1.5 : Améliorer une tour	1
F3.1.6 : Fin du niveau	0
F3.2 : Fin de la partie	0
F3.2.1 : Afficher le résultat	1
F3.2.2 : Quitter	0

4 Livrables

4.1 Échéancier

1ère semaine : cahier des charges
2ème semaine : conception
7ème semaine : P1 + version finale
Les livrables seront envoyé par mail en version électronique.
Les documents texte seront au formats *.pdf*

4.2 Description des livrables

4.2.1 CDC : Cahier des charges

Expression et analyse du besoin.
Fichier : *towerDefense_ delinCalvarin-CdC.pdf*

4.2.2 C1 : Document de conception v1.0

Fichier *towerDefense_ delinCalvarin-Conception.pdf*

4.2.3 P1 : Prototype P1

Ce prototype porte essentiellement sur la création de la map et sur l'affichage.
Mise en oeuvre des fonctionnalités : F1, F2, F3.1.1, F3.1.2, F3.1.3, F3.1.4, F3.1.5
Livré dans une archive au format *.zip* ou *.tgz*
Contient un manuel d'utilisation dans le fichier *readme.txt*

4.2.4 P2 : Prototype P2

Ce prototype réalise toutes les fonctionnalités.
Ajout à P1 des fonctionnalités F3.1.6, F3.2
Livré dans une archive au format *.zip* ou *.tgz*
Contient un manuel d'utilisation dans le fichier *readme.txt*

4.2.5 VF : Version finale

Archive finale au format *.zip* ou *.tgz*
La version finale contient toutes les versions des documents : cahier des charges, conception, et les deux prototypes.