

2D PLATFORMER

thank you so much for purchasing asset
documentation

Asset

1. INFO
2. INSTALLATION
3. HIERARCHY
4. SCRIPTS
5. IMAGES
6. MOBILE PLATFORM

INFO

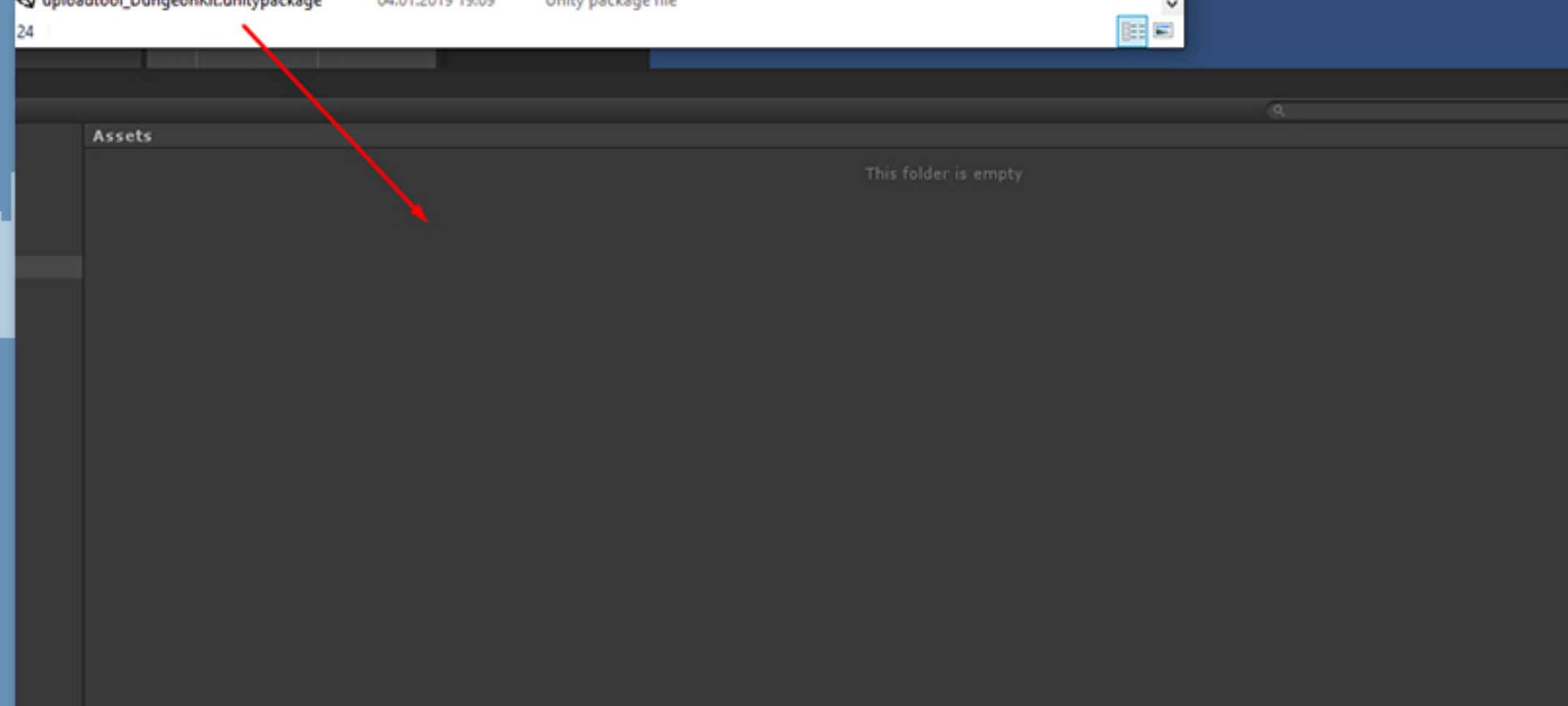
this is an asset of the complete dungeon
game. in this asset:

1. top down controller
2. combat system
3. ai (simple, but easy to edit)
4. dialog system
5. shop
6. level logic and more

