



GameDev.tv

Game Jam Starter Kit

Puzzle Platformer Project

About

Thank you for downloading the GameDev.tv Game Jam 'Puzzle Platformer' Starter Kit!

This kit provides you the basic components you'll need to create a simple puzzle platformer game.

The kit including:

- 2D character controller
- Collectable coins and unlockable end-level goal
- Spike hazards
- Tilemap rule tile

This document will guide you through the setup process and explain everything you need to get started!

Contents

About.....	1
Requirements.....	2
Asset Overview	2
Setup	3
Setting Up 2DTilemapExtras	3
Setting Up Collision Layers.....	4
Goal Unlock Mechanic Explained.....	5
Tilemap Rule Tile Explained	5
Code Overview	6
PlayerController.cs	6
LevelManager.cs	6
Spike.cs.....	6
Wallet.cs.....	6
Pickup.cs.....	6
Goal.cs.....	6
Need Help?.....	7

Requirements

- Unity 2021.2 or later
- Input System 1.3 or later
- 2D Tilemap Extras 2.2.1 or later

Asset Overview

Once imported, you will find everything organized into several folders:

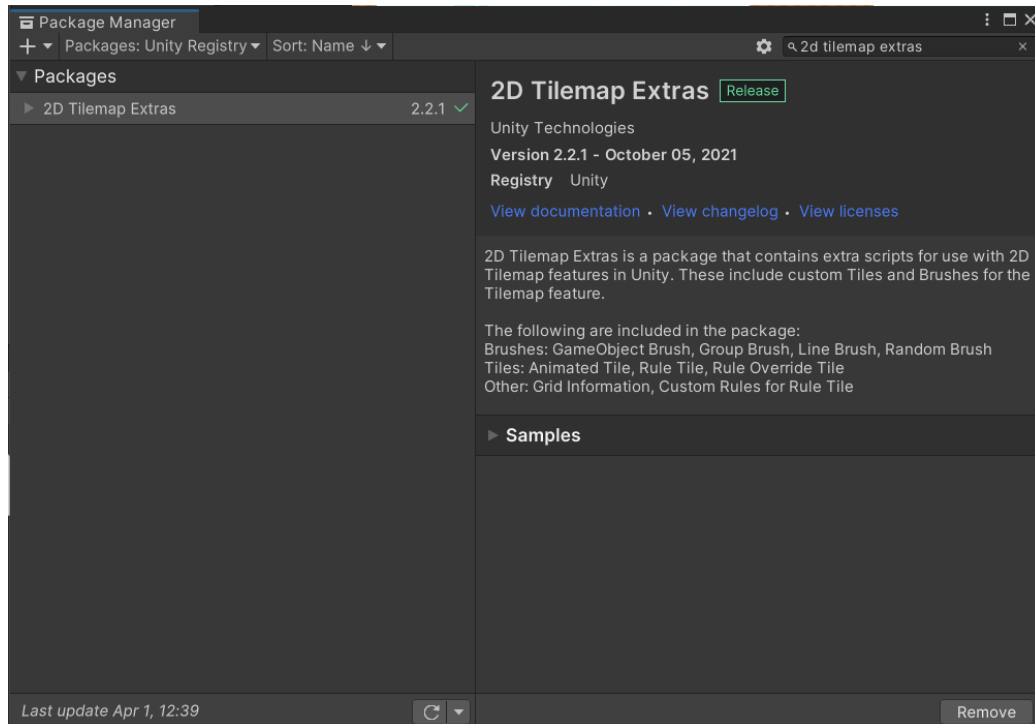
- Asset Packs – Includes several sprites, generously provided by Kenney.nl
- Input – Contains the input settings and actions for the Unity Input System
- Materials – Contains a low-friction physics material that prevent the player from sticking to walls
- Prefab – Includes the; player, and several environment prefabs
- Scenes – Includes the; “Sandbox” demo level
- Scripts – Contains the player controller, level manager, and everything related to the coin pickups and goal unlock msechanic.

Setup

Setting Up 2D Tilemap Extras

This package will allow you to create and use rule tiles in your tilemap.

1. Go to; Window > Package Manager
2. Select “Unity Registry” from the Packages drop-down menu
3. Search for the latest version of 2D Tilemap Extras
4. Press the install button in the bottom right



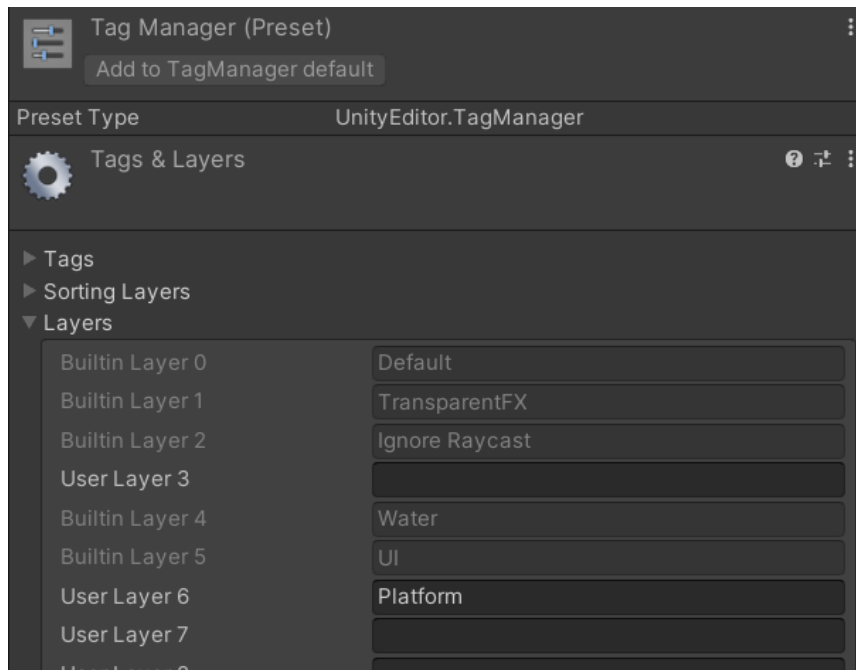
Setting Up Collision Layers

For the player to collide with the platforms, you must set the “Platform Tilemap” gameobject layer to “Platform”.

