Documentation

Vehicle Factory

Warning! We would recommend you to make a reskin to prevent problems with Google Play and App Store (game can be banned).

Integration of side plugins is your own responsibility. We don’t support modified projects (except reskin).

We are not engaged in additional customization of the project.

Technical Requirements

For correct work of the project, please observe all requirements

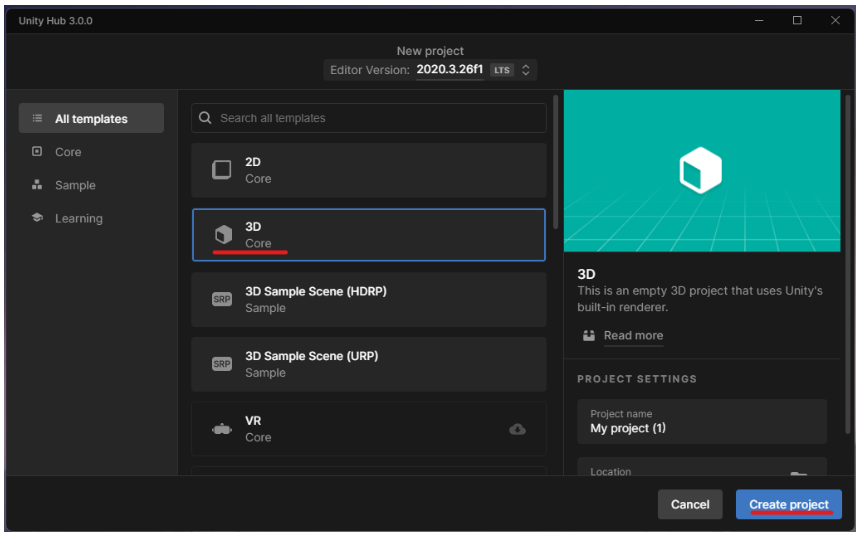
Unity version: Last LTS version (Unity 2020.3.X)
Target platforms: iOS, Android
Minimum iOS version: 10.0
Minimum Android version: API Level 29
Scripting Runtime Version: .NET 4.x Equivalent

Project Structure

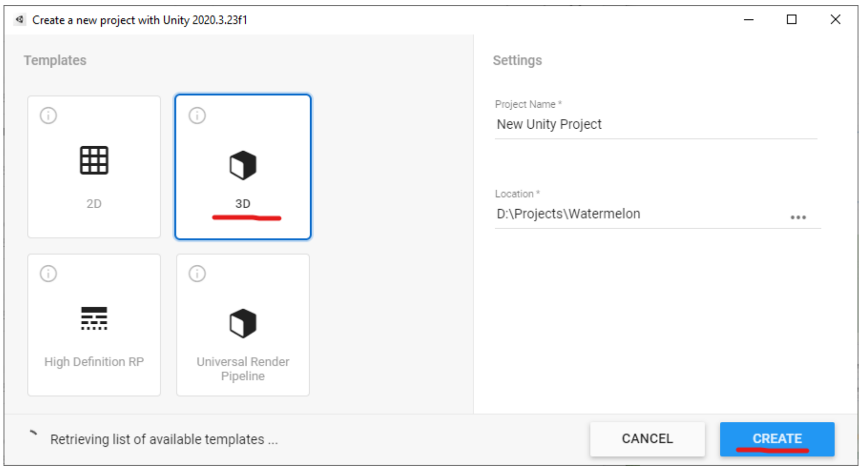
Assets/Project Data/Content - folder with all project settings files and databases.
Assets/Project Data/Content/Settings - folder with all basic game settings.
Assets/Project Data/Content/Settings/Editor/Define Settings.asset - define manager.
Assets/Project Data/Content/Settings/Ads Settings.asset - advertising settings.
Assets/Project Data/Content/Settings/Audio Settings.asset - audio settings.
Assets/Project Data/Content/Settings/IAP Settings.asset - IAP settings.
Assets/Project Data/Content/Settings/Project Init Settings.asset - init settings.
Assets/Project Data/Content/LevelsSystem/Level Database.asset - levels database object.
Assets/Project Data/Content/LevelsSystem/Levels/ - levels files location.
Assets/Project Data/Game/ - game resources.
Assets/Project Data/Game/Scenes/ - scenes folder.
Assets/Project Data/Game/Audio/ - game audio files.
Assets/Project Data/Game/Images/Icon/Icon.png - game icon.

How to start

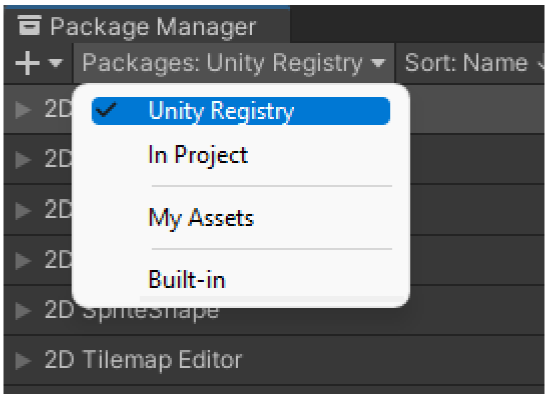
- Download and install recommended Unity version - [Download]
- Create a new Unity project using 3D template.



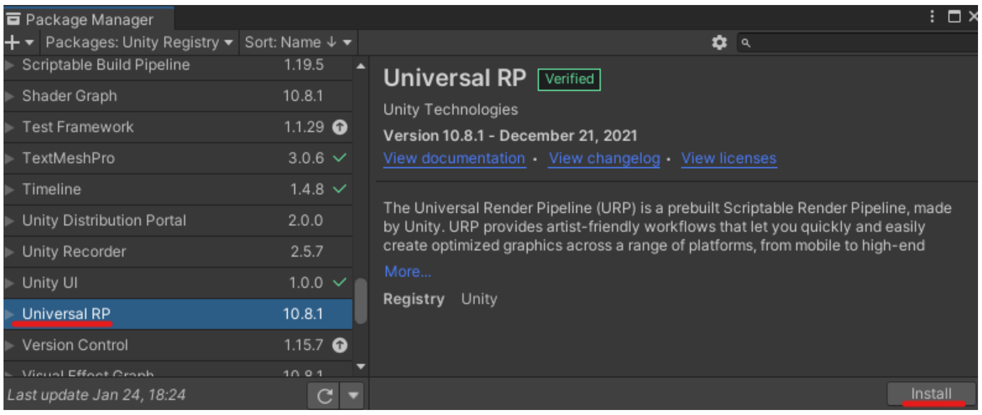
or in older version



- Import URP package:
  a. Click **Window** -> **Package Manager**
  b. On left top select **Packages: Unity Registry**



c. Find **Universal RP** and click **Install**



Scriptable Build Pipeline	1.19.5
Shader Graph	10.8.1
Test Framework	1.1.29
TextMeshPro	3.0.6 ✓
Timeline	1.4.8 ✓
Unity Distribution Portal	2.0.0
Unity Recorder	2.5.7
Unity UI	1.0.0 ✓
Universal RP	10.8.1
Version Control	1.15.7

Last update Jan 24, 18:24

Universal RP Verified

Unity Technologies

Version 10.8.1 - December 21, 2021

View documentation • View changelog • View licenses

The Universal Render Pipeline (URP) is a prebuilt Scriptable Render Pipeline, made by Unity. URP provides artist-friendly workflows that let you quickly and easily create optimized graphics across a range of platforms, from mobile to high-end

More...

Registry Unity

- You can remove **Scenes** folder generated by default.
- Import template:
  a. You can use **Asset Store** (if you bought there)
  b. Or manually add package: click **Assets** -> **Import Package** -> **Custom Package**
- Open **Build Settings** (File -> Build Settings):
  a. Add all scenes (from Scenes folder) in the right order to **“Scenes In Build”**.
  b. Select target platform **Android** or **IOS**
  c. Click **Switch Platform**
- Open Park Inc\Game\Scenes\Game scene to run the game.
- Build the game.

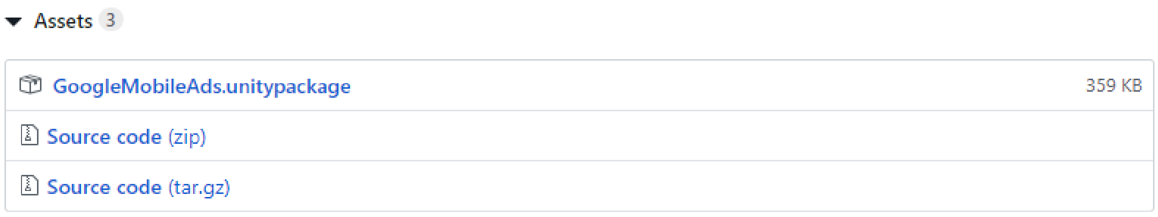
IMPORTANT: To quickly access all important files and settings use **Setup Guide** (Tools -> Project Setup Guide)

Advertisement Setup

AdMob

Google Mobile Ads Unity plugin integration guide - link

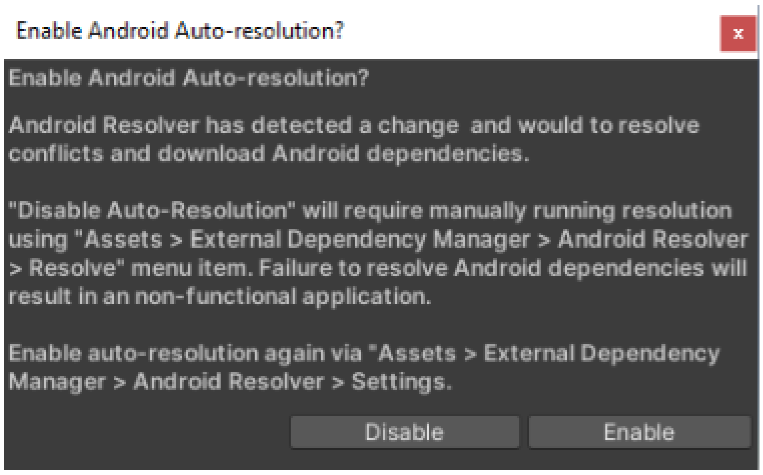
1. Download the latest version of Google Mobile Ads Plugin - download



Assets 3

GoogleMobileAds.unitypackage	359 KB
Source code (zip)	
Source code (tar.gz)	

2. Click Assets - Import Package - Custom Package and select the downloaded file to import.
3. [For ANDROID] If next pop up appears:



Enable Android Auto-resolution?

