

# ÉCOLE NATIONALE DES INGÉNIEURS DE BREST

CAHIER DES CHARGES

MDD-PROJET

---

## Spazz

---

Noé MAILLARD et Allan DANO

26 mai 2014

Version 1.1

## Table des matières

<b>1</b>	<b>Objectifs</b>	<b>3</b>
1.1	Description générale . . . . .	3
<b>2</b>	<b>Expression du besoin</b>	<b>3</b>
2.1	Règles du jeu . . . . .	3
2.2	L'interface . . . . .	3
2.2.1	Visuel . . . . .	3
2.2.2	Interaction . . . . .	3
2.3	Manuel Utilisateur . . . . .	3
2.4	Contraintes techniques . . . . .	4
2.5	Scénario d'utilisation . . . . .	4
<b>3</b>	<b>Analyse du besoin</b>	<b>5</b>
3.1	Fonctionnalités . . . . .	5
<b>4</b>	<b>Critères de validité et de qualité</b>	<b>5</b>
4.1	Validation . . . . .	5
4.2	Qualité . . . . .	5
<b>5</b>	<b>Livrables</b>	<b>5</b>
5.1	Echéancier . . . . .	5
5.2	Description des livrables . . . . .	6

# 1 Objectifs

## 1.1 Description générale

Dans le cadre du cours de MDD, nous avons eu l'idée de créer un jeu vidéo mélangeant le concept du Snake et celui de jeu par niveau.

# 2 Expression du besoin

## 2.1 Règles du jeu

- l'utilisateur déplace une entité nommée Spazz qui doit se déplacer dans le niveau en ramassant des jetons
- À chaque fois que Spazz ramasse un jeton, il grandit en taille
- À la fin d'un niveau, le nombre de points est le nombre de jetons ramassés par Spazz.
- le niveau est terminé lorsque Spazz se cogne contre un mur ou lui-même.

Le but du jeu est de manger le plus de jetons dans un niveau

## 2.2 L'interface

### 2.2.1 Visuel

Dans le niveau, l'utilisateur voit les différents items du menu principal et une flèche qui indique quel item est sélectionné. Pendant une partie l'utilisateur voit le niveau en cours avec Spazz qui se déplace en fonction des touches appuyées, en dessous il y a une banderole qui affiche le score total, le temps restant et les nombre de jetons restants. Dans l'éditeur de niveaux, l'utilisateur voit le niveau, le curseur et deux nombres indiquant la position de son curseur et le mode dans le quel il se trouve.

### 2.2.2 Interaction

Dans le menu, lorsqu'il y a plusieurs items, l'utilisateur navigue à l'aide des touches Z et S puis confirme par la touche E, lorsque l'utilisateur entre du texte, il confirme avec la touche entrée. Pendant une partie, l'utilisateur agit sur le Spazz l'aide des touches Z , Q , S et D. Ces touches changent la direction dans laquelle Spazz se dirige. Dans l'éditeur de niveaux, l'utilisateur déplace le curseur à l'aide des touche Z, Q, S et D. en appuyant sur la barre d'espace il change le mode du curseur.

## 2.3 Manuel Utilisateur

**Lancer le Jeu :** \$python Spazz

**Choisir le nom :** dans le menu, sélectionner l'item "Change name", entrer le nom choisi et confirmer.

**Choisir la difficulté :** dans le menu, sélectionner l'item "Change Difficulty", et entrer 1, 2 ou 3 en fonction de la difficulté désirée.

**Choisir le niveau :** dans le menu, sélectionner l'item "Select level", entrer le numéro du niveau désiré puis confirmer.

**Jouer le niveau :** dans le menu, sélectionner l'item "play".

**Quitter le Jeu :** dans le menu, sélectionner l'item "quit game".

**Diriger le Spazz :** dans le jeu, appuyer sur les touches Z, Q, S, ou D pour changer la direction dans laquelle le Spazz se dirige

