

# R5.8 : Qualité de développement

Projet : Space Zinzins

Cahier des  
charges

# Tables des Matières :

## I. Présentation du projet

- A. Présentation du contexte
- B. Objectifs du cahier des charges

## II. Objectifs du projet

- A. Objectif du jeu
- B. Objectif du code
- C. Objectif du projet

## III. Périmètre

- A. Le personnage principal
- B. Les ennemis
- C. Les bonus
- D. Le jeu
- E. Exclusions du périmètre
- F. Possibilités d'amélioration

## IV. Exigences fonctionnelles

- A. Liste des fonctionnalités
- B. Comportement du jeu :

## V. Exigences non fonctionnelles

- A. Contraintes Techniques
- B. Performances
- C. Ergonomie

## VI. Contraintes de réalisation

## VII. Livrables attendus

## VIII. Critères de validation

# I. Présentation du projet

## A. Présentation du contexte

Ce projet est le résultat d'une réalisation lors d'un projet d'étude dans la matière R5.8 Qualité de développement du BUT3 Informatique. Lors de cette ressource, il est demandé de produire un jeu fonctionnel avec un code propre. De plus, il faut appliquer une gestion de projet. Nous avions plusieurs choix de type de jeu à développer. Un jeu type Pacman, un jeu de dames, un Space Invaders, un Pong, un morpion et un puissance 4. Le type de jeu qui a été retenu est celui du Space Invaders. Notre équipe est composée de quatre personnes. Julian Cazin, Clément Duberseuil, Corentin Dupuy et Arthur Jacquot. Les parties prenantes du projet sont le professeur, le groupe chargé du développement du projet et les utilisateurs du jeu. L'équipe présentée juste avant se chargera du développement et de la réalisation du jeu ainsi que des tests pour s'assurer que le projet est fonctionnel, tandis que le professeur aura le rôle de client.

## B. Objectifs du cahier des charges

Ce cahier des charges vise à formaliser le besoin et de le rendre compréhensible aux différents acteurs en définissant tous les aspects du projet (fonctionnel, organisationnel, technique, juridique, administratif, etc.).

# II. Objectifs du projet

## A. Objectif du jeu

Le jeu doit être fonctionnel et les mécaniques du jeu de base Space Invaders implémentées. Il doit aussi pouvoir être jouable sans avoir un python installé sur la machine. Le joueur est un vaisseau situé au bas de l'écran. Il peut tirer pour toucher des ennemis qui descendent du haut de l'écran. Les ennemis apparaissent par vagues. Lorsqu'une vague est vaincue, une nouvelle arrive. Le joueur possède 3 vies et en gagne une après chaque vague jusqu'à 5. Lorsque son nombre de vie tombe à 0, le joueur a perdu. Au bout de la

## B. Objectif du code

Le code produit doit être propre et respecter les normes de bonne programmation telles que le nommage clair des variables, classes et fonctions et l'ajout de commentaire. En plus de cela, le projet doit être accompagné de tests concrets qui assurent le

fonctionnement des différents aspects du jeu, ces tests seront techniques au travers de plusieurs tests unitaires assurant que les différentes fonctions du code sont toutes fonctionnelles, mais aussi fonctionnelles. Le code sera publié sur GitHub, un dépôt distant utile pour le versionnage du code et vérifier la participation de chacun dans le développement..

### C. Objectif du projet

Le projet doit contenir une documentation technique claire, un cahier des charges, un plan de tests et une documentation fonctionnelle. Ces documents servent de référence pour l'ensemble des parties prenantes. Le but étant, grâce à cela, d'aboutir à un projet fonctionnel, et de bonne qualité dans les meilleurs délais.

## III. Périmètre

### A. Le personnage principal

Caractéristiques	Actions possibles
un score reflétant le nombre de point gagné par chaque ennemi tué	bouger latéralement avec les flèches directionnelles
un nombre de vie commençant à 3 et allant jusqu'à 5 maximum	tirer lorsqu'on appuie sur la barre espace
une réserve de bonus	perdre une vie lorsqu'un tir le touche
	Attraper des bonus
	Activer des bonus
	"Game Over" lorsque ses vies arrivent à zéro

### B. Les ennemis

Ils doivent pouvoir :

- bouger latéralement de manière automatique et régulière
- lorsqu'ils touchent un mur ils descendent
- certains doivent pouvoir tirer
- subir des dégâts lorsqu'ils se font toucher

- disparaître lorsqu'ils n'ont plus de vie
- Ils ont :
- un score lorsqu'ils sont éliminés

## C. Les bonus

- Ils peuvent être stocker
- Ils peuvent être

## D. Le jeu

La page du menu contient :

- Le titre
- Le bouton jouer
- Le bouton quitter
- Le bouton pour couper le son

L'affichage du jeu :

Dans la partie haute de l'écran:

- Le bouton pause
- Le score en haut à gauche
- Le nombre de vie en haut à droite

Dans la partie centrale de l'écran :

- Les ennemis sont positionnés en haut
- Le joueur est en bas

Lorsque le joueur a perdu :

- Une modale avec le score
- Le bouton rejouer
- Le bouton quitter

## E. Exclusions du périmètre

Les fonctionnalités suivantes ne seront pas inclus :

