# Rapport Technique - RunnerGame

#### Contexte

RunnerGame est un jeu de type Runner 2D infini développé en .NET avec SFML. L'objectif du joueur est d'éviter des obstacles en sautant tout en accumulant un score basé sur le temps de survie. Le jeu propose un menu principal, un système de score et une gestion des collisions avec les obstacles.

## **Architecture du Projet**

Organisation du projet :

```
RunnerGame/
                            # Contient les fichiers de ressources
     assets/
        - 🎨 background.jpg
                                # Image de fond du jeu
        - 🎨 ground.png
                               # Texture du sol
       – 🎨 obstacle.jpg
                              # Texture des obstacles
       – 🎨 player.png
                              # Sprite du joueur
        – 🎨 font.ttf
                           # Police d'écriture utilisée pour l'affichage du texte
                          # Fichiers compilés
     - 📂 bin/
                          # Fichiers intermédiaires de build
    – 📂 obi/
     RunnerGame.exe
                                  # Exécutable unique du jeu
      📜 Game.cs # Classe principale du jeu, gère le rendu, la logique du menu et du jeu
      Obstacle.cs
                               # Gestion des obstacles et de leur mouvement
        Player.cs
                              # Gestion du joueur : mouvements et collisions
      Score.cs
                              # Système de gestion du score, lecture/écriture des scores
        scores.txt
                              # Fichier contenant les scores des parties précédentes
      RunnerGame.csproj
                                    # Fichier de configuration du projet .NET
        RunnerGame.sln
                                   # Solution Visual Studio du projet
        GUIDE.md
                       # Documentation du projet, instructions d'installation et d'utilisation
```

#### Description des classes principales

Classe	Description
game	Gère la boucle principale du jeu, les événements utilisateur et le rendu graphique.
player	Gère le joueur : déplacement, collision, gestion des limites de l'écran.
obstacle	Définit les obstacles et leur mouvement sur l'écran.
score	Gestion des scores et sauvegarde dans score.txt

## **Prérequis**

- Windows (64 bits recommandé) ou Linux/macOS
- .NET SDK 9.0+
- SFML.NET (CSFML) installé et accessible
- **Un terminal** (cmd, PowerShell, bash)
- **Git** (optionnel, pour récupérer le dépôt)

### Tests Unitaires effectués avec NUnit et NUnit3TestAdapter

- Test de la gestion des collisions avec les obstacles (tests/CollisionTests)
- Test de la gestion du score (tests/ScoreTests)
- Test de la gestion des assets (Textures et police)(tests/AssetsTests)
- Test pour empêcher la sortie d'écran du joueur (tests/PlayerTests)

Pour effectuer les tests dans le bash :
cd RunnerGame.tests
dotnet clean
dotnet restore
dotnet build
dotnet test
Jouer au jeu
Lancer RunnerGame.exe
Lancer RunnerGame.exe ou
ou
ou Aller dans le <b>bash</b>
ou Aller dans le <b>bash</b> cd RunnerGame
ou Aller dans le bash cd RunnerGame dotnet clean
ou Aller dans le bash cd RunnerGame dotnet clean dotnet restore