

# Infinite Jumper Starter Kit

1.0

If you have any questions, or suggestions for improvements, please email  
[robert.wells@gandhigames.co.uk](mailto:robert.wells@gandhigames.co.uk).

## Initial Setup

Infinite Jumper requires the following tags and layers.

Tags:

- Darkness
- Platform
- Shadow

Tag 0	Darkness
Tag 1	Platform
Tag 2	Shadow

Layers:

- Platform

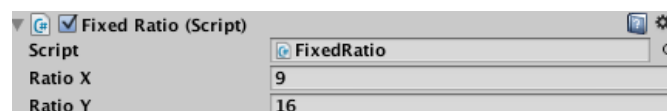
Builtin Layer 0	Default
Builtin Layer 1	TransparentFX
Builtin Layer 2	Ignore Raycast
Builtin Layer 3	
Builtin Layer 4	Water
Builtin Layer 5	UI
Builtin Layer 6	
Builtin Layer 7	
User Layer 8	Platform
User Layer 9	
User Layer 10	
User Layer 11	

Lastly, add the two scenes to the build settings (File -> Build Settings).

Scenes In Build	
<input checked="" type="checkbox"/> InfiniteJumper/Scenes/MainMenu.unity	0
<input checked="" type="checkbox"/> InfiniteJumper/Scenes/Game.unity	1

## Camera Aspect Ratio

Attached to the camera object in each scene is a FixedRatio class. This class locks the screen aspect ratio at 16:9 (used by the majority of mobile devices) when running on a mobile device. Depending on your preference you can change the aspect ratio in the inspector or remove this component completely.



## Adding Level Sections

Overview:

1. Create level text file (or edit existing) and save in 'Resources/Levels' folder.
2. Add LevelChunk component to Level Builder in Game scene hierarchy.

3. Fill in the Level File Name (the name of the file you saved in step 1) and give the level section a weight (described in the next section).
4. And that's it, you're good to go! The level section will be loaded and added to the pool of sections that can be used.

In Detail:

The game's level is split up into a number of sections. The sections are loaded from text files, located under the 'Resources/levels' folder. A sample text file will contain a number of characters (as shown below).

```

x - p - - - - - p - x
x - s - - - p p p s - x
x - - - - s s s - - x
x - - - - - p p p x
x - - - - - p s s s x
x - - - p p p s - - x
x - - - s s s - - - x
x p p - - - - - - x
x s s - - - - - - x
x - - - p p p - - - x
x - - - s s p p - - x
x - - - - s p p p - x
x - - - - s s s s - x
x - - - - - s - - p x
x - - - - - p p p s x
x - - - p p p s s s - x
x p p - s s s - - - x
x s s p p - - - - x
x - - s s p p p p - x
x - - - - s s s s - x

```

These characters are read at runtime and converted into tiles to be used in the game. They are read as follows:

x = wall  
p = platform  
s = shadow  
- = ignored

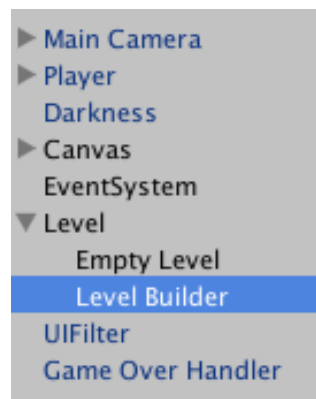
To create a platforming section you would start by copying the emptyLevel file (under Resources/Levels) and then add platforms by changing the '-' character to 'p' and 's'. For example:

x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	p	p	p	p	-	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	s	s	s	s	-	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	p	-	-	-	p	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	s	-	-	-	s	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	-	p	p	-	-	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	-	s	s	-	-	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	p	-	-	-	-	-	-	p	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	p	p	p	-	-	-	-	p	p	p	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	s	p	s	-	-	-	-	s	p	s	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	s	-	p	-	-	p	-	s	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	s	-	-	s	-	s	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	p	-	-	-	-	-	-	p	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	s	-	-	-	-	-	-	s	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	-	p	p	-	p	-	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	p	-	s	s	-	p	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	p	-	-	-	p	p	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	p	p	-	-	s	s	-	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	p	-	p	s	-	-	-	s	-	p	x	
x	-	-	-	-	-	-	-	-	-	-	x	x	s	-	s	-	-	-	-	p	p	p	s	x
x	-	-	-	-	-	-	-	-	-	-	x	x	-	-	-	s	-	-	-	s	s	s	-	x