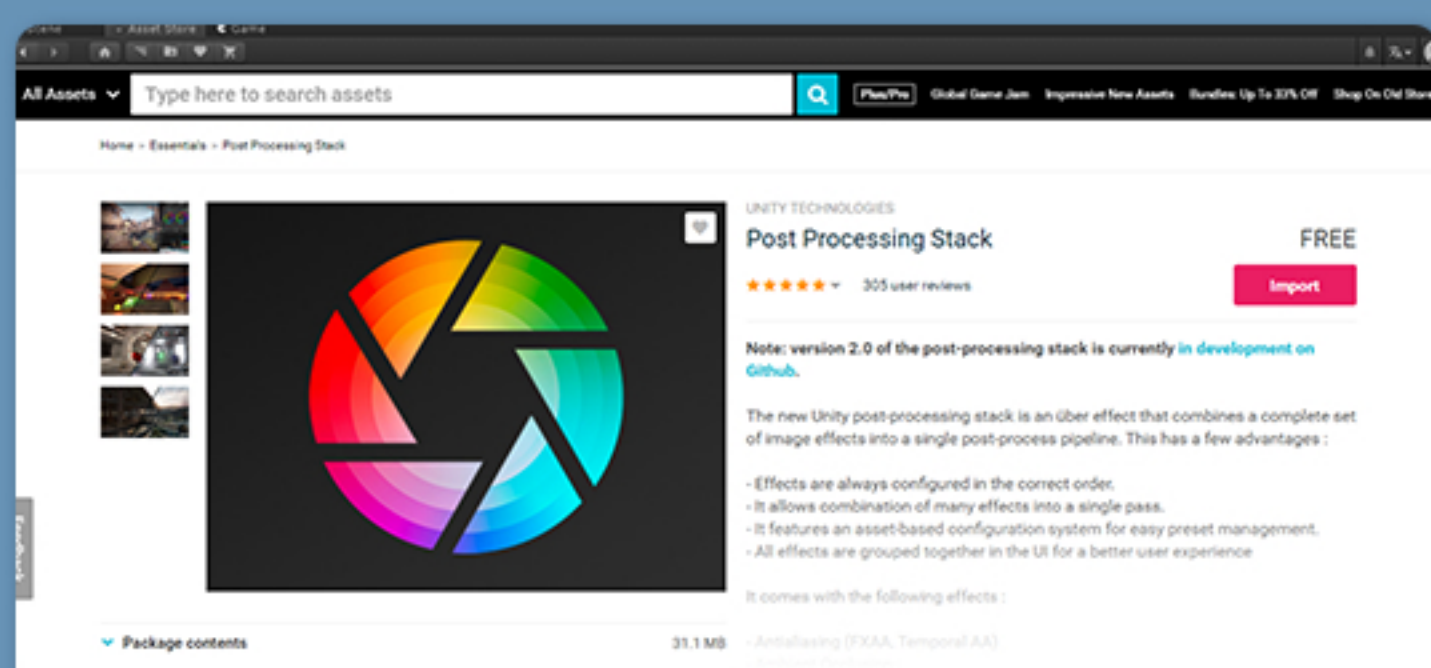
Asset

Installation

transfer the file to the editor



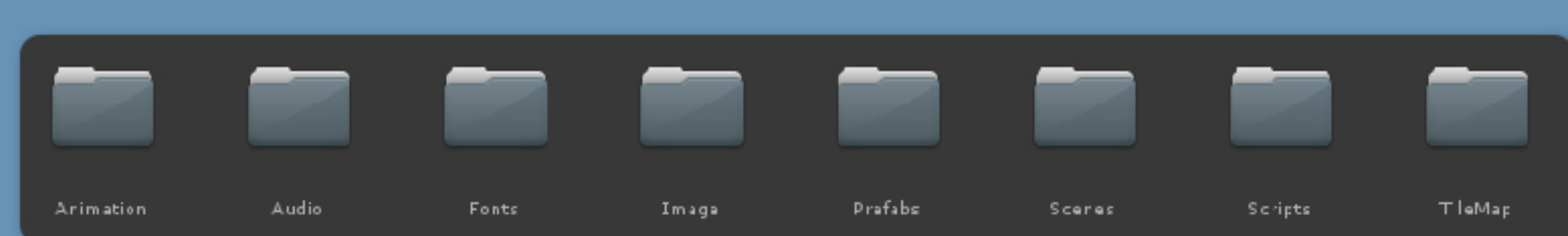
go to the product page, download



Asset

Hierarchy

PROJECT:



- **ANIMATION** - stores the files responsible for the animation .anim, .controller
- **AUDIO** - keeps all the music and sounds
- **Fonts** - keeps all the fonts
- **Images** - stores all images (ui, sprites, etc.)
- **Prefabs** - keeps all copies of game prefabs
- **Scenes** - keeps all the scenes
- **Scripts** - stores all scripts responsible for logic
- **Tilemap** - stores all files associated with the tilemap system

Scene:

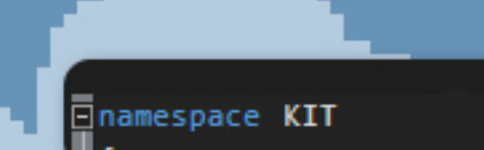


- **main camera** - main camera on scene
- **player** - player gameobject
- **next level door** - next level door gameobject
- **[level]** - keeps most of the level
- **[level]/grid** - level tilemap
- **[level]/static** - static objects such as boxes, etc
- **[level]/ai** - keeps ai gameobjects
- **[managers]** - keeps all managers
- **[managers]/eventsystem** - default object for ui to work correctly (do not delete)
- **[managers]/managers** - on the object are scripts responsible for all the logic

Asset

Scripts

editing scripts is not difficult if you are familiar with c#



- **namespace** is used for all scripts to avoid conflicts with your scripts.

```
//Play animation in animator PlayAnimation("Animation name")
public void PlayAnimation(string animationName)
{
    animator.Play(animationName);
}

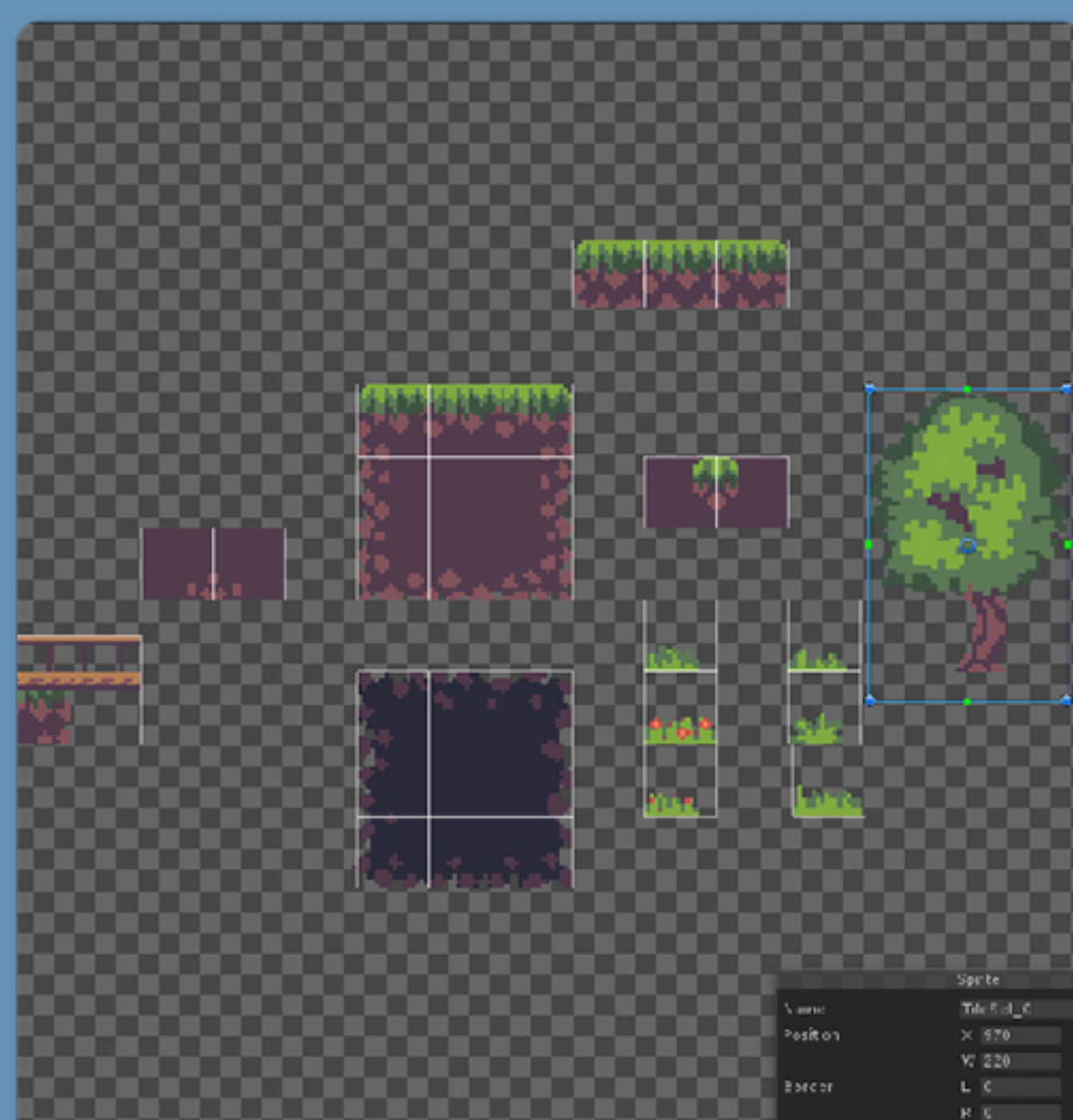
//Play clip in playable director PlayPlayableDirector(timelineAssets[id], DirectorWrapMode. )
public void PlayPlayableDirector(TimelineAsset timelineAsset, DirectorWrapMode directorWrap)
{
    playableDirector.extrapolationMode = directorWrap; //Set director wrap mode in playableDirector
    playableDirector.playableAsset = timelineAsset; // Set clip
    playableDirector.Play(); //Play
}
```