- Une histoire
- Un système de déplacement “intelligent” des ennemis
- Le stockage du score

## F. Possibilités d'amélioration

Cette partie liste l'ensemble des fonctionnalités que nous pourrions développer si nous possérons le temps nécessaire. Nous n'avons pas encore réfléchi à ces fonctionnalités de manière approfondie. :

- Des ennemis plus variés et complexes (résistance accrue, tir multiple, ...)
- Une vague contenant un "boss" (ennemi très résistant et difficile)
- Des pouvoirs pour améliorer le joueur
- Des tirs différents (une touche supplémentaire pour une attaque spéciale faisant des dégâts de zone par exemple)

# IV. Exigences fonctionnelles

## A. Liste des fonctionnalités

- Un menu d'accueil
- La possibilité de couper le son via un bouton
- Le bouton quitter qui ferme le jeu
- Le bouton jouer qui lance une partie
- La génération d'une vague d'ennemis
- Les contrôles du personnage du joueur (il peut aller de gauche à droite avec les flèches directionnelles et il peut tirer avec la barre espace)
- Les collisions entre les tirs et les personnages (ennemis et joueur)
- Lorsqu'un joueur perd toutes ses vies, le jeu affiche une modale de "Game Over"
- Lorsqu'un ennemi est éliminé, le score augmente
- Les ennemis balayent l'écran de gauche à droite. Dès qu'ils touchent un bord de l'écran, ils descendent.
- Des bonus doivent apparaître aléatoirement
- Deux types de bonus : le tir par 3 et le bouclier
- Les bonus peuvent être ramassés et utilisés quand le joueur le décide

Comportement du jeu :

Objectif : éliminer le maximum d'ennemis possible sans mourir

Fonctionnement :

- Les ennemis apparaissent par vague.
- Une vague est complétée quand tous les ennemis sont vaincus

- Lorsqu'une vague est complétée le joueur récupère une vie
- À chaque vague, les ennemis se déplacent plus vite et la cadence de tir augmente jusqu'à un certain point.

## V. Exigences non fonctionnelles

### A. Contraintes Techniques

- Le jeu est réalisé en Python 3.12.3 avec PyGame 2.6.0.
- Le projet sera partagé sur Git entre tous les développeurs.
- Il devra être compatible avec les systèmes d'exploitation Windows 10 et 11, il prendra la forme d'un fichier avec une extension .exe qu'il faudra exécuter pour pouvoir le lancer.
- Le jeu adoptera une résolution minimale de 1920\*1080p.

### B. Performances

- Le temps de chargement entre le clique sur le bouton et le lancement du niveau devra être inférieur à 5 secondes.
- Le temps d'apparition des ennemis à chaque vague ne devra pas dépasser 10 secondes.
- Le jeu ne devra pas utiliser plus de 1Go de Ram et ne devra pas prendre plus de 1Go d'espace disque.
- Le taux de rafraîchissement sera de 60 FPS.

### C. Ergonomie

- Le jeu devra présenter une police lisible et adaptée à la taille de l'écran sur le menu d'accueil.
- Le score ainsi que le nombre de vie restante devront être présents tout au long de la partie et devront être clairs et lisibles pour le joueur.
- Les ennemis, le personnage et les blocs devront être assez grands et visibles pour être vu par le joueur.

## VI. Contraintes de réalisation

Le projet est réalisé par cinq développeurs disposant chacun d'un ordinateur personnel pour réaliser le développement. Les développeurs disposent de 6 séances de 2 heures pour réaliser le projet. La date limite de rendu est définie pour le 23 novembre 2025.

Etant donné que ce projet s'inscrit dans un cadre éducatif, aucun budget concernant les infrastructures ou d'éventuelles aides payantes tierces ne sera établi, par conséquent tous les moyens utilisés dans la création de ce projet sont soit fait par les développeurs, soit issus de sources gratuites.

## VII. Livrables attendus

Les livrables attendus pour ce projet sont :

- Le présent cahier des charges contenant les objectifs, le périmètre, les exigences, les contraintes et les critères de validation du projet.
- Une documentation fonctionnelle complète contenant les fonctionnalités du jeu, des cas d'utilisations, des maquettes ainsi que les règles de gestion et les flux fonctionnels.
- Une documentation technique contenant l'architecture technique, les technologies utilisées, les informations sur la base de données, les consignes d'installation, de déploiement et de configuration ainsi que les procédures de test technique.
- Un plan de test complet et détaillé.
- Un code contenant tout le nécessaire pour que le jeu fonctionne

## VIII. Critères de validation

Le projet sera considéré comme mené à bien si :

- Les exigences fonctionnelles ainsi que toutes les fonctionnalités inclus dans la description du périmètre sont implémentées et fonctionnent selon leur description
- Les exigences techniques sont toutes respectées
- Les contraintes de délai et de ressource ont été tenus
- Les parties du jeu décrites avant sont toutes implémentées selon leur description
- Les documents cités ont été remis à la fin du projet
- Le jeu est jouable selon la description qui en a été faite
- L'ensemble des tests définis dans le plan de test doivent être réussi à hauteur de 90% minimum