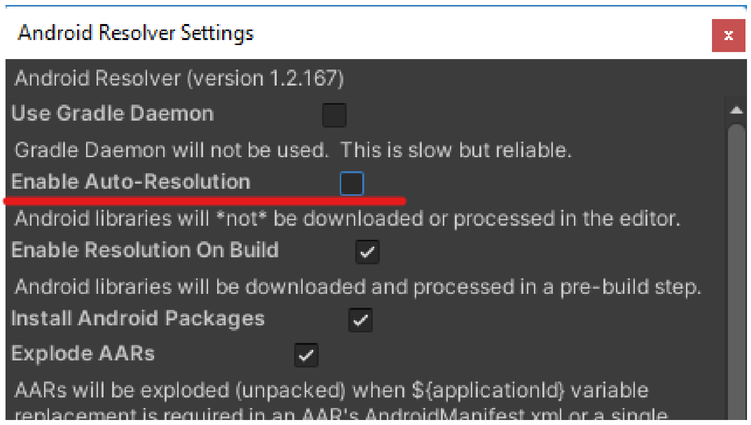
Enable Android Auto-resolution?

Android Resolver has detected a change and would to resolve conflicts and download Android dependencies.

"Disable Auto-Resolution" will require manually running resolution using "Assets > External Dependency Manager > Android Resolver > Resolve" menu item. Failure to resolve Android dependencies will result in an non-functional application.

Enable auto-resolution again via "Assets > External Dependency Manager > Android Resolver > Settings.

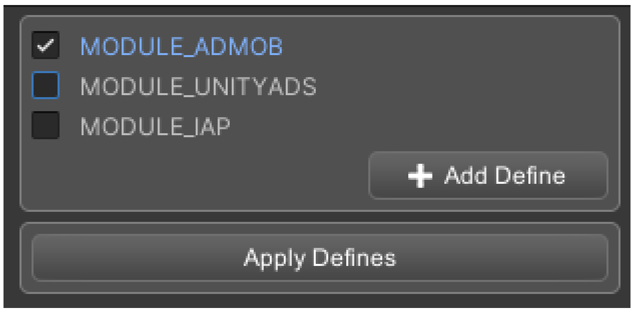
Click **Disable**
Otherwise click Assets - External Dependency Manager - Android Resolver - Settings and uncheck **Enable Auto-Resolution** scroll down and click **OK**



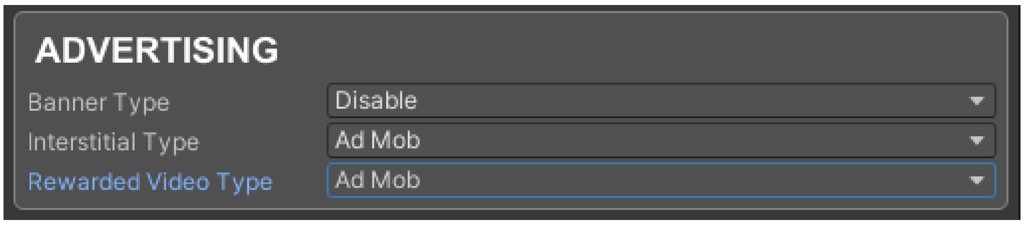
Android Resolver Settings

Android Resolver (version 1.2.167)
Use Gradle Daemon ☐
Gradle Daemon will not be used. This is slow but reliable.
Enable Auto-Resolution ☐
Android libraries will *not* be downloaded or processed in the editor.
Enable Resolution On Build ✓
Android libraries will be downloaded and processed in a pre-build step.
Install Android Packages ✓
Explode AARs ✓
AARs will be exploded (unpacked) when ${applicationId} variable replacement is required in an AAR's AndroidManifest.xml or a single

4. Click Tools - Editor - Define Manager to select **Define Manager** asset.
5. Enable **MODULE_ADMOB** and press **Apply Defines**.

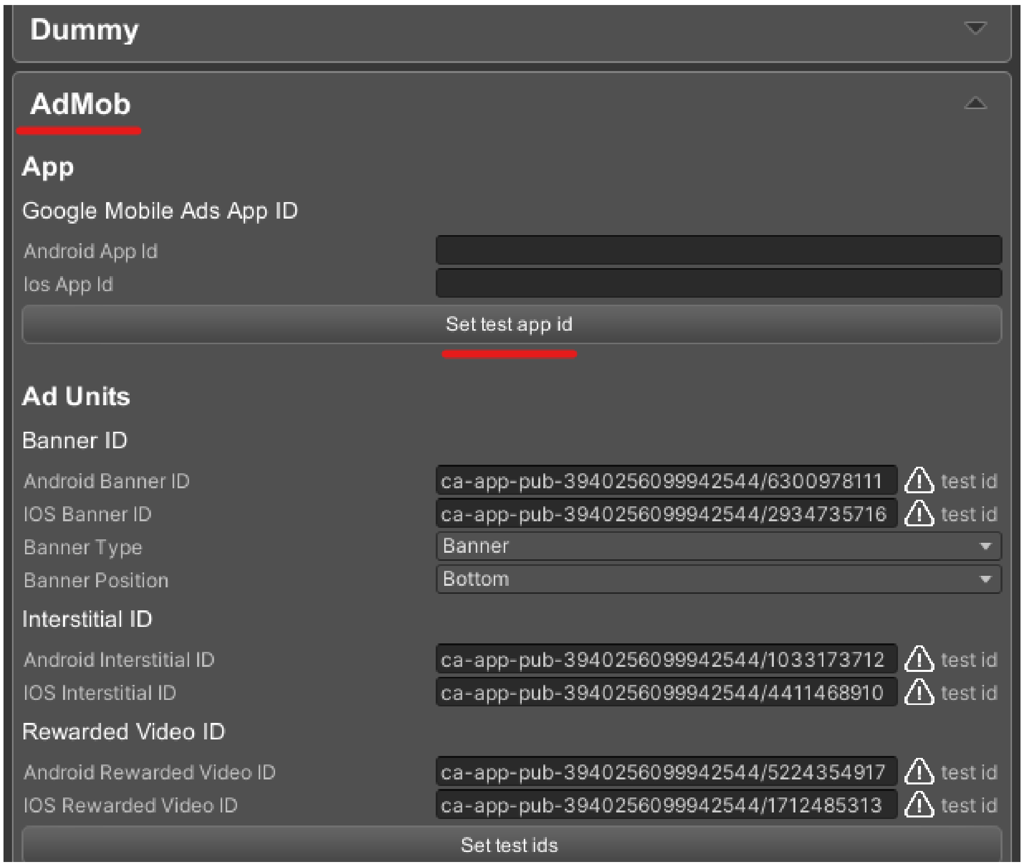

- [✓] MODULE_ADMOB
- [ ] MODULE_UNITYADS
- [ ] MODULE_IAP

6. Open **Advertising** tab in **Setup Guide** window (Tools -> Project Setup Guide).
7. Switch to **AdMob** or Disable for each ads type depending on your needs.


ADVERTISING

Banner Type	Disable
Interstitial Type	Ad Mob
Rewarded Video Type	Ad Mob

8. Unfold **AdMob** tab located below. Click **Set test app id** button.


Dummy

AdMob

App

Google Mobile Ads App ID

Android App Id	
Ios App Id	

Ad Units

Banner ID

Android Banner ID	ca-app-pub-3940256099942544/6300978111	test id
IOS Banner ID	ca-app-pub-3940256099942544/2934735716	test id
Banner Type	Banner	
Banner Position	Bottom	

Interstitial ID

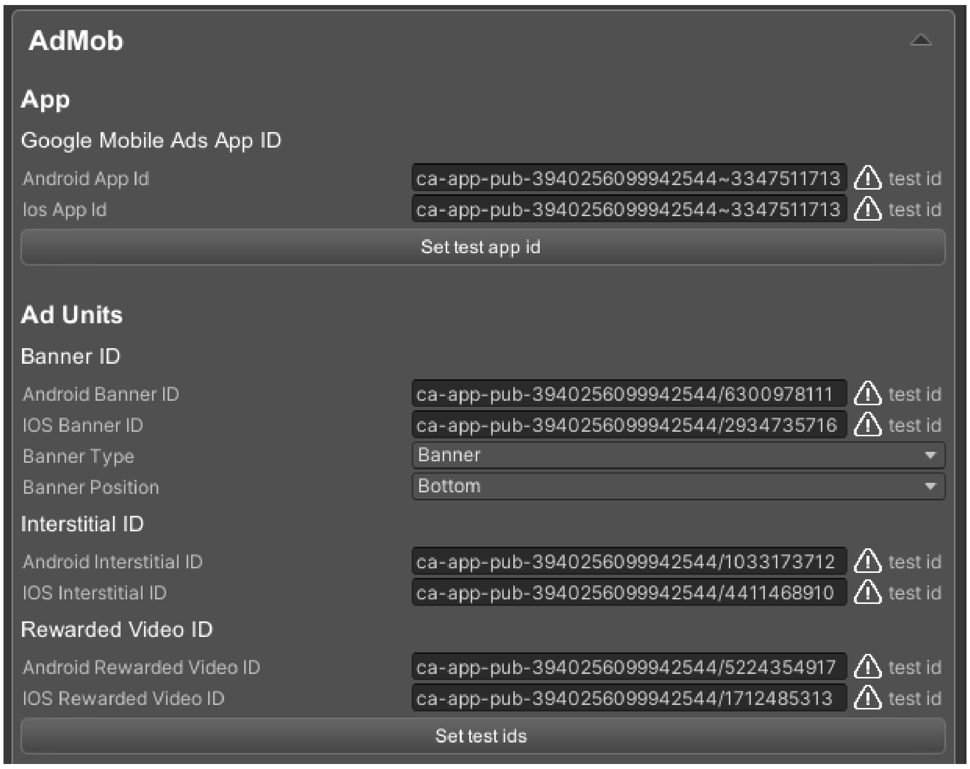
Android Interstitial ID	ca-app-pub-3940256099942544/1033173712	test id
IOS Interstitial ID	ca-app-pub-3940256099942544/4411468910	test id

Rewarded Video ID

Android Rewarded Video ID	ca-app-pub-3940256099942544/5224354917	test id
IOS Rewarded Video ID	ca-app-pub-3940256099942544/1712485313	test id

Now you can test your app with default ids.
Follow the next steps only when you're ready to publish the game, it's an AdMob requirement.