- **all code** has comments describing each action, so editing is easy

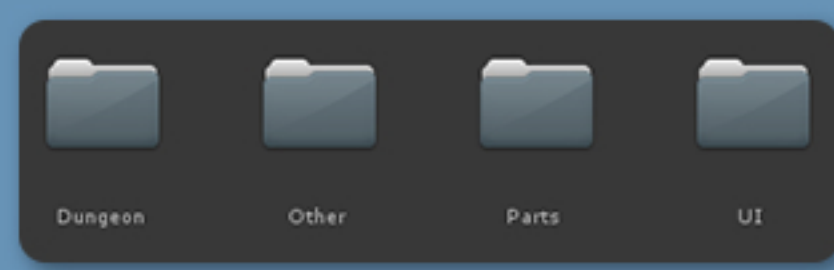
Asset

Images

images are pretty easy to edit.



- if you want to redraw the level, you can simply edit the file **dungeon_tileset**, save with the replacement and the tiles will automatically change

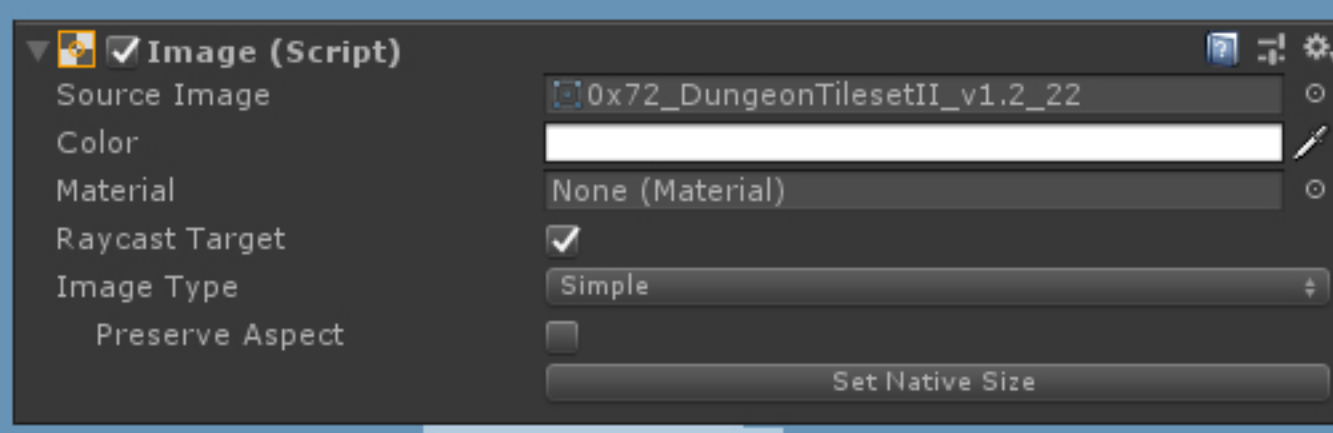


- **everything else** you will find in the folder **image**

- **sprite objects** on the scene have a component **sprite renderer**, takes **sprite**