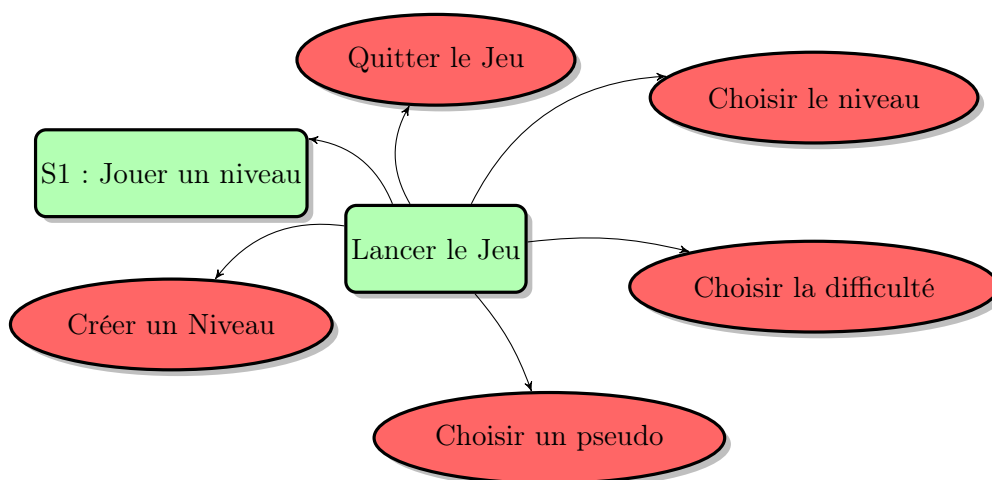
**Créer un niveau :** dans l'éditeur de niveau, appuyer sur les touches Z, Q, S ou D pour déplacer le curseur, appuyer sur la barre d'espace pour commencer à placer des murs ou effacer des murs déjà placés et ré-appuyer sur la barre d'espace pour seulement déplacer le curseur. Appuyer sur la touche entrée pour sauvegarder le niveau et revenir au menu.

## 2.4 Contraintes techniques

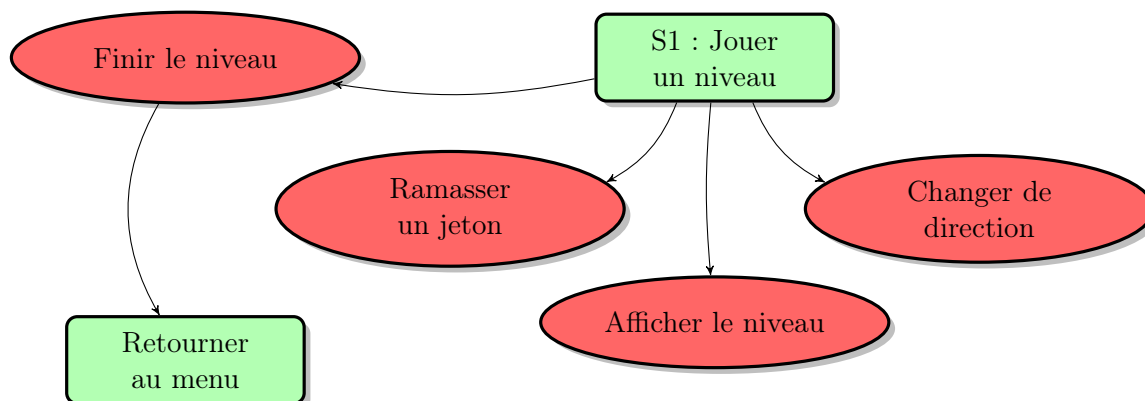
- Le logiciel crée est évalué par les professeurs sur un ordinateur de salle de TP, il faut donc que le jeu s'exécute et soit jouable sur ces machines
- Le cours porte sur le langage Python, il est donc évident que le jeu soit écrit en Python
- Le paradigme utilisé est celui de la programmation procédurale
- L'interface doit être en mode texte dans le terminal

## 2.5 Scénario d'utilisation

**S0 : Scénario du menu**



**S1 : Jouer un niveau**



## 3 Analyse du besoin

### 3.1 Fonctionnalités

- F1** : Choisir un pseudo
- F2** : Choisir la difficulté
- F3** : Choisir le niveau
- F4** : Jouer un niveau
  - F4.1** : Afficher le Jeu
  - F4.2** : Changer de direction
  - F4.3** : Ramasser un jeton
  - F4.4** : Finir le niveau
- F5** : Afficher les meilleurs scores
- F6** : Quitter le jeu
- F7** : Créer des Niveaux

## 4 Critères de validité et de qualité

### 4.1 Validation

La validation du logiciel sera validée 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 bien été implémentées.

### 4.2 Qualité

Différent critères permettront d'évaluer la qualité du jeu :

- La jouabilité : L'interface devra être suffisamment ergonomique pour permettre au joueur de jouer un niveau sans difficultés techniques (Lenteur, bugs).
- La robustesse.
- Le respect de méthodes de codage données en cours de MDD.

## 5 Livrables

### 5.1 Echancier

Avant la 3<sup>ème</sup> UC : Prototype P1  
Avant la 5<sup>ème</sup> UC : Prototype P2  
Avant la Dernière UC : Version Finale

## 5.2 Description des livrables

**Cahier des Charges** Expression et analyse du besoin.

Fichier : `Cahier_des_charges.pdf`

**Document de conception** Fichier : `Document_de_conception-vx.x.pdf`

**Code Source** Fichiers contenus dans le dossier `src`

**Prototype P1** Ce prototype porte sur la création et l'affichage d'un niveau.

Mise en œuvre de fonctionnalités : F1, F2, F3.2, F3.2

Fichier : `Spazz-P1.tgz`

**Version finale** Fichier : `Spazz-v1.0.tgz`