

PROJET DRAGON BALL ESQUIVE

Lewis CRANE

Table des matières

<i>Présentation Générale</i>	2
Besoin Initial	2
Solution Proposée	2
<i>Plan et Maquette du Jeu</i>	2
<i>Dictionnaire de Données et MCD</i>	3
Dictionnaire de Données	3
Modèle Conceptuel de Données (MCD)	3
<i>Descriptif Technique de l'Application</i>	3
Technologies et Langages Utilisés	3
Organisation du Code	3
<i>Procédure d'Installation</i>	4
<i>Descriptif des Tâches et Attributions</i>	4
<i>Déroulement du Projet et Difficultés Rencontrées</i>	4
Problèmes et Solutions	4
<i>Conclusion</i>	5

Projet Jeu Dragon Ball Esquive

Présentation Générale

Besoin Initial

L'objectif du projet est de développer un jeu en Python avec Pygame, inspiré de l'univers de Dragon Ball. Ce jeu met en scène un personnage principal (Goku) qui doit éviter des obstacles (balles) tout en collectant des bonus pour survivre le plus longtemps possible.

Solution Proposée

Pour répondre à ce besoin, nous avons développé une application en Pygame, avec une interface graphique simple et un gameplay fluide basé sur des déplacements, collisions et gestion des événements.

Plan et Maquette du Jeu

Le jeu se compose des éléments suivants :

- Un écran principal où se déroulent les actions.
- Un personnage contrôlé par le joueur (Goku) qui peut se déplacer dans toutes les directions.
- Des balles en mouvement qui servent d'obstacles.
- Des bonus (vie, vitesse, spécial) à collecter.
- Une interface affichant les points de vie et un message d'instructions.

Schéma de l'interface :



Dictionnaire de Données et MCD

Dictionnaire de Données

Nom	Type	Description
perso	Objet `Perso`	Représente le personnage contrôlé par le joueur
balle	Objet `Balle`	Représente un obstacle en mouvement
bonus	Objet `Bonus`	Représente un élément à collecter
vie	Entier	Nombre de points de vie du joueur
dx, dy	Entiers	Direction et vitesse des balles
fenetre	Objet `pygame.Surface`	Fenêtre principale du jeu

Modèle Conceptuel de Données (MCD)

Il n'y a pas de MCD pour ce projet.

Descriptif Technique de l'Application

Technologies et Langages Utilisés

- Python pour la logique du jeu
- Pygame pour la gestion graphique et des événements
- Bibliothèque random pour les déplacements aléatoires des objets
- Gestion des collisions via `math.sqrt()` pour détecter les contacts

Organisation du Code

- Classe mère pour les objets graphiques
- Classe pour le personnage jouable
- Classe pour les obstacles
- Classe pour les objets à collecter
- Boucle principale : Gère l'affichage et les interactions

Procédure d'Installation

1. Installer Python (version 3.10 ou ultérieure)
2. Installer Pygame avec la commande :
```bash pip install pygame```
3. Télécharger le projet depuis GitHub ou un autre dépôt
4. Lancer le jeu avec la commande :  
```bash python main.py```

6. Guide d'Utilisation

- Déplacements :
 - Flèche gauche : Aller à gauche
 - Flèche droite : Aller à droite
 - Flèche haut : Monter
 - Flèche bas : Descendre
- Quitter le jeu :
 - Appuyer sur `Échap`

Objectif : Éviter les balles et collecter les bonus pour survivre le plus longtemps possible.

Descriptif des Tâches et Attributions

Tâche	Réalisée par
Développement du moteur du jeu	CRANE Lewis
Création des assets graphiques	CRANE Lewis
Intégration du son et musiques	CRANE Lewis
Tests et débogage	CRANE Lewis
Rédaction de la documentation	CRANE Lewis

Déroulement du Projet et Difficultés Rencontrées

Problèmes et Solutions

- Gestion des collisions imprécise → Solution : Calcul de distance via ``math.sqrt()``.
- Bonus supprimés pendant l'itération → Solution : Utilisation de ``liste[:]`` pour éviter les erreurs.
- Personnage invincible après une collision → Solution : Ajout d'un compteur pour limiter l'invulnérabilité.

Conclusion

Ce projet nous a permis de renforcer nos compétences en développement Python et Pygame, en gestion de projet, et en résolution de problèmes techniques. Nous avons appris à structurer un projet en plusieurs modules et à optimiser l'expérience utilisateur.