9. Go to your Google Mobile Ads account - link
10. Set up an app in AdMob. Help
11. Open **Advertising** tab in **Setup Guide** window (Tools -> Project Setup Guide). Unfold **AdMob** block and enter data from the website into appropriate fields.



AdMob

App

Google Mobile Ads App ID

Android App Id	ca-app-pub-3940256099942544~3347511713	⚠ test id
Ios App Id	ca-app-pub-3940256099942544~3347511713	⚠ test id

Ad Units

Banner ID

Android Banner ID	ca-app-pub-3940256099942544/6300978111	⚠ test id
IOS Banner ID	ca-app-pub-3940256099942544/2934735716	⚠ test id
Banner Type	Banner	
Banner Position	Bottom	

Interstitial ID

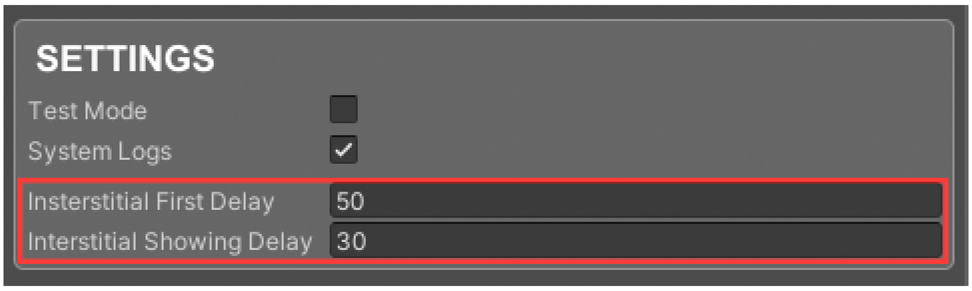
Android Interstitial ID	ca-app-pub-3940256099942544/1033173712	⚠ test id
IOS Interstitial ID	ca-app-pub-3940256099942544/4411468910	⚠ test id

Rewarded Video ID

Android Rewarded Video ID	ca-app-pub-3940256099942544/5224354917	⚠ test id
IOS Rewarded Video ID	ca-app-pub-3940256099942544/1712485313	⚠ test id

Make sure there's no "test id" warnings on the right side of the fields.

12. Click Assets - External Dependency Manager - Android Resolver - Settings and check **Enable Auto-Resolution** scroll down and click **OK**
13. If resolution did not start automatically click Assets - External Dependency Manager - Android Resolver - Resolve
14. If you want to change the advertising frequency, set it on the Advertising tab.



SETTINGS

Test Mode	☐
System Logs	✓
Insterstitial First Delay	50
Interstitial Showing Delay	30

Interstitial First Delay - delay in seconds between game launch and first interstitial appearing.

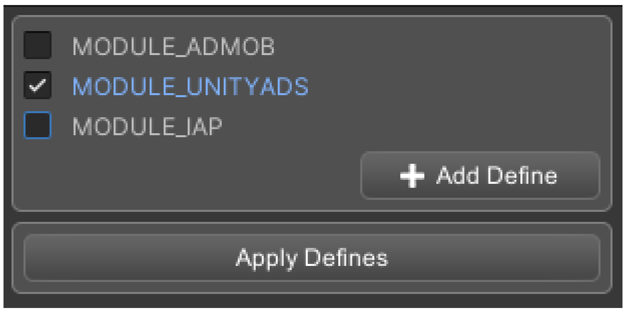
Interstitial Showing Delay - min delay in seconds between interstitial appearings.

15. Now you can publish the game.
Note, after publishing you'll need to wait until AdMob approves the game. More info here.

Unity Ads

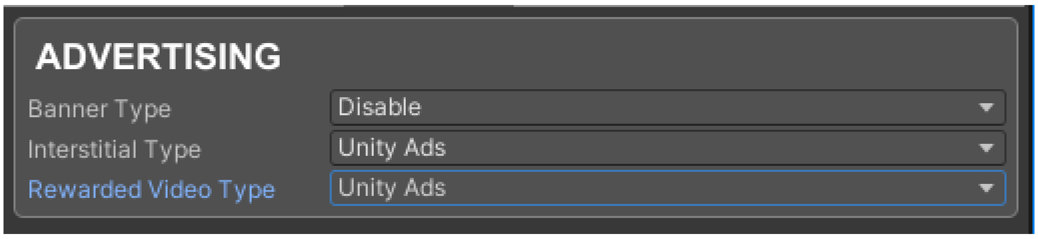
Unity Ads quick start guide - link

1. Follow Unity Ads quick start guide to setup services.
2. Click Tools - Editor - Define Manager to select Define Manager asset.
3. Enable **MODULE_UNITYADS** and press **Apply Defines**.



- [ ] MODULE_ADMOB
- [x] MODULE_UNITYADS
- [ ] MODULE_IAP

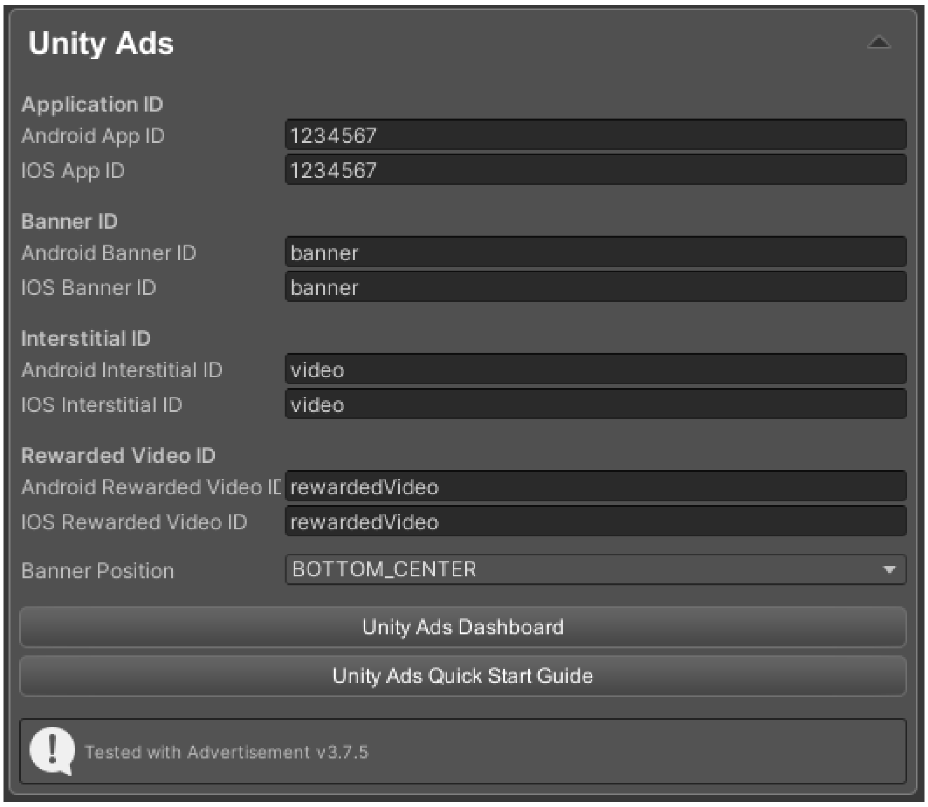
4. Open **Advertising** tab in **Setup Guide**.
5. Switch to **Unity Ads** or Disable for each ads type depending on your needs.



ADVERTISING

Banner Type	Disable
Interstitial Type	Unity Ads
Rewarded Video Type	Unity Ads

6. Unfold **Unity Ads** block at window bottom and put data from the site in the appropriate fields.

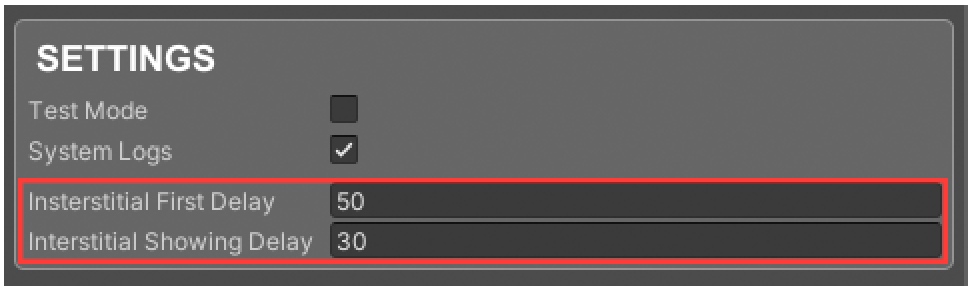


Unity Ads

Application ID	
Android App ID	1234567
IOS App ID	1234567
Banner ID	
Android Banner ID	banner
IOS Banner ID	banner
Interstitial ID	
Android Interstitial ID	video
IOS Interstitial ID	video
Rewarded Video ID	
Android Rewarded Video ID	rewardedVideo
IOS Rewarded Video ID	rewardedVideo
Banner Position	BOTTOM_CENTER

Tested with Advertisement v3.7.5

7. If you want to change the advertising frequency, set it on the Advertising tab.


SETTINGS

Test Mode ☐

System Logs ✓

Insterstitial First Delay 50

Interstitial Showing Delay 30

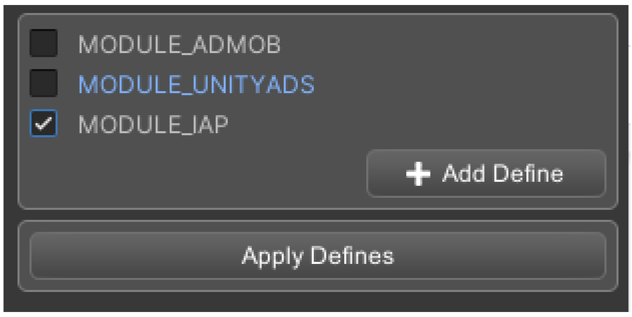
Interstitial First Delay - delay in seconds between game launch and first interstitial appearing.

