Documentation

V3.0

NOTE FROM THE AUTHOR

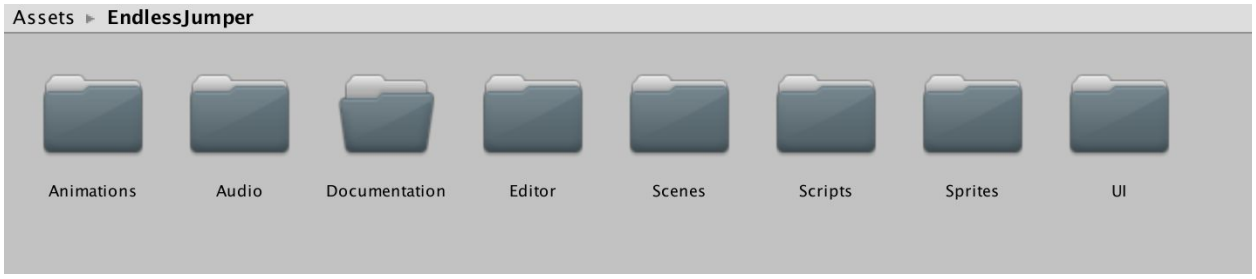
Thank you for purchasing the asset. If you require any help or face any issues, feel free to contact me for support.

This is a major update (after 3 years) of a popular asset. The update brings a lot of new features and gameplay changes which make this the most easy to use package for an endless jump game.

I will try to release updates more frequently from now onwards.

GETTING STARTED

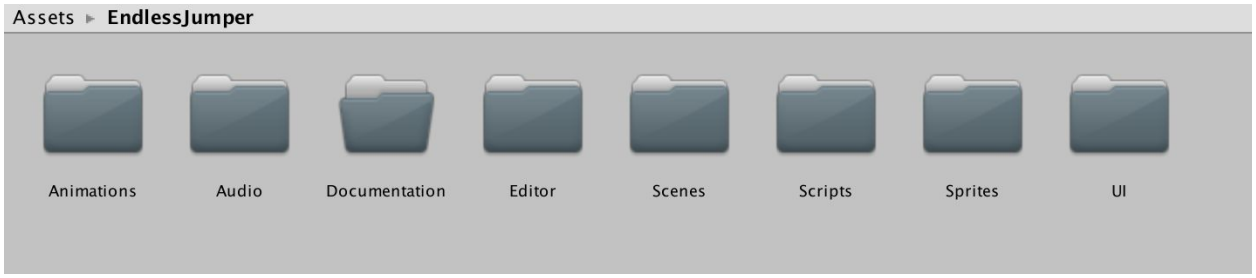
When you import the package, a folder named EndlessJumper will be added to your project.



To get started straight away, open and run the game template EndlessJumper scene in the scenes folder.

PROJECT FOLDERS

The project folders have names that are self explanatory.



Animations: The package uses Unity animations (except for the shake on game over) and all animation controllers and animation files are in this folder

Audio: All sound files are in this folder.

Documentation: The documentation pdf is in this folder.

Editor: A small editor utility to delete playerprefs or to add coins is in here.

Scenes: The package uses one scene only named EndlessJumper - and it can be found here.

Scripts: The different scripts used for gameplay are found here.

Scripts/UIPanels: The package uses the an approach in which each UI Panel has it's own separate MonoBehavior attached. All buttons of a panel call a function from the parent panel. All these classes are under this folder.

Sprites: The different sprites used in gameplay.

UI: The sprites used for the Canvas.

SCRIPTS

GameManager.cs: The game manager controls all the gameplay related functions. It does pooling of object at the start and initializes the different tiles, enemies, powerups etc.

Enemy.cs: Each enemy has this. It controls the movement of the enemy depending on it's type.

GameSettings.cs: A serializable list of parameters (items appear probability, speed, jump distance etc.) that are controlled via GameManager Object in the hierarchy.

GUIManager.cs: A singleton class that controls all the navigation of the different panels via a stack.

JumpPowerup.cs: Each jump collectible has this. It enables the player to have the jetpack ability for a certain period of time when the powerup is collected.

Player.cs: The player object has this. It controls the movement of the player - it's ability to appear from the other side etc.

PlayerCamera.cs: The camera has this. To follow the player, the camera smoothly lerps in this monobehavior.

Scale.cs: This scales the sprite renderers based on the size of the screen. E.g. the platform has this so that it does not go out of bounds if you are say on an iPhoneX.

SkinsManager.cs: This is a container for the different skins used in the game. You can add more skins from the SkinsManager object in the project hierarchy and the parameters are self explanatory.

CoinPowerup.cs: Similar to JumpPowerup.cs - this is used by a coin.

Tile.cs: Each tile has it and it is used to move the tile depending on the type and jump the player on trigger enter.

SoundManager.cs: A singleton class to play sound effects when needed.

