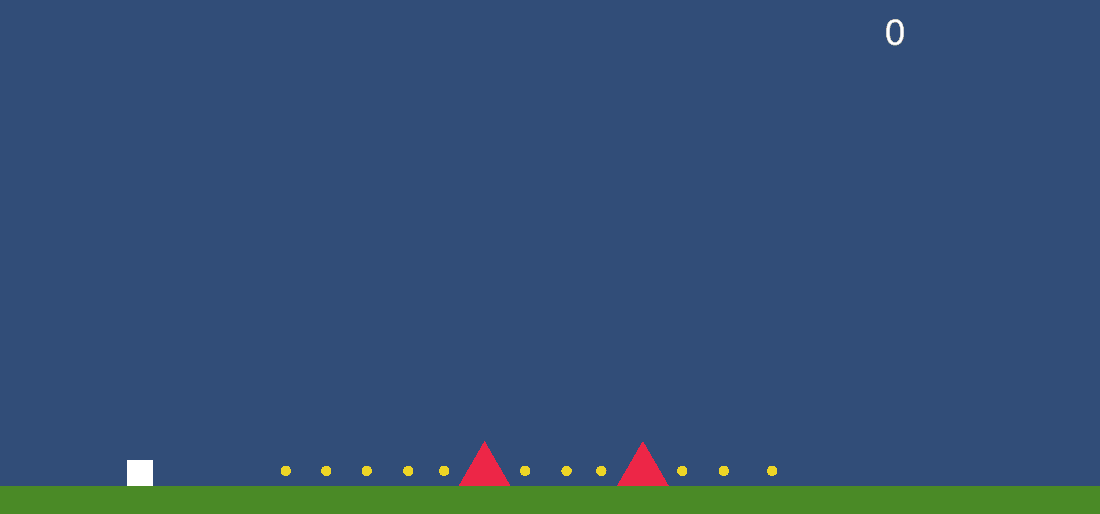
Eval Unity



Voici, ci-dessus, ce que vous devez reproduire.

Pour ce projet, pas besoin de sprites. Utiliser des formes primitives (carré cercle et triangle).

Dans ce projet il y a :

Un sol (le joueur ne doit pas passer au travers).  
Un carré, représentant le joueur, avec lequel on peut se déplacer latéralement et sauter.  
Pour un déplacement plus simple, il faut réduire/retirer la friction sur le carré. (Explications plus bas)  
Des pièces (cercle jaunes)  
Un son pour la collecte des pièces (donné plus bas)  
Des obstacles (triangle rouge)  
Un compteur de points  
Une UI pour le Gameover avec un texte, l’affichage du score, un bouton permettant de relancer une partie (recharger la scène), un bouton permettant de quitter le jeu, ainsi qu’un fond semi-transparent.  
Un game manager avec :  
 la gestion des points (avec l’affichage)  
 le rechargement de la scène (retry)  
 la gestion du gameover  
 la fonction permettant de quitter le jeu.

Fonctionnalités :

Déplacement horizontal et saut du joueur  
Collecte de pièces. 1 pièce = 1 point  
Lors de la collecte d’une pièce un sont doit être joué.  
Si le joueur touche un triangle -> gameover  
Le gameover arrête le temp, et affiche l’ui du gameover et cache l’ui de la partie

Pour retirer la friction il faut ajouter un « Physics Material 2D » (fenêtre « Projet »)Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Editez le material et réduisez la friction. (Ici je l’ai appelé NoFriction)

Il faur ensuite appliquer le material au rigidbody2D du gameobject du joueur  
Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Ressources :

Son pour la collecte des pièces :

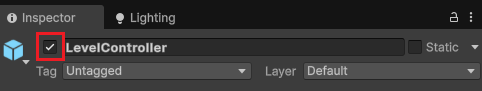
<https://drive.google.com/file/d/1_NKccKZhF1SrXAq5R_JAvjnYuksITysE/view?usp=drive_link>

Pour stopper le temps :

Time. timeScale = 0

(Pensez à le remettre à 1 quand scène est lancée, soit au cilc du bouton « rejouer » soit dans l’Awake d’un script gameManager par exemple)

Pour cacher par défaut l’UI vous pouvez juste décocher la case à coté du nom du GameObject dans l’inspecteur



Pour l’afficher ou le cacher en code utilisez :

<objet>.SetActive(false ou true)

(Remplacez « <object> » par l’objet que vous voulez activer/désactiver.)  
(ne le faite pas dans un script sur l’objet ou enfant de l’objet, sinon un fois l’objet désactivé vous ne pourrez plus le réactiver)

Conseil : faites un UIManager pour gérer l’ui.

Pour quitter le jeu :

Application.Quit();

(cela ne fonctionne seulement quand on « build » le jeu, mais mettez-le quand même)

Pour l’effet « semi opaque » de l’ui GameOver, vous pouvez ajouter un panel (UI->Panel) et dans son component image changer l’opacité de la couleur.