Interstitial Showing Delay - min delay in seconds between interstitial appearings.

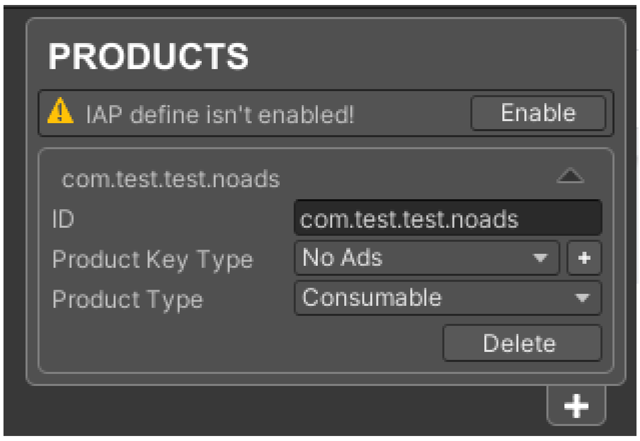
IAP Setup

Setting up Unity IAP integration guide - [link]()

1. Follow the guide above to import IAP package
2. Select Define Manager (Tools - Editor - Define Manager)
3. Check **MODULE_IAP** and press on **Apply Defines** button



4. Open **Products** tab in **Setup Guide** window



IAP define isn't enabled!

com.test.test.noads

ID: com.test.test.noads

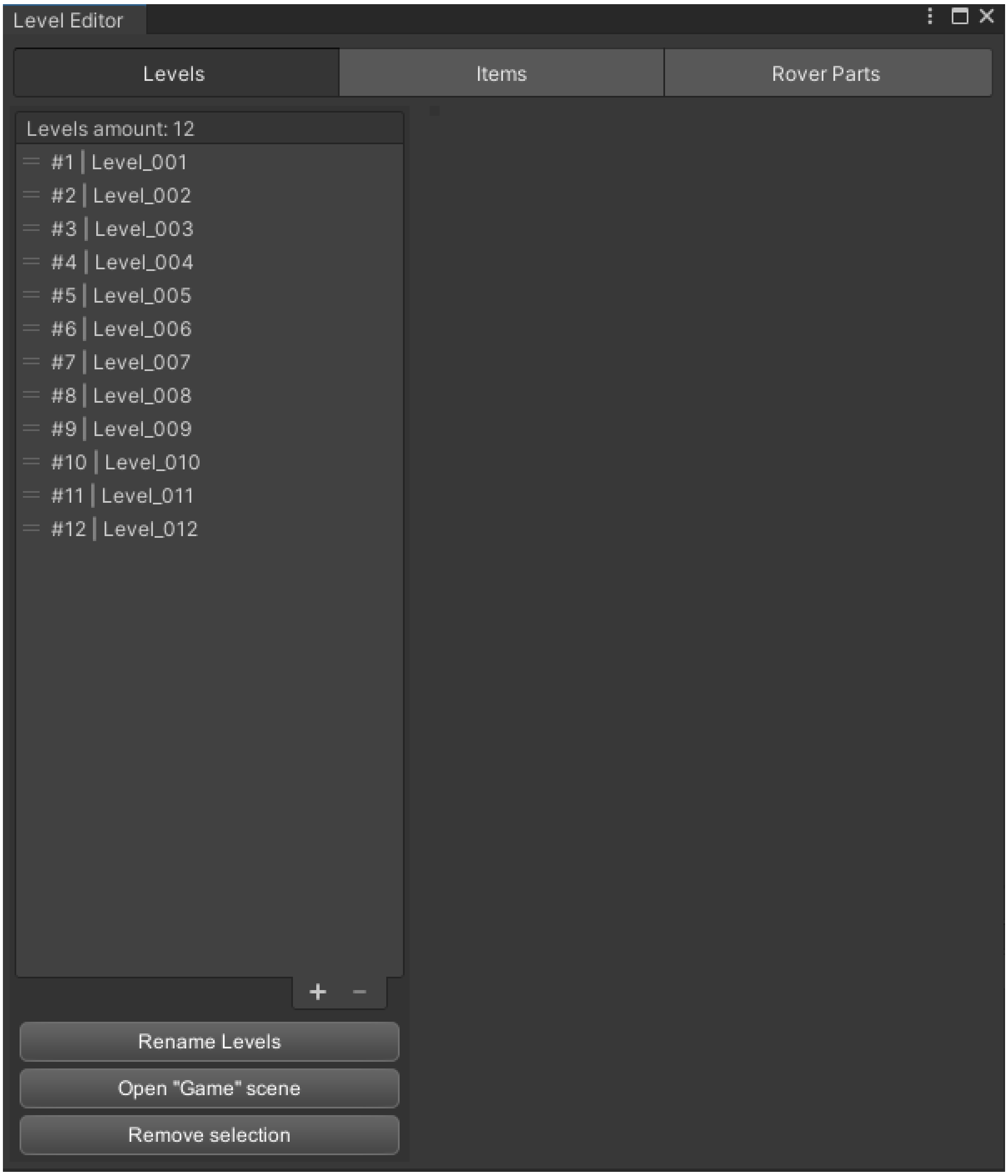
Product Key Type: No Ads

Product Type: Consumable

5. Change default ID with your

Level editor

Open LevelEditorWindow : Tools - Level Editor.

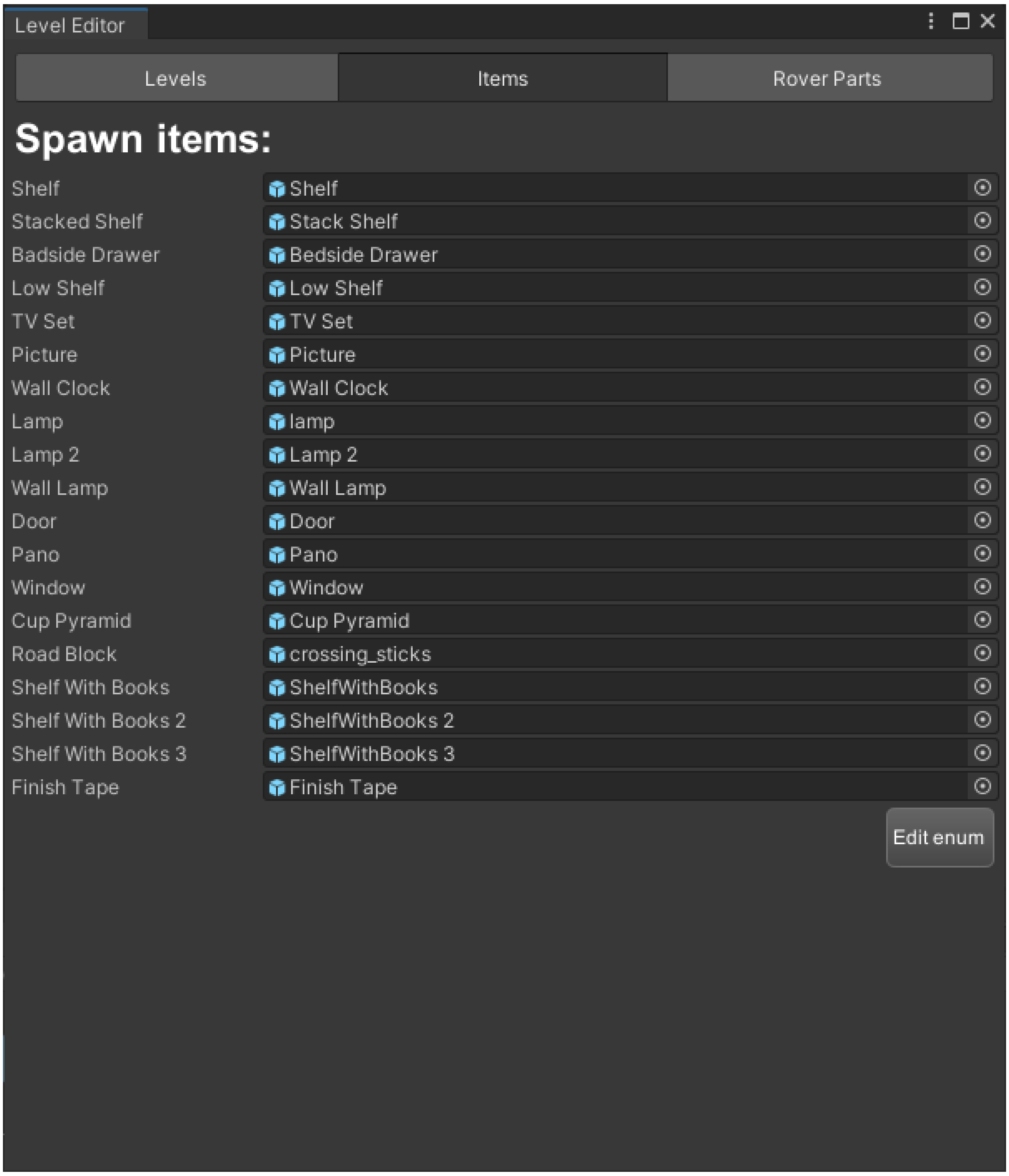


At the top of the window, you can see 3 tabs:

- **Levels** - for creating and modifying existing levels.
- **Items -** for modifying the list of available items.

- **Rover parts** - for editing rover parts and their slots.

It's better to configure Items and Rover parts before you start creating levels. That's why we look at these 2 tabs first.