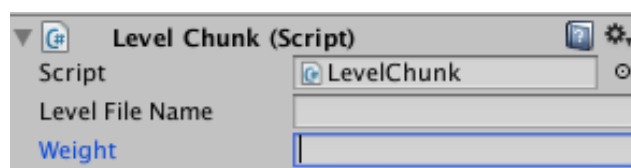
*Empty level file on left, my edits on the right.*

Save this file in the 'Resources/Levels' folder (I called my file level8).

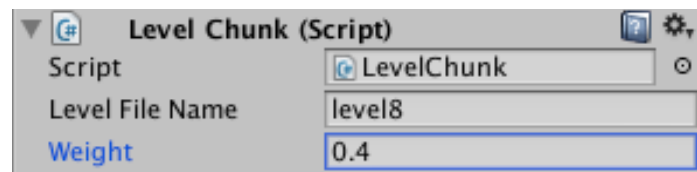
The next step is to open the 'Game Scene' and in the hierarchy select the Level Builder:



Add a new 'LevelChunk' component, like so:



And fill in the details: provide the level name that you saved the file under (in this example I named it level8) and provide a weight. The weighting system is described in the next section, for now I just set that at 0.4.



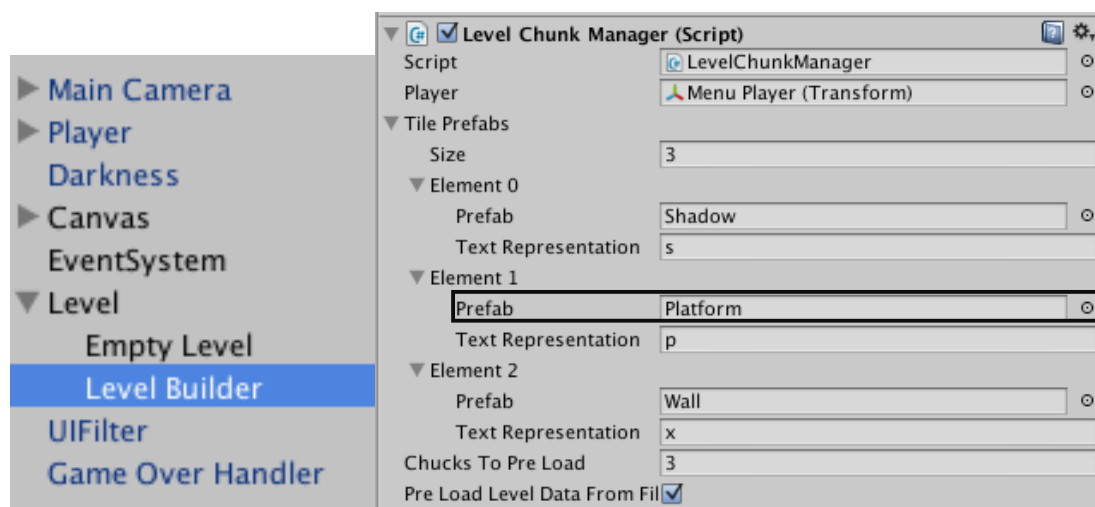
And that's it, your level section is now added to the pool of possible sections that will be used in the game.

## Level Chunk Weighting

The weighting provided for each sections defines how often the section will spawn. It is relative to the other section weights. For example, if you have 3 sections with weightings of 1, 2, and 3. The third section is three times more likely to spawn than the first, while the second section is two times more likely to spawn. Using this system you can create harder sections that spawn less often and vice versa.

## Change Characters Used in The Text Files

If you would like to change what characters are used in the text file (e.g. instead of an 'x' for the wall, you could use a 'w') select the Level Builder in the Game scene hierarchy and examine the LevelChunkManager component.



Under the Tile Prefabs heading, you will find the Text Representation field. Change this to your desired character.

## Change Tiles

To change the tiles used in the game, either edit the Shadow, Wall, and Platform prefabs directly (stored in the Prefab folder), or follow the steps for changing the text representation (outlined above) and then change the prefabs associated with the Tile Prefabs on the LevelChunkManager object.