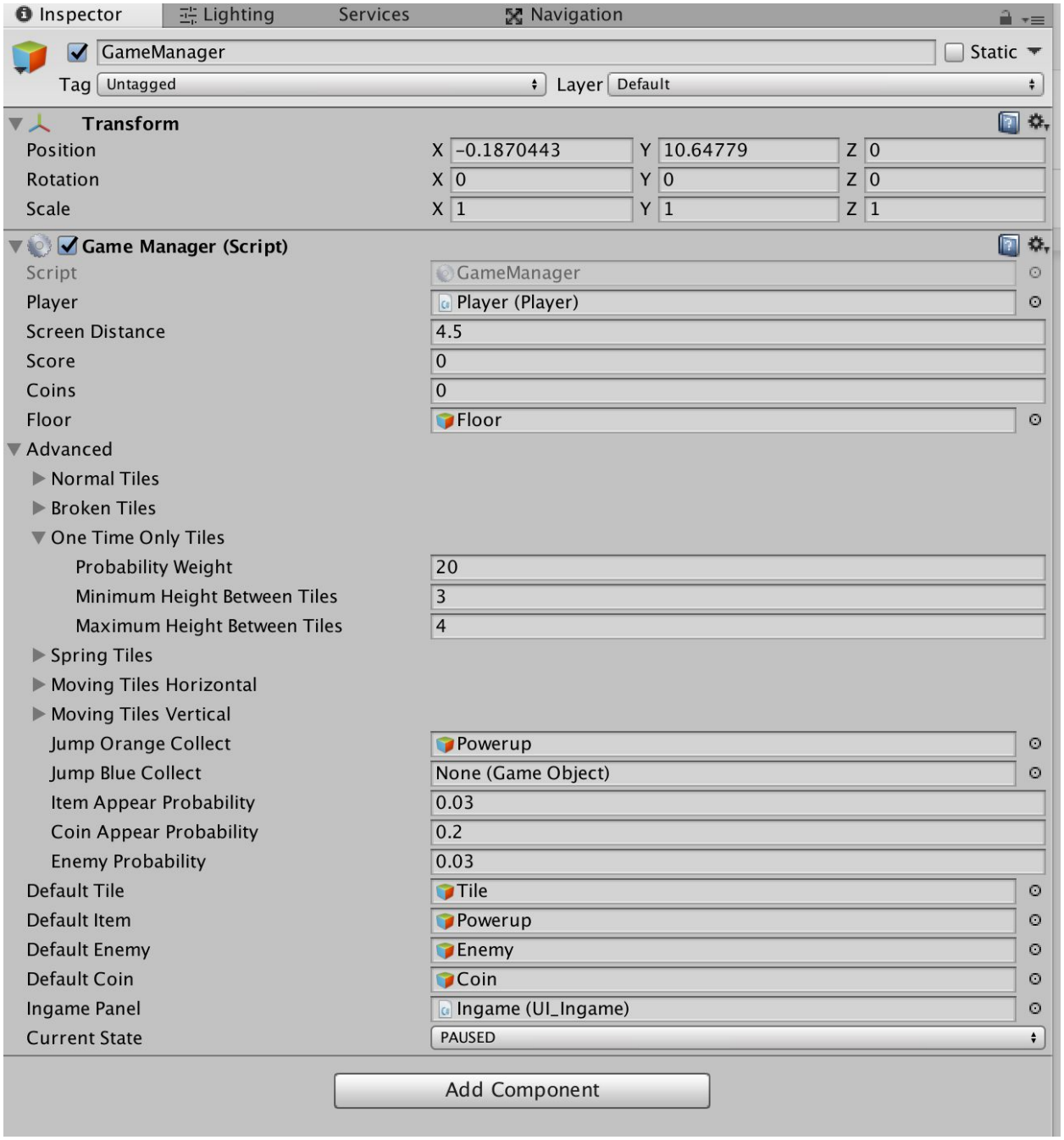
MusicManager.cs: A singleton class to play music when needed.

UI_Ingame.cs, UI_GameOver.cs, UI_MainPanel.cs, UI_Pause.cs, UI_Skins.cs: Panel classes that show and control everything in that particular canvas panel.

GAME SETTINGS

The Settings are the a very important part of the package to tune the game to your liking.

To Edit the **Game Settings, click on the GameManager GameObject from the hierarchy window:**



In the inspector window, you will see numerous settings under the **Advanced group:**

Screen Distance: *Float* This variable stores the distance of the screen and objects are created from –distance to +distance. So if you want to spread tiles, make the distance as large as possible (that fits the screen).

Score: *Integer* The score of the player in the game session. Cleared when restarted.

Floor: *GameObject* This GameObject triggers when the user's character has fallen. It is also used in determining whether a tile has gone below the camera view so that it can be destroyed.

Advanced: All Tiles Common:

The first variable in this section for every tile is is the GameObject of that particular tile (broken, one time, normal etc).

ProbabilityWeight: *Float:* The higher the weight, the more chances are there of that tile appearing.

MinimumHeight/MaximumHeight: *Float*: The minimum/maximum height difference between this tile and the next generated tile. The higher the range, the further apart the tiles will be generated (y only).

JumpOrange/jumpBlue: *GameObject*: The Sprites/GameObjects to be generated (powerups).

ItemAppearProbability: *Float: 0-1* The probability of creating items. The higher the number, the more the items will be generated on the level.

EnemyAppearProbability: *Float: 0-1* The probability of creating enemies. The higher the number, the more the enemies will be generated on the level.

Enemies: *GameObject []*: Array of enemy gameobjects.

Thank you

Thank you for purchasing the asset. If you require any help or face any issues, feel free to contact me for support.