Level Editor

Levels | Items | Rover Parts

Spawn items:

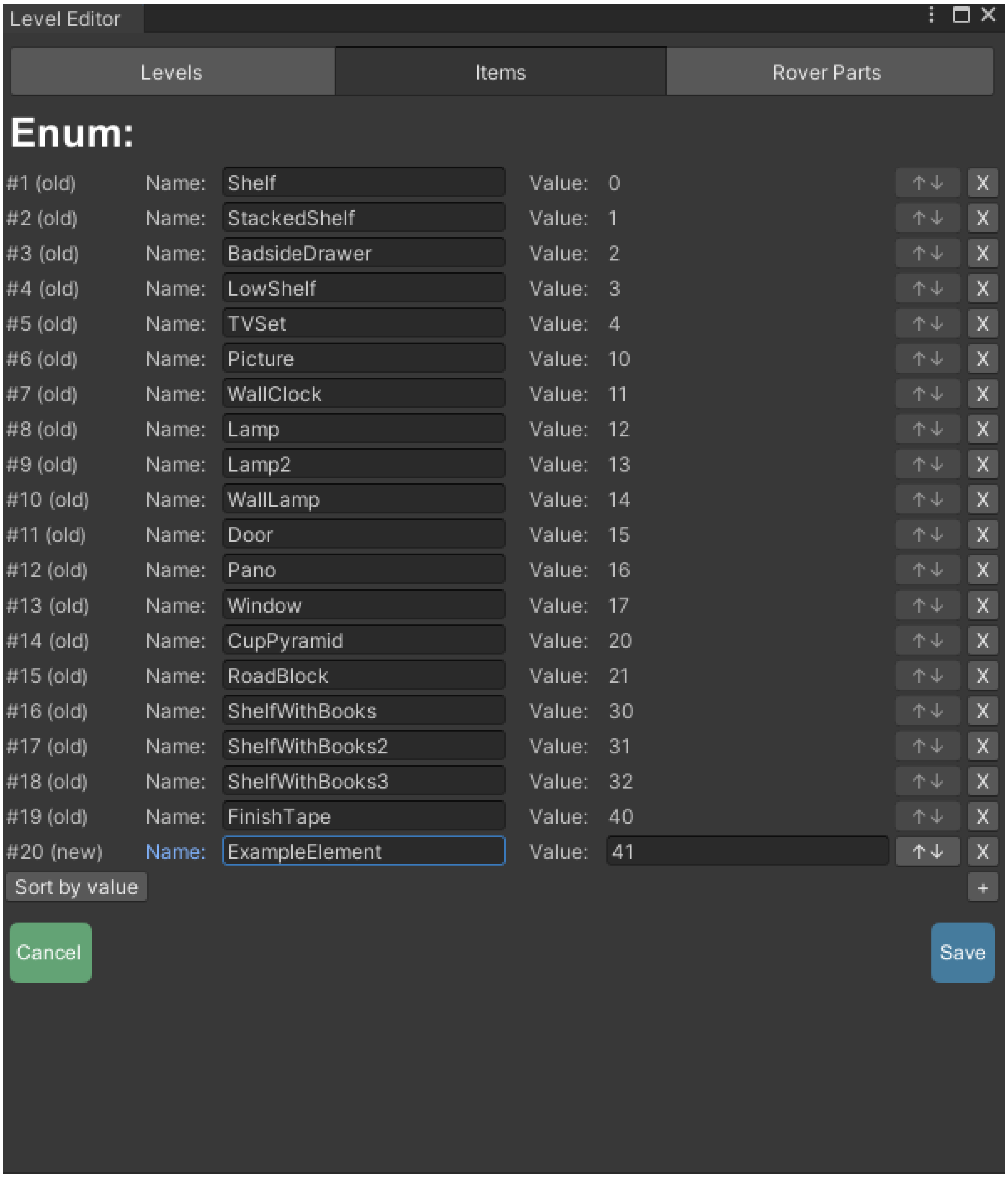
Shelf	Shelf
Stacked Shelf	Stack Shelf
Badside Drawer	Bedside Drawer
Low Shelf	Low Shelf
TV Set	TV Set
Picture	Picture
Wall Clock	Wall Clock
Lamp	lamp
Lamp 2	Lamp 2
Wall Lamp	Wall Lamp
Door	Door
Pano	Pano
Window	Window
Cup Pyramid	Cup Pyramid
Road Block	crossing_sticks
Shelf With Books	ShelfWithBooks
Shelf With Books 2	ShelfWithBooks 2
Shelf With Books 3	ShelfWithBooks 3
Finish Tape	Finish Tape

Edit enum

How to add new items:

1. Press the "Edit enum" button.
2. Press the "+" button to add a new element.
3. Write a unique "name" for the new element.

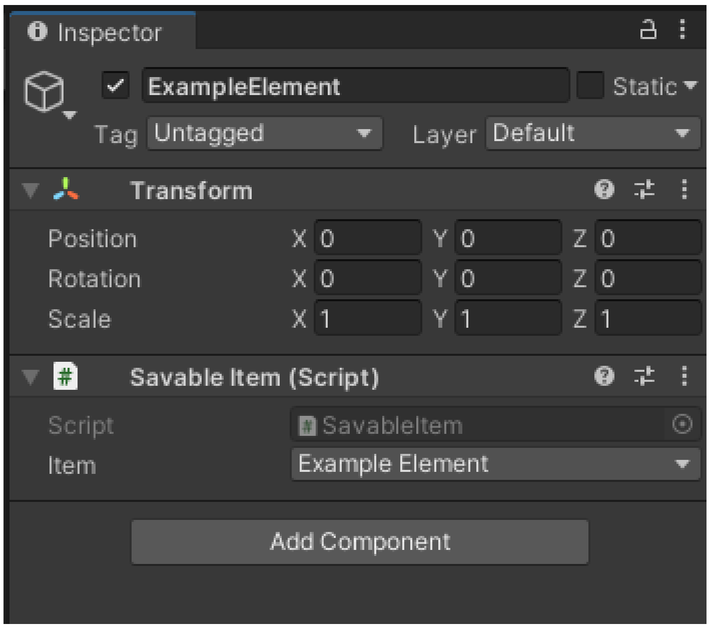
4. If you wish, you can write a unique "value" for the new element. Or use "↑↓" to move it to a safe spot.
5. If you have done everything correctly you have something similar to the example below. Otherwise, you need to follow instructions and resolve validation errors.



Enum:

#1 (old)	Name:	Shelf	Value:	0
#2 (old)	Name:	StackedShelf	Value:	1
#3 (old)	Name:	BadsideDrawer	Value:	2
#4 (old)	Name:	LowShelf	Value:	3
#5 (old)	Name:	TVSet	Value:	4
#6 (old)	Name:	Picture	Value:	10
#7 (old)	Name:	WallClock	Value:	11
#8 (old)	Name:	Lamp	Value:	12
#9 (old)	Name:	Lamp2	Value:	13
#10 (old)	Name:	WallLamp	Value:	14
#11 (old)	Name:	Door	Value:	15
#12 (old)	Name:	Pano	Value:	16
#13 (old)	Name:	Window	Value:	17
#14 (old)	Name:	CupPyramid	Value:	20
#15 (old)	Name:	RoadBlock	Value:	21
#16 (old)	Name:	ShelfWithBooks	Value:	30
#17 (old)	Name:	ShelfWithBooks2	Value:	31
#18 (old)	Name:	ShelfWithBooks3	Value:	32
#19 (old)	Name:	FinishTape	Value:	40
#20 (new)	Name:	ExampleElement	Value:	41

6. Press the "Save" button.
7. Open the prefab that you want to add and add the "Savable Item" component. Inside the "Savable Item" component select the appropriate enum value.



ExampleElement

Tag: Untagged | Layer: Default

Transform

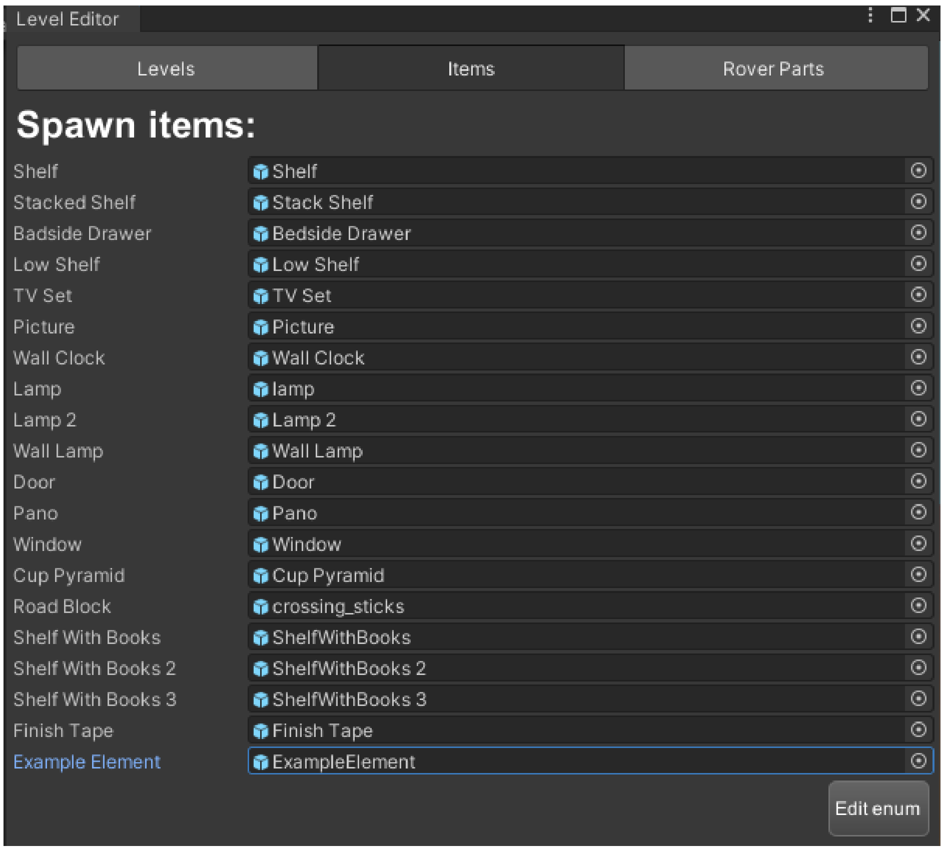
	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Savable Item (Script)