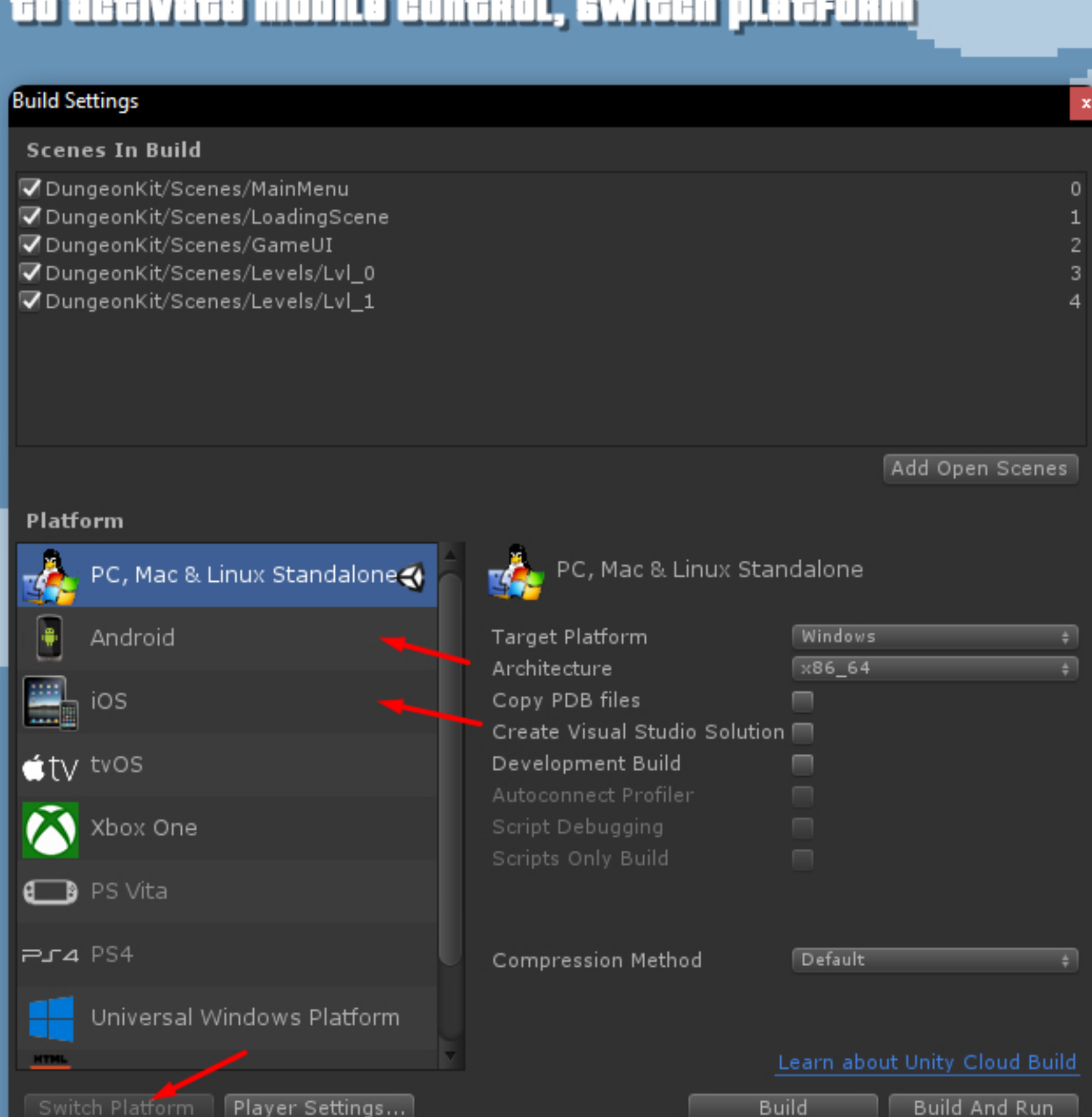
- **ui objects** have an **image** component, takes **sprite**



Asset

Platform

to activate mobile control, switch platform



when you turn on the mobile platform, the joysticks will run when you go to levels