MISSION OCÉAN

Jeu d'arcade 2D réalisé avec **Godot 4** en **GDScript**. Projet de Terminale STI2D, spécialité SIN.

Jouer en ligne : https://mou.alwaysdata.net/jeu-dd/

Auteurs

- Rayan MOUAK
- Classe : Terminale STI2D SIN Année scolaire : 2026-2027

Projet individuel.

Thème de développement durable

Le jeu traite de **la pollution des océans par les déchets plastiques**, en lien avec l'**Objectif de développement durable n°14 de l'ONU : vie aquatique**.

Plus le joueur laisse passer de déchets, plus l'océan se dégrade : l'eau devient trouble, les coraux blanchissent et les poissons disparaissent. À la fin de chaque partie, une information réelle sur la pollution marine est affichée (« Le savais-tu ? »).

Principe du jeu

Le joueur pilote un **drone sous-marin de dépollution**. Les objets arrivent par la droite de l'écran et dérivent vers la gauche.

Objet	Effet
Déchets (bouteille, canette, sac plastique, gobelet)	+10 points (× le combo), la pollution baisse un peu
Bonus cœur	+1 vie (5 vies au maximum)
Bonus aimant	attire les déchets proches pendant 6 secondes
Baril toxique, nappe de pétrole	−1 vie
Déchet qui s'échappe par la gauche	la pollution monte de 7 % et le combo est perdu

- **Combo** : ramasser plusieurs déchets à la suite sans erreur multiplie les points (×2 à ×5).
- **Jauge de pollution** : elle commence à 40 %.
- **Record** : le meilleur score est enregistré.
- **Difficulté progressive** : le jeu est découpé en **5 zones** (Lagon, Récif corallien, Grand large, Abysses, Continent de plastique). On change de zone tous les 12 déchets. À chaque zone, les objets arrivent plus vite et plus souvent, il y a plus de dangers et l'eau s'assombrit.

Moteur choisi et justification

J'ai choisi **Godot 4** :

- il est **gratuit** et libre, aucun logiciel payant n'est nécessaire ;

- il est fait pour la **2D** : nœuds CharacterBody2D, Area2D, Timer, particules… ;
- le langage **GDScript** ressemble à Python ;
- il permet d'**exporter le jeu pour le Web** (rendu Compatibility, sans Thread Support).

Organisation du projet :

```
MISSION_OCEAN/
├── project.godot         configuration (scène de départ : main.tscn, rendu
Compatibility)
├── export_presets.cfg    export Web (Thread Support désactivé)
├── conception.txt        fiche de conception du jeu
├── scenes/
│   ├── main.tscn         scène principale (monde, joueur, affichage)
│   ├── joueur.tscn       le drone (CharacterBody2D)
│   └── objet.tscn        un objet qui dérive (Area2D)
├── scripts/
│   ├── main.gd           les règles du jeu (score, vies, pollution, zones, victoire,
défaite)
│   ├── joueur.gd         le déplacement du drone
│   ├── objet.gd          le mouvement et la collision des objets
│   ├── decor.gd          le fond marin dessiné par le code
│   ├── hud.gd            l'affichage (score, vies, jauges, écrans)
│   └── sons.gd           les sons fabriqués par le calcul
└── assets/               les images (SVG) dessinées pour le jeu
```

Les scripts communiquent par des **signaux** : un objet envoie `attrape` quand le drone le touche, et `sorti` quand il quitte l'écran. Le script `main.gd` reçoit ces signaux et applique les règles. Un **Timer** (`MinuteurObjets`) fait apparaître un nouvel objet à intervalle régulier.

Commandes

- **Flèches** ou **Z Q S D** : déplacer le drone
- **Espace** ou **Entrée** : commencer / rejouer
- **P** ou **Échap** : pause
- **M** : couper le son

Victoire / Défaite

- **Victoire** : ramasser **60 déchets**. L'océan est sauvé, et chaque vie restante rapporte 100 points de bonus.
- **Défaite** : le drone n'a plus de vie, **ou** la pollution atteint **100 %** (l'océan est mort).
- L'écran de fin affiche le score, le record, le nombre de déchets ramassés, le meilleur combo et une information sur la pollution marine. Espace permet de rejouer.

Répartition du travail

Projet individuel : je suis seul sur ce projet, il n'y a donc pas de répartition entre plusieurs élèves.

Bug ou difficulté principale

Sans protection, un drone qui touche un baril toxique risque de **perdre plusieurs vies d'un coup** : il reste au contact du danger pendant plusieurs images, ou il en touche un deuxième

juste après, sans avoir le temps de réagir.

Solution trouvée

Après un choc, le drone devient **invincible pendant 1,5 seconde** (variable `temps_invincible` dans `joueur.gd`, réglée par `DUREE_INVINCIBLE` dans `main.gd`). Pendant ce temps, il clignote et les dangers ne lui font rien. Le joueur voit qu'il a été touché et a le temps de s'éloigner.

Amélioration après le test croisé

(À compléter après avoir fait jouer quelqu'un au jeu.)

- Testeur : .....
- Ce qu'il a remarqué : .....
- Ce que j'ai changé (ou ce que je changerais) : .....

Sources des médias et ressources

- **Images** : dessinées pour le jeu au format SVG (dossier `assets/`).
- **Décor** : dessiné directement par le code (`decor.gd`, fonction `_draw()`).
- **Sons** : fabriqués par le calcul dans `sons.gd` (ondes sinusoïdales), sans fichier audio.
- **Police** : police par défaut de Godot.
- **Moteur** : Godot Engine 4, gratuit, licence MIT — https://godotengine.org