Script: SavableItem

Item: Example Element

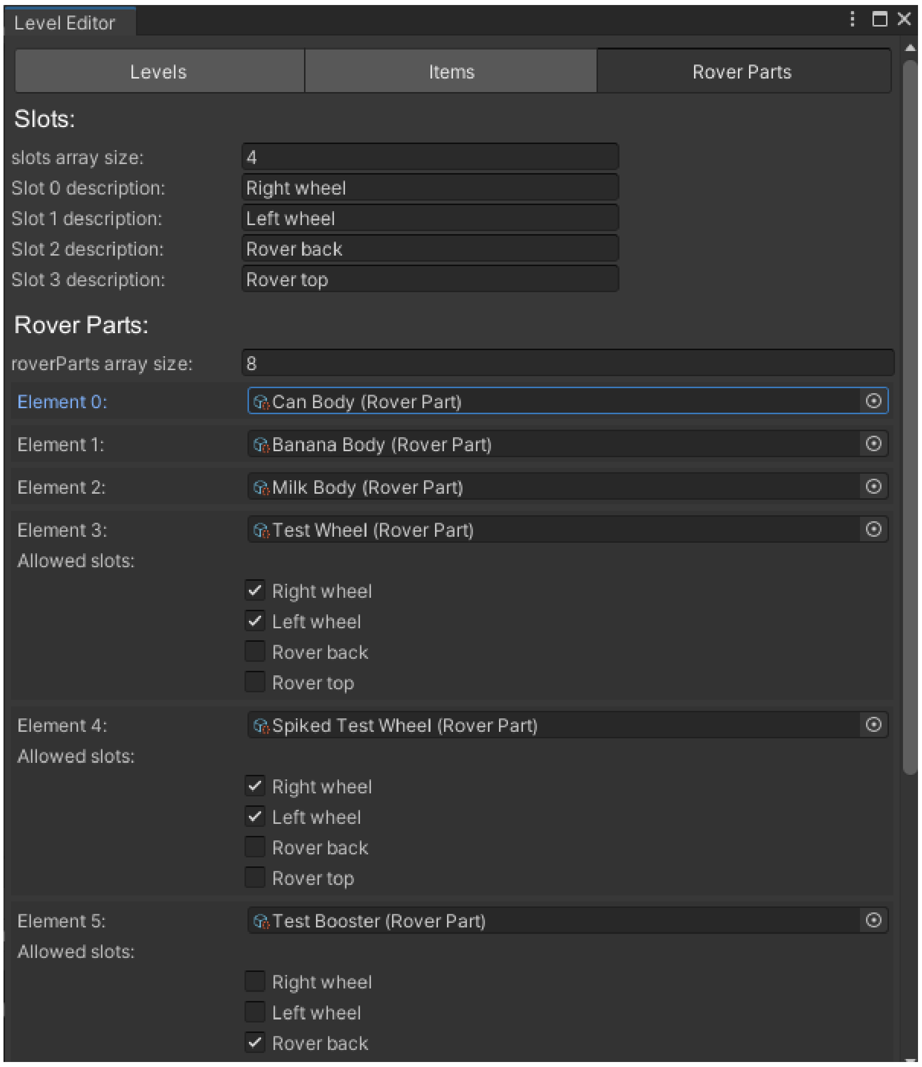
8. Now you can drag your prefab to the appropriate field inside the "Items" tab.



Spawn items:

Shelf	Shelf
Stacked Shelf	Stack Shelf
Badside Drawer	Bedside Drawer
Low Shelf	Low Shelf
TV Set	TV Set
Picture	Picture
Wall Clock	Wall Clock
Lamp	lamp
Lamp 2	Lamp 2
Wall Lamp	Wall Lamp
Door	Door
Pano	Pano
Window	Window
Cup Pyramid	Cup Pyramid
Road Block	crossing_sticks
Shelf With Books	ShelfWithBooks
Shelf With Books 2	ShelfWithBooks 2
Shelf With Books 3	ShelfWithBooks 3
Finish Tape	Finish Tape
Example Element	ExampleElement

Rover Parts Tab


Level Editor

Levels | Items | Rover Parts

Slots:

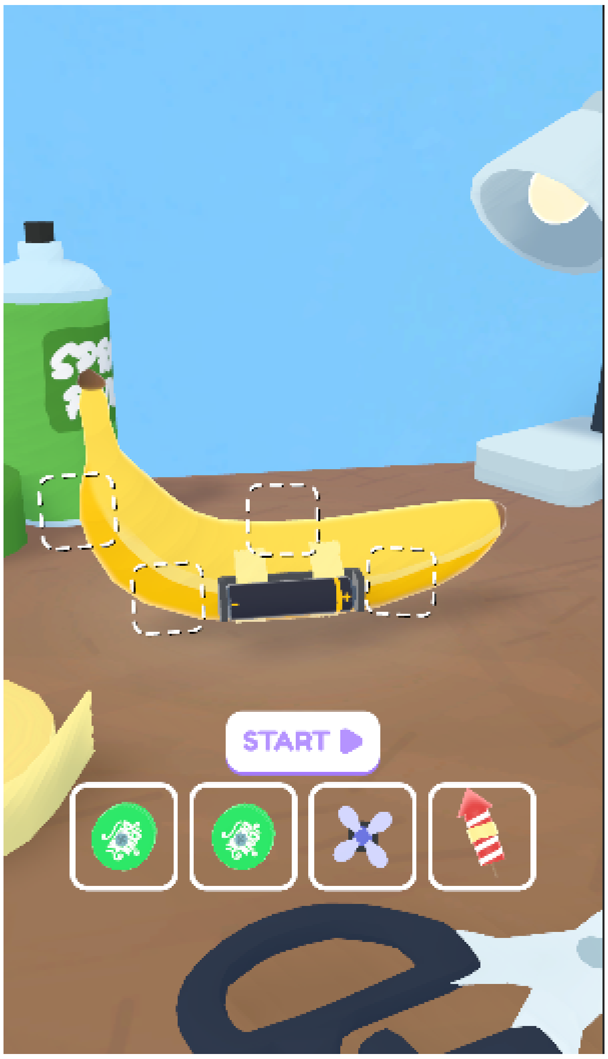
slots array size:	4
Slot 0 description:	Right wheel
Slot 1 description:	Left wheel
Slot 2 description:	Rover back
Slot 3 description:	Rover top

Rover Parts:

roverParts array size:	8
Element 0:	Can Body (Rover Part)
Element 1:	Banana Body (Rover Part)
Element 2:	Milk Body (Rover Part)
Element 3:	Test Wheel (Rover Part)
Allowed slots:	✓ Right wheel ✓ Left wheel ☐ Rover back ☐ Rover top
Element 4:	Spiked Test Wheel (Rover Part)
Allowed slots:	✓ Right wheel ✓ Left wheel ☐ Rover back ☐ Rover top
Element 5:	Test Booster (Rover Part)
Allowed slots:	☐ Right wheel ☐ Left wheel ✓ Rover back

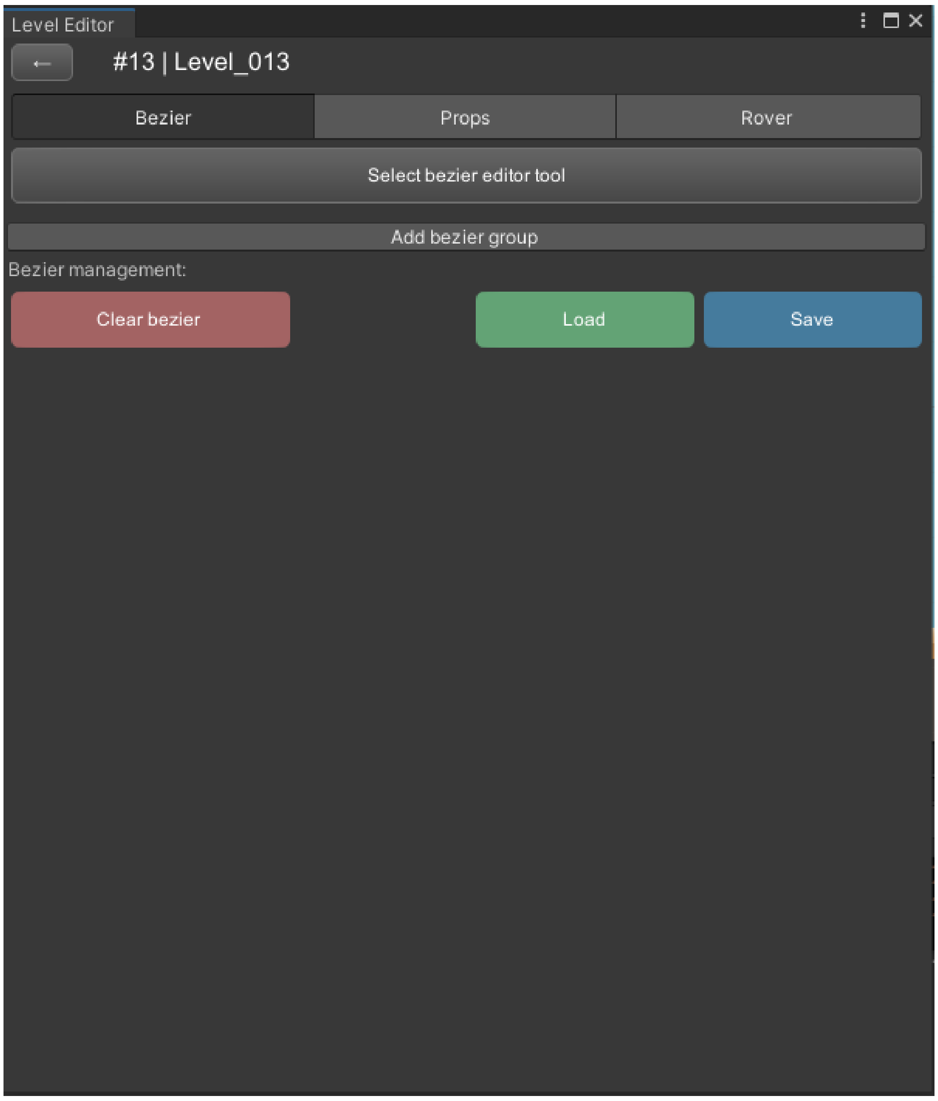
- The first section named "Slots" contains the variable "slots array size" responsible for the maximum number of configured slots. Each slot has a description that is only used in the editor for user convenience.
- The second section named "Rover parts" contains all available rover parts and suitable slots for them (If the rover part`s type is a body then it doesn`t need suitable slots). To add rover part increment variable "roverParts array size", drag your roverPart to the appropriate field, and check suitable slots.

To understand what a slot is look at this picture below. Here we have 4 rover parts that can be connected to the rover body in 4 different slots.

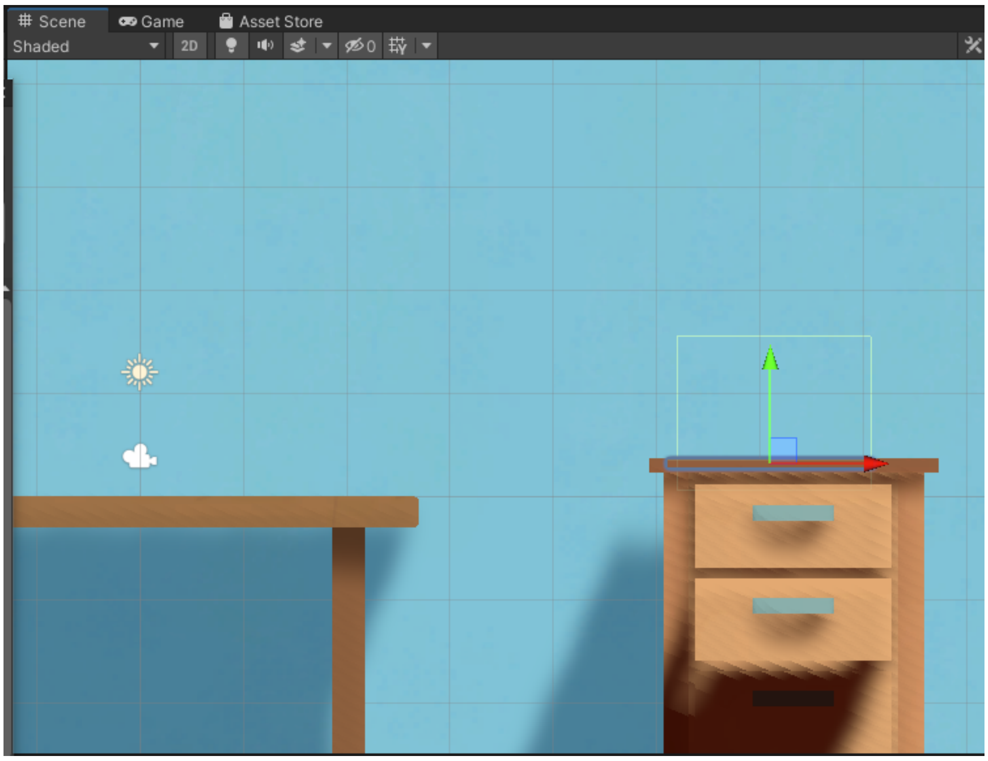


How to add level

1. Go to the **Levels** tab.
2. Press the "+" button to create a new level and automatically select it.
3. Press the **Edit level** button. Now your editor should look something like this:



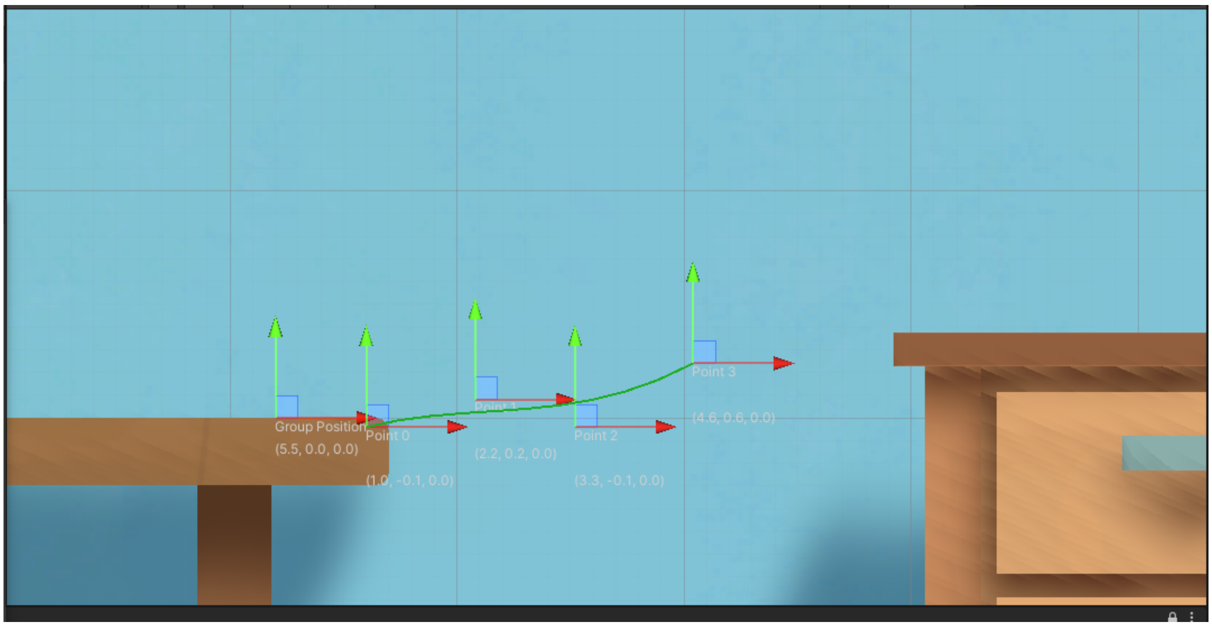
4. Go to the **Props** tab.
5. Press on the **Bedside drawer** on another object with a collider to spawn it on the level editor scene. This object will be our destination.
6. Toggle 2d view on level editor scene.
7. Now inside the level editor scene move the object to the right. This object will be our destination. And the work table already present is our starting point.
8. Press **Select finish gameobject** inside the editor and drag it inside our finish object. The result should look like this:



9. Press **Save**.
10. Go to the **Bezier** tab.
11. Press **Add bezier group**.
12. Press **Add new slice**.
13. Press **Select bezier editor tool**.

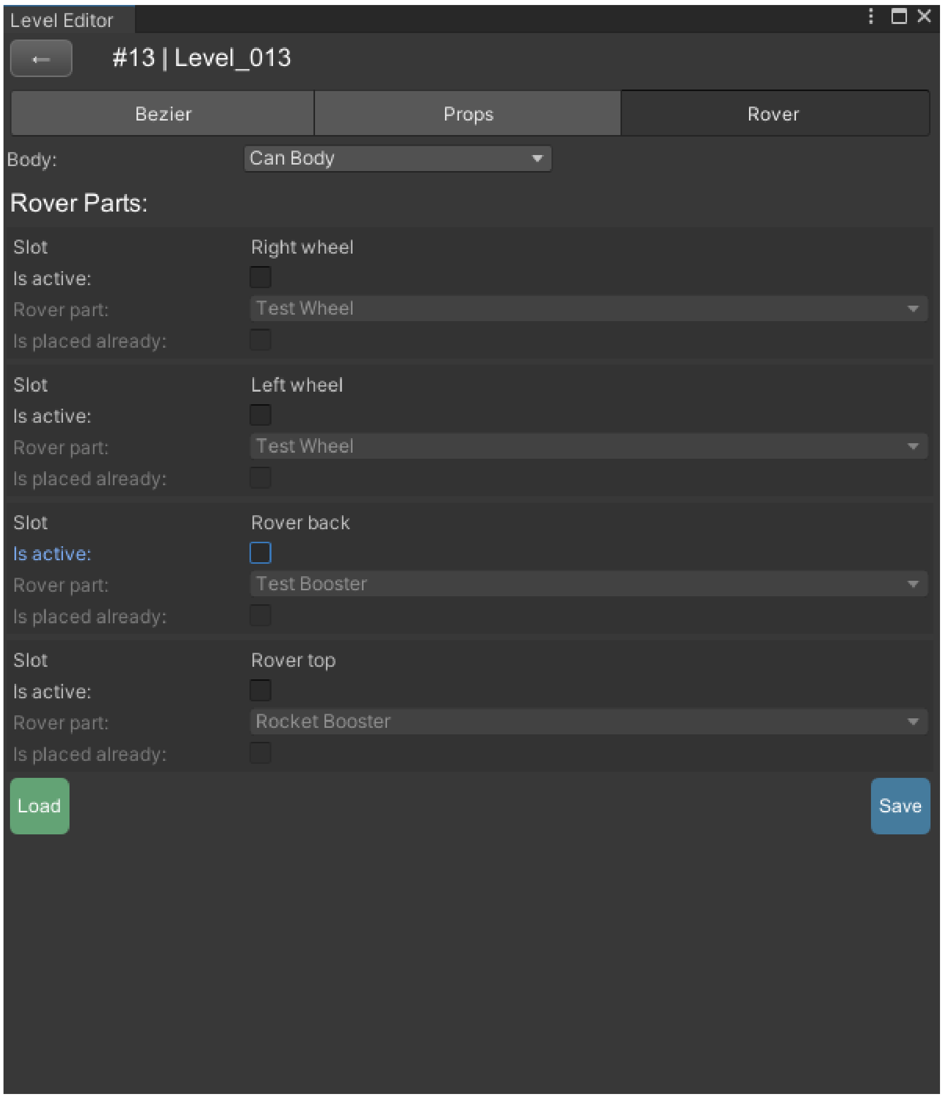
Note: Bezier group contains slices. And each slice is a line that is configured by 4 points. Each slice can also contain coins automatically distributed on the surface. To place or remove click the **Coins Toggle** button next to a slice.

14. Now you should be able to edit the slice. Which looks like this:



Note: If you can`t then check if the gizmos option is enabled in SceneView. If you accidentally selected another object on the scene you can always click "Select bezier editor tool" again.

15. Add as many slices as you need to build a road from start to finish.
16. Because our bezier group is connected to the object at the start you need to enable the **startTape** variable to spawn a tape in place of connection during the game.
17. Following the same logic as the previous step normally you would also enable the **endTape** variable but not in this case. Our object is also a finish and therefore we need to use **finishTape**.
18. Press **Save**.
19. Go to **Props** tab.
20. Press on **Finish tape** and drag it to the point where the road ended.
21. Press **Save**.
22. Go to **Rover** tab.
23. Select body and parts. You need 2 wheels as the bare minimum. Also having a buster on the back is a good idea in case your road is too steep. So you will have something like this:



#13 | Level_013

Body: Can Body

Rover Parts:

Slot	Right wheel
Is active:	☐
Rover part:	Test Wheel
Is placed already:	☐
Slot	Left wheel
Is active:	☐
Rover part:	Test Wheel
Is placed already:	☐
Slot	Rover back
Is active:	☐
Rover part:	Test Booster
Is placed already:	☐
Slot	Rover top
Is active:	☐
Rover part:	Rocket Booster
Is placed already:	☐

Feel free to experiment with different variations.

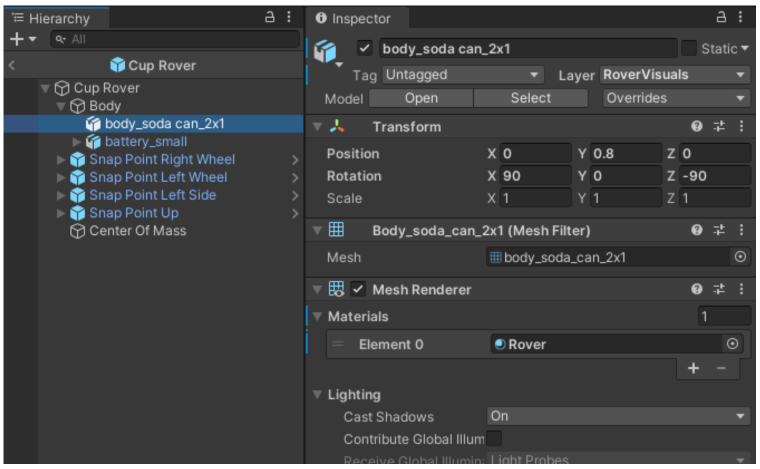
24. Press **Save**. And now we have a prototype of our level.
25. Press "←" to return to the **Levels** tab.
26. Press **Set this level as current level**.
27. Press **Open Game scene**.
28. Now you can close the editor and launch the game to test your level.
29. Edit the route and test it until you are satisfied.
30. Add objects from the **Props** tab to decorate your level.

Other features worth mentioning:

- If you move levels around in the levels list "Rename" levels button lets you quickly rename all your levels according to their position.
- If you have enough levels to enable pagination and need to move levels between pages you can right-click on an element in the levels list and select **Set position**.

How to add a new vehicle

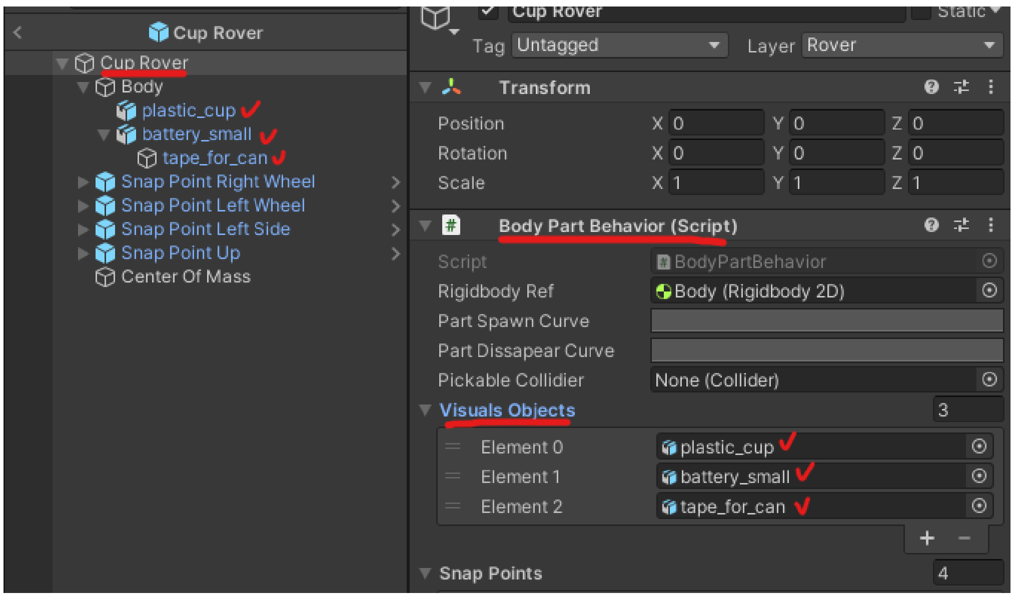
1. Select **Can Rover** prefab located at Game/Prefabs/Rover/Bodies
2. Press **Ctrl+D** (Comand + D) to duplicate the prefab, name it as you wish.
3. Double click to open prefab
4. You'll find the **body_soda_can_2x1** object as a child of the **Body** object you need to replace it with a new model.
   Drag the new model near the old one, adjust position, scale, and rotation so they have an approximately similar size. Delete old model.



```
Hierarchy
Cup Rover
  Cup Rover
    Body
      body_soda can_2x1
      battery_small
    Snap Point Right Wheel
    Snap Point Left Wheel
    Snap Point Left Side
    Snap Point Up
    Center Of Mass
```

```
Inspector
✓ body_soda can_2x1                      Static
Tag Untagged            Layer RoverVisuals
Model  Open  Select  Overrides
Transform
Position   X 0    Y 0.8   Z 0
Rotation   X 90   Y 0     Z -90
Scale      X 1    Y 1     Z 1
Body_soda_can_2x1 (Mesh Filter)
Mesh       body_soda_can_2x1
✓ Mesh Renderer
Materials                                1
  Element 0   Rover
Lighting
  Cast Shadows   On
  Contribute Global Illum
  Receive Global Illumin: Light Probes
```

5. I'm going to use the plastic_cup model located at the project. You can leave **battary_small** object as additional visual detail or remove it.
6. Select prefab origin object. Find Body Part Behaviour script in the inspector. Unfold Visuals Objects variable and drag all models you've just added.


Cup Rover
Cup Rover
Body
plastic_cup
battery_small
tape_for_can
Snap Point Right Wheel
Snap Point Left Wheel
Snap Point Left Side
Snap Point Up
Center Of Mass

Cup Rover
Tag Untagged
Layer Rover

Transform

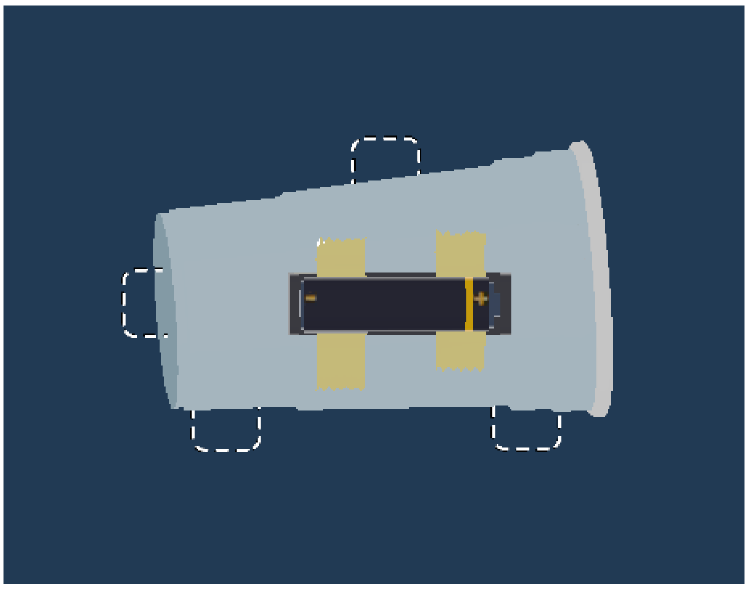
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Body Part Behavior (Script)

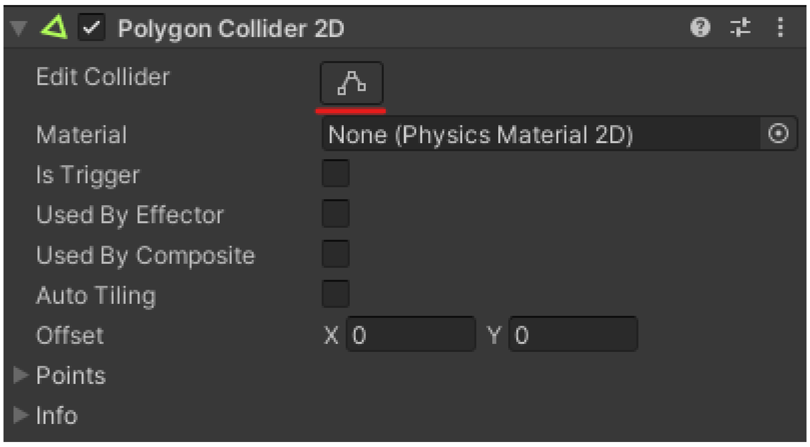
Script	BodyPartBehavior
Rigidbody Ref	Body (Rigidbody 2D)
Part Spawn Curve	
Part Dissapear Curve	
Pickable Collidier	None (Collider)
Visuals Objects	3
Element 0	plastic_cup
Element 1	battery_small
Element 2	tape_for_can
Snap Points	4

You can add additional slots or remove empty using “+” and “-” buttons. Make sure all visual objects are referenced.

7. Adjust 4 **Snap Point** objects position if required.