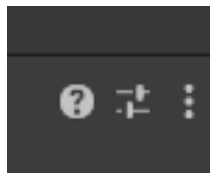
To make setting up the required layers easier, there is a TagManager.preset file in the “Scenes” folder.

To apply these presets,

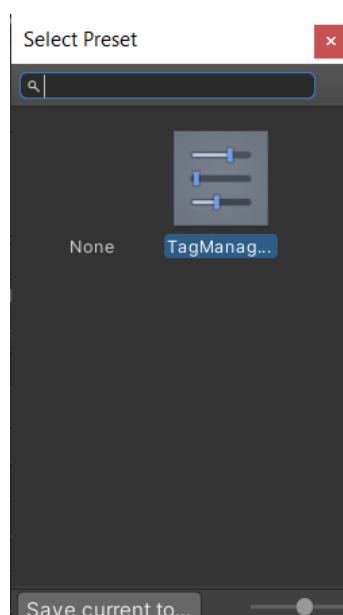
1. Go to the layers window



2. Select the middle ‘slider’ icon in the top right of the window.



3. Click the TagManager preset to apply it



Goal Unlock Mechanic Explained

This project contains a mechanic that prevents the player from completing the level until a set number of coins have been collected, and can be used as the basis for some interesting puzzle mechanics in your game.

In the 'Sandbox' level provided, there are several coins placed around the level. Once 10 have been collected the goal flag will change color and the level will reset once it has been reached.

The number of collected coins is stored in the "Wallet" attached to the player.

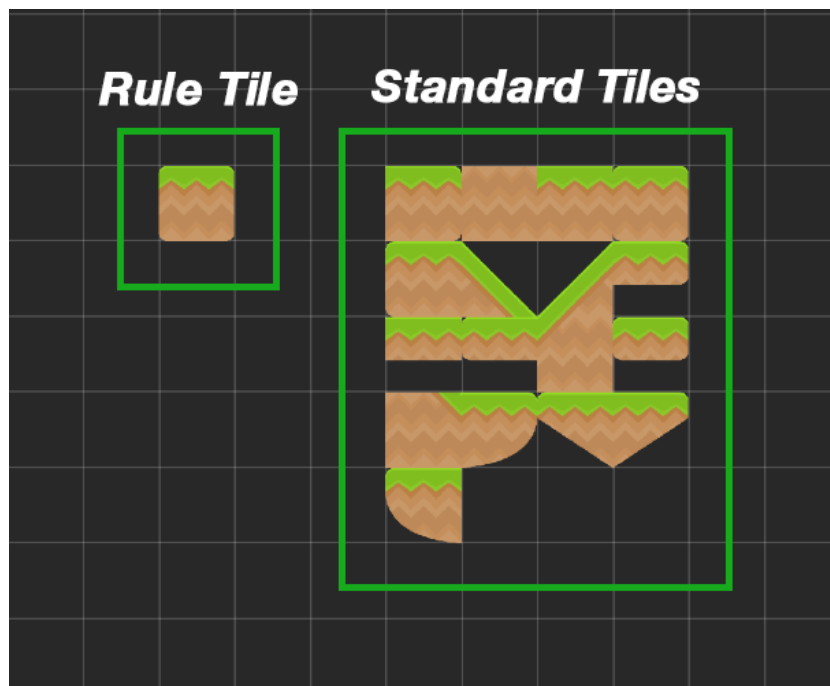
You can set the number of coins required from `Goal.cs`, which is attached to the "GoalFlag" prefab. You can also change the default and unlock colors for the flag from this script.

Tilemap Rule Tile Explained

Within the tilemap palette you will find a rule tile, which can be used to drastically speed up your workflow when creating new level layouts.

Paint this rule tile on your tilemap and it will automatically update the sprite based on the tiles around it.

For additional design control, a standard set of tiles has also been created.



Code Overview

All of the code is organized to be as readable as possible for beginners and additional comments have been added where necessary.

However, here's a brief overview of the included classes.

PlayerController.cs

This script is attached to the "Player" prefab and is responsible for handling the movement of the player.

Input is handled using the new Unity Input system and uses 'Send Message' rather than the more complex event driven system.

LevelManager.cs

This script is attached to the "SceneManager" prefab and is responsible for disabling the player controller and reloading the level.

This is a singleton class and the relevant variables and methods associated with that functionality have been grouped into their own region within the script.

Spike.cs

This script is attached to the "Spike" prefab and will reset the level if the player collides with it.

Wallet.cs

This script is attached to the "Player" prefab and allows you to keep track of how many pickups have been collected.

Pickup.cs

This script is attached to the "Coin" prefab and is responsible for updating the coins variable within `Wallet.cs`.

Goal.cs

This script is attached to the "Goal" will reload the level once the player reaches it with the required number of coins.

This script will also update the color of the flag sprite once it has been unlocked.

Need Help?

If you're not already part of our amazing community, why not come and say hi!

- [Community Forum](#)
- [Discord](#)
- [Facebook](#)
- [Twitter](#)
- [Instagram](#)
- [YouTube](#)

We also have a ton of great courses to help you develop your skills in; Unity, Blender, Pixel Art, and more.

Swing by [GameDev.tv](#) and supercharge your game development skills today!