

# Documentation technique

BUT3 DevALT (R5-8)

Equipe :

- Guerby Naharro
- Nicolas Grandjean
- Nathan Bonnard
- Arthur Aurignac

## Introduction

Cette documentation décrit la structure interne, les choix techniques, les mécanismes de fonctionnement et les éléments nécessaires au développement, à la maintenance et à l'évolution du jeu **Débarquement 1945**, un shoot'em up réalisé avec le moteur de jeu [Godot](#) (le code écrit GDScript).

Elle s'adresse aux développeurs, intégrateurs et mainteneurs du projet.

## Architecture technique

### Structure générale du projet

Organisation recommandée des répertoires Godot :

```
/assets/  
  /sprites/  
  /backgrounds/  
  ...  
/Components/ -> nodes/scenes that can be reused as it is on other scenes  
/Levels/  
  level_1.tscn  
/Scenes/ -> nodes/scenes that can't be reused except maybe on the level  
itself  
  /Player/  
    player.gd  
    player.tscn  
  /Items/  
/Singletons/ -> game's singletons
```

```
game_manager.gd  
/ui/ -> menus and such
```

## Schéma d'architecture logique

### **Game Manager** (nœud central)

-> Gère le spawn des ennemis

### **Player**

-> Reçoit les entrées de la souris

-> Tire automatiquement

-> Collisionne avec ennemis

### **Enemy / Boss**

-> IA simple de déplacement (descente verticale ou pattern spécial)

-> Tir de projectiles

-> Système de points de vie

### **Projectile**

-> Mouvement linéaire

-> Destruction hors écran

### **Bonus**

-> Apparition aléatoire

-> Effet temporaire ou instantané (buffs/heal)

### **HUD**

-> Affichage score, PV, bonus actifs

-> Affichage et changement des menus

## Technologies utilisées

### **Moteur et langage**

- **Godot Engine** v4.5
- **GDScript** (langage principal)
  - et donc suivi de ses [règles/guidelines](#)

### **Outils externes**

- **GitHub** (versionnement, pull requests, issues, Kanban)
- **Éditeur d'assets** (Aseprite ou équivalent)

## Spécifications techniques des fonctionnalités

## Déplacement du joueur

- Suivi direct de la position de la souris.
- Limitation dans les bordures de l'écran.
- Vitesse réglable dans l'éditeur sur la ressource des stats

## Système de tir

### Joueur

- Cadence fixe
  - modifiable via buff ingame
  - modifiable dans la partie shooting\_component
- Direction verticale vers le haut.
- Instanciation d'un projectile via un Timer.

### Ennemis

- Tirs périodiques ou aléatoires.
- Direction verticale, d'en haut vers le bas.

## Ennemis

- Apparition via un spawner intégré au Game Manager et avec un timer.
  - gère aussi le nombre maximum d'ennemis qui peuvent apparaître en même temps.
- Points de vie individuel pour chaque ennemis.
- Vitesse et fréquence d'apparition ajustables dans les parties stats des ennemis .
- Destruction lorsque  $PV \leq 0$  ou lorsque qu'ils sortent du cadre.

## Collisions

Gestion centralisée via les zones *Area2D* Godot.

Règles :

- Projectile joueur → ennemi
  - le projectile en lui même détecte la collision et dans son code, contient les informations du joueur et de l'ennemis atteint
- Ennemi → joueur
  - l'ennemis détecte la collision avec le joueur et utilise le HealthComponent du joueur pour lui infliger des dégâts
- Bonus → joueur
  - Le bonus détecte une collision avec le joueur et commence à modifier les stats du joueur via ses méthodes/propriétés

- Mur (walls) -> ennemis
  - Destruction automatique des objets sortant de l'écran.

## Scoring

- +1 par ennemi éliminé
- Stockage dans le GameManager

## Bonus

Types possibles :

- Amélioration temporaire du tir (l'aire de tir augmente)
- Soins
- Augmentation de dégâts

Probabilité modifiable pour les buffs et un seul buff par ennemis.

## Écrans et menus

### Menu principal

Le premier menu, propose de jouer ou de quitter.

### HUD

- Score
- PV
- Bonus actifs

### Game Over

- Saisie pseudonyme
- Enregistrement du score
- Retour menu
- Option de Rejouer

## Gestion des erreurs

Erreurs internes identifiées :

Code	Description	Action
E001	Ressource manquante (sprite/audio)	Charger une ressource par défaut

Code	Description	Action
E002	Impossible d'instancier un ennemi	Log + ignorer l'instance
E003	Fichier de score illisible	Recréation du fichier
E004	Valeur hors limites (PV, score)	Clamp automatique

Des logs sont stockés dans `user://logs/`.

## Installation, déploiement et configuration

### Installation

1. Installer Godot 4.5
2. Cloner le dépôt GitHub
3. Ouvrir le projet depuis Godot
4. Vérifier les dépendances (addons éventuels)
5. Lancer en debug depuis l'éditeur

### Déploiement

- Export Godot vers Windows / Linux / Web
- Vérifier la configuration d'export dans `export_presets.cfg`

## Procédures de test technique

### Tests unitaires

Utilisation d' `assert` dans godot.

Une fonction en gdscript qui permet de s'assurer qu'une condition à l'intérieure retourne vraie ou sinon retourne une exception `AssertionFailed`.

### Tests de performance

- Vérification du nombre maximum d'entités à l'écran
- Pas de perte >10 fps pendant actions intenses

## Maintenance et évolutivité

### Bonnes pratiques

- Nommage clair des nœuds Godot
- Séparation stricte des scripts par rôle
- Utilisation des signaux pour limiter les dépendances

## **Extensions futures possibles**

- Nouveaux types d'ennemis
- Multiples niveaux
- Modes de difficulté
- Système de power-ups avancés