

Documentation BuckShot Roulette

Cette documentation présente les aspects fonctionnels et techniques de l'application de BuckShot Roulette 2D développée en Python.

SOMMAIRE :

1. Analyse Fonctionnelle :	2
- 1.1. Presentation :	2
- 1.2. Mécaniques du jeu :	2
2. Analyse Technique :	2
- 2.1. Architecture Logicielle :	2
- 2.2. Spécifications Techniques :	3
- 2.3. Installation et Exécution :	3

1. Documentation Fonctionnelle

Présentation

Buckshot Roulette 2D est une adaptation numérique inspirée du jeu "Buckshot Roulette". Le joueur affronte un croupier (le Dealer) dans une version mortelle de la roulette russe utilisant un fusil à pompe.

Mécaniques de Jeu

- **Initialisation** : Le joueur saisit son pseudonyme sur un contrat avant de commencer la partie.
- **Le Fusil (Arme)** : Le fusil est chargé avec un mélange de munitions réelles (rouges) et à blanc (blanches). Le nombre de chaque type est révélé en début de manche avant d'être mélangé.
- **Prise de décision** : À chaque tour, le joueur peut choisir de se tirer dessus ou de tirer sur le Dealer.
 - S'il se tire dessus avec une munition à blanc, il rejoue immédiatement.
 - S'il reçoit une munition réelle, il perd un point de vie et le tour passe au Dealer.
- **Objets stratégiques** : Le joueur peut utiliser des cigarettes pour regagner des points de vie.
- **Interface et Ambiance** : Le jeu propose un menu complet avec gestion du son, des règles et une esthétique visuelle sombre.

2. Documentation Technique

Architecture Logicielle

Le projet suit une structure modulaire pour séparer les responsabilités :

- **main.py** : Point d'entrée initialisant l'affichage et lançant le menu principal.
- **game.py** : Moteur principal gérant la logique des tours, les phases de jeu et les interactions entre les entités.
- **joueur.py** : Définit la classe Joueur gérant les points de vie et l'état (joueur ou dealer).
- **arme.py** : Gère le chargement aléatoire des munitions et le mécanisme de tir.
- **animations.py** : Gère les effets visuels tels que la fumée après un tir.
- **map.py** : Gère l'affichage de l'environnement (table, arrière-plan).
- **sons.py** : Centralise le chargement et la lecture des effets sonores et des musiques.

Spécifications Techniques

- **Langage et Bibliothèques** : Python 3.12 utilisant **Pygame** pour le rendu 2D et la gestion des événements.
- **Gestion des ressources** :
 - **Images** : Utilisation de fichiers PNG pour les sprites (fusil, munitions, cigarettes) et les éléments d'interface.
 - **Audio** : Support des formats .ogg pour les effets (tir, recharge) et .mp3 pour l'ambiance sonore.
- **Système de Coordonnées** : Le jeu utilise un affichage en 1080p (1920x1080) avec des éléments positionnés dynamiquement ou par constantes selon l'écran.

Installation et Exécution

1. Activer l'environnement virtuel via les scripts fournis dans venv/bin/.
2. Lancer la commande python main.py.