

Game Design Document (GDD)

Titre du jeu :

Squid Game 2D - Red Light, Green Light

Objectif du jeu :

Le joueur (ou les joueurs contrôlés par IA) doit atteindre la ligne d'arrivée sans se faire éliminer lorsqu'il bouge pendant que la poupée regarde.

Concept :

Basé sur le célèbre jeu "Un, deux, trois, soleil", les joueurs avancent seulement lorsque la poupée ne regarde pas. Si un joueur est en mouvement pendant que la poupée regarde, il est éliminé.

Personnages :

Joueur Principal :

Le joueur principal est contrôlé par l'utilisateur et se distingue des autres joueurs IA par les éléments suivants :

Contrôles :

- Déplacement horizontal : touches fléchées ou A/D.
- Pause : touche P.
- Redémarrage de la partie : touche R.

Fonctionnalités :

- Le joueur peut bouger librement lorsque la poupée ne regarde pas (`isTurned == false`).
- Si le joueur bouge pendant que la poupée regarde (`isTurned == true`), il est éliminé (`Die()`).
- Le joueur a un compte à rebours de 30 secondes. Si le temps expire avant de gagner, il perd.
- L'interface utilisateur (UI) affiche le temps restant. Il devient rouge si < 5 secondes.
- Si le joueur entre dans une zone de victoire (`WinZone`), il gagne la partie (`GameWon()`).

Scripts utilisés :

✓ PlayerController.cs

- Gère le déplacement du joueur avec Rigidbody2D.
- Vérifie l'état de la poupée.
- Gère le timer, l'élimination, la victoire, la pause et le redémarrage.

Joueurs secondaires (IA) :

- Déplacement aléatoire vers l'avant.
- Risque d'élimination s'ils bougent quand la poupée regarde.
- État : Vivant / Éliminé

Poupée (PoupeeTurn) :

- Se retourne à intervalles aléatoires.
- Quand tournée, elle "surveille" les joueurs.
- Quand dos tourné, les joueurs peuvent avancer librement.

Mécaniques principales :

Fonction	Description
Déplacement Joueur Principal	Contrôlé par le clavier (gauche/droite). S'il bouge pendant que la poupée regarde, il est éliminé.
Déplacement IA (Joueurs secondaires)	Mouvement aléatoire à intervalles réguliers. Risquent l'élimination si mouvement pendant que la poupée regarde.
Poupée (PoupeeTurn)	Alterne entre "regarde" et "ne regarde pas". Détecte les mouvements durant son regard.
Élimination	Joueur principal : instantanée s'il bouge quand la poupée regarde. IA : 50% de chance d'élimination si en mouvement quand la poupée regarde.
Timer	30 secondes pour atteindre l'arrivée. Affiché à l'écran, devient rouge sous les 5 sec restantes.
Victoire	Atteindre la zone de fin (winZone) sans être éliminé.

Défaite	Temps écoulé ou mouvement détecté pendant que la poupée regarde.
Pause	Appui sur P : arrête le temps et le mouvement.
Restart	Appui sur R : redémarre la scène actuelle.

Commandes :

Touche	Action
E	Élimine manuellement un joueur non éliminé
Espace	Arrête temporairement tous les joueurs

Composants principaux (Scripts) :

- ☒ JoueurSecondaire.cs
 - Gère le déplacement aléatoire.
 - Vérifie l'état de la poupée.
 - Gère l'élimination avec animation.
 - Méthodes : Eliminer(), StopTemporarily(), etc.
- ☒ PoupeeTurn.cs
 - Change régulièrement l'état isTurned.
 - Utilisé par les joueurs pour savoir s'ils peuvent bouger ou non.
- ☒ GameControllerCustom.cs
 - Gère tous les joueurs.
 - Permet de les arrêter ou les éliminer.
 - Peut afficher les logs de fin de partie.

UI & Affichage :

- Une scène 2D avec :
 - Les joueurs en ligne de départ.
 - La poupée en haut (centre).
 - Une ligne d'arrivée.
- Feedback visuel :
 - Poupée tournée ou pas.
 - Joueurs tombent quand éliminés.

Paramètres techniques :

Élément	Valeur
Nombre max de morts	4
Vitesse de déplacement joueur	2f

Intervalle changement poupée	Aléatoire entre 1s et 3s
Élimination	50% de chance si mouvement quand poupée regarde

États de la poupée :

- isTurned = true → La poupée regarde → Danger.
- isTurned = false → Les joueurs peuvent avancer.

Fin de partie :

- Si nbTotalMorts >= maxMorts → stop des mouvements, message de fin.

Extensions futures :

- Mode multijoueur local.
- Système de score ou chronomètre.
- Ajout d'obstacles ou camouflage.
- Voix off ou musique.

DIAGRAMME DE CLASS :

