

CAHIER DES CHARGES

SHOOTER3000 S2

Type	Cahier des charges
Nom du projet	Shooter3000
Commentaire	Projet IPI, S2, ENIB
Auteur	Ewen Daniel (e3daniel@enib.fr)
Version	1.0
Date	07/02/24

Table des matières

1 Objectifs.....	3
1.1 Description générale	3
1.2 Contexte.....	3
2 Expression du besoin.....	3
2.1 Règles du jeu.....	3
2.2 L'interface utilisateur.....	3
2.2.1 Visuel.....	3
2.2.2 Interaction.....	4
2.3 Manuel utilisateur.....	4
2.4 Contraintes techniques.....	4
2.5 Scénario d'utilisation.....	5
3 Analyse du besoin :.....	6
3.1 Fonctionnalités.....	6
3.2 Critères de validité et de qualité.....	6
3.2.1 Validation.....	6
3.2.2 Qualité	6
3.2.3 Importance des fonctionnalités.....	7
4 Livrables.....	8
4.1 Echéancier.....	8
4.2 Description des livrables.....	8
4.2.1 CDC : Cahier des charges.....	8
4.2.2 Conception : Document de conception v1.0.....	8
4.2.3 Projet : Code + Document de conception v2.0.....	8
4.2.4 Code Objet :	8

1 Objectifs

1.1 Description générale

Dans la lignée du Projet Informatique(IPI), du S2, je souhaite réaliser un jeu de shooter. Ce document constitue le cahier des charges du jeu « Shooter3000 ».

1.2 Contexte

Le cours IPI du semestre 2 propose de mettre en œuvre un cycle de développement informatique complet. Ici le type de jeu vidéo est assez complexe avec une visuel basique et des variables moyennes.

2 Expression du besoin

« Shooter 3000 » est un shooter qui reprend le principe du jeu Galaga.

Il y a des adversaires de plus en plus nombreux en fonction du niveau, un vaisseau qui se défend et un nombre de vies

2.1 Règles du jeu

Je m'appuie sur les règles classiques de jeu tels que Space Invader ou Galaga :

- Les tirs adverses rebondissent une fois sur le sol derrière le vaisseau et les murs, plus de rebonds quand le niveau augmente(version difficile du jeu).
- Le joueur pilote le vaisseau
- Si le vaisseau est touché, on perd une vie.
- Si le tire touche un des vaisseaux adversaires alors il est éliminé.
- Quand un vaisseau adverse est touché, il disparaît
- Quand il n'y a plus d'ennemis dans le niveau on passe au niveau suivant
- Quand on a terminé tout les niveaux, on a gagné
- Quand on n'a plus de vie, la partie s'arrête

2.2 L'interface utilisateur

2.2.1 Visuel



2.2.2 Interaction

L'utilisateur interagit uniquement avec le clavier :

- La flèche gauche ou la touche « q » permettent de déplacer le vaisseau vers la gauche
- La flèche de droit ou la touche « d » permettent de déplacer le vaisseau vers la droite
- La flèche avant ou la touche « z » permettent de déplacer le vaisseau vers l'avant
- La flèche arrière ou la touche « s » permettent de déplacer le vaisseau vers l'arrière
- Le clique de la souris ou la touche « espace » permet de tirer

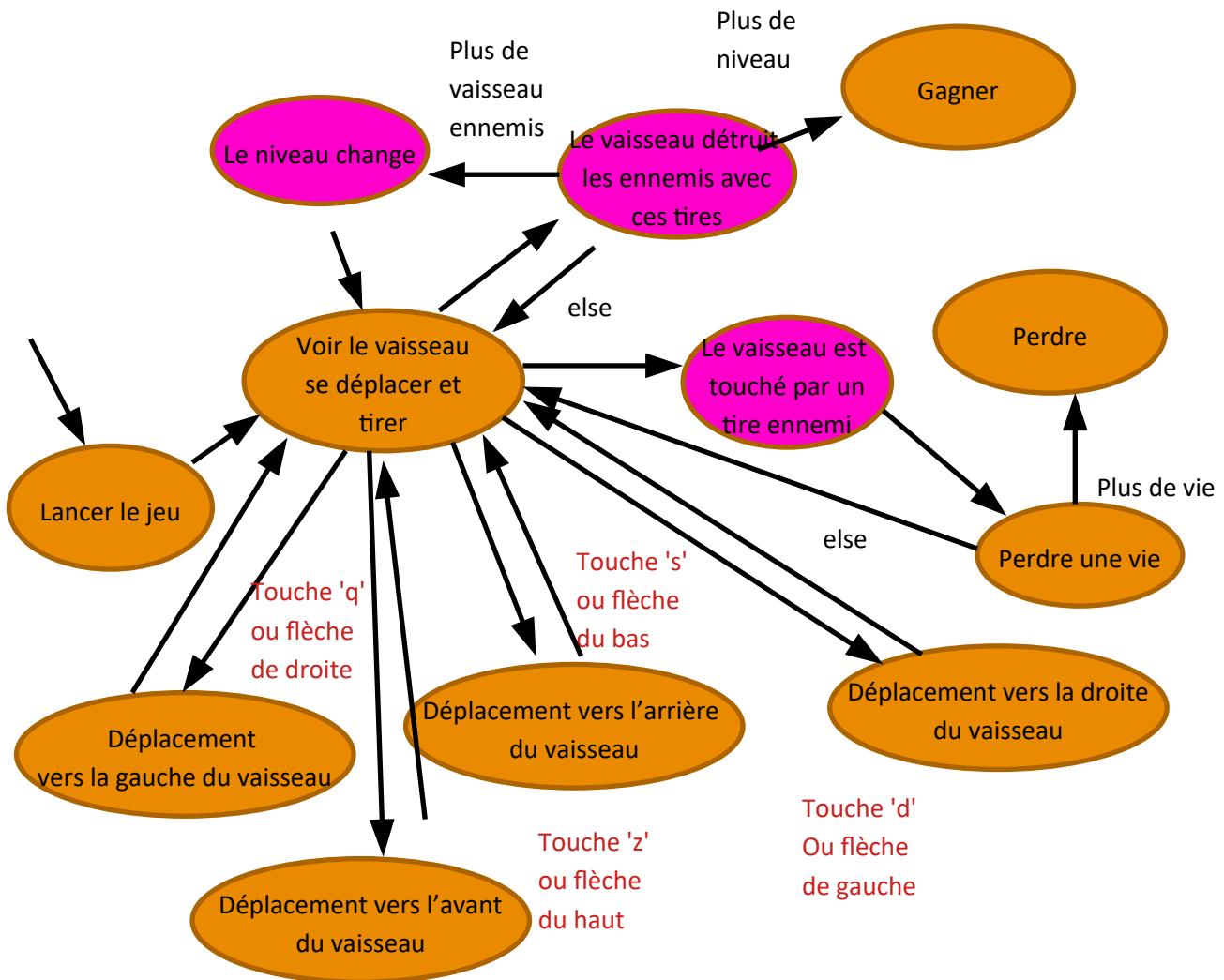
2.3 Manuel utilisateur

- **Lancer le jeu** : \$python3 main.py
- **Jouer** : z,q,s,d pour diriger le vaisseau.

2.4 Contraintes techniques

- Le logiciel doit fonctionner sur les machines de TP de l'ENIB.
- 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 IPI : barrière d'abstraction, modularité, unicode, etc...
- L'interface sera réalisée en mode texte dans un terminal de type linux en utilisant les séquence d'échappement ANSI.

2.5 Scénario d'utilisation



3 Analyse du besoin :

3.1 Fonctionnalités

- F0: Démarrer une partie
- F1 : Déplacer le vaisseau
 - F1.1 : Éliminer un ennemi avec le canon
 - F1.2 : Éviter les tirs ennemis
 - F1.3 : Récupérer des bonus pour avoir des vaisseaux en plus
 - F1.4 : Récupérer des bonus pour se protéger
 - F1.5 : Déplacer le vaisseau vers la gauche
 - F1.6 : Déplacer le vaisseau vers la droite
 - F1.7 : Déplacer le vaisseau vers l'avant
 - F1.8 : Déplacer le vaisseau vers l'arrière
- F3 : Changer de niveau
- F4 : Gagner
- F5 : Perdre
- F6 : Afficher le jeu
 - F6.1 : Afficher le vaisseau
 - F6.2 : Afficher les ennemis
 - F6.3 : Afficher les tirs
 - F6.4 : Afficher le score
 - F6.5 : Afficher le numéro de niveau
 - F6.6 : Afficher le nombre de vies

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'enchaîner rapidement les niveaux.
- La robustesse
- Le respect des méthodes de conception et de codage données en cours de IPI.

3.2.3 Importance des fonctionnalités

0 : Indispensable

1 : Forte valeur ajouté au projet

2 : Optionnelle

F0: Démarrer une partie	0
F1 : Déplacer le vaisseau	0
F1.2 : déplacer le vaisseau vers la gauche	0
F1.3 : déplacer le vaisseau vers la droite	0
F1.4: déplacer le vaisseau vers l'avant	0
F1.5 : déplacer le vaisseau vers la droite	0
F2 : Détruire un vaisseau adverse	
F3 : Changer de niveau	1
F4 : Gagner	0
F5 : Perdre	0
F6 : Afficher le jeu	0
F6.1 : afficher les vaisseau ennemis	0
F6.2 : afficher le vaisseau principal	0
F6.3 : afficher les tires	0
F6.4 : afficher le score	2
F6.5 : afficher le numéro de niveau	2
F6.6 : afficher le nombre de vies	0

4 Livrables

4.1 Echéancier

Semaine 2 : cahier des charges

Semaine 6 : conception

Semaine 10 : Projet

Semaine 13 : Code Objet

Les livrables seront envoyé par mail au format .pdf, .zip ou .tgz

4.2 Description des livrables

4.2.1 CDC : Cahier des charges

Expression et analyse du besoin.

Fichier : CB-CdC.pdf

4.2.2 Conception : Document de conception v1.0

Fichier CB-Conception.pdf

4.2.3 Projet : Code + Document de conception v2.0

Archive .zip ou .tgz contenant :

répertoire avec le code + fichier Readme.txt

répertoire avec la documentation :

CB-Conception.pdf v2.0 à jour

CB-CdC.pdf

4.2.4 Code Objet :

Archive du code .zip ou .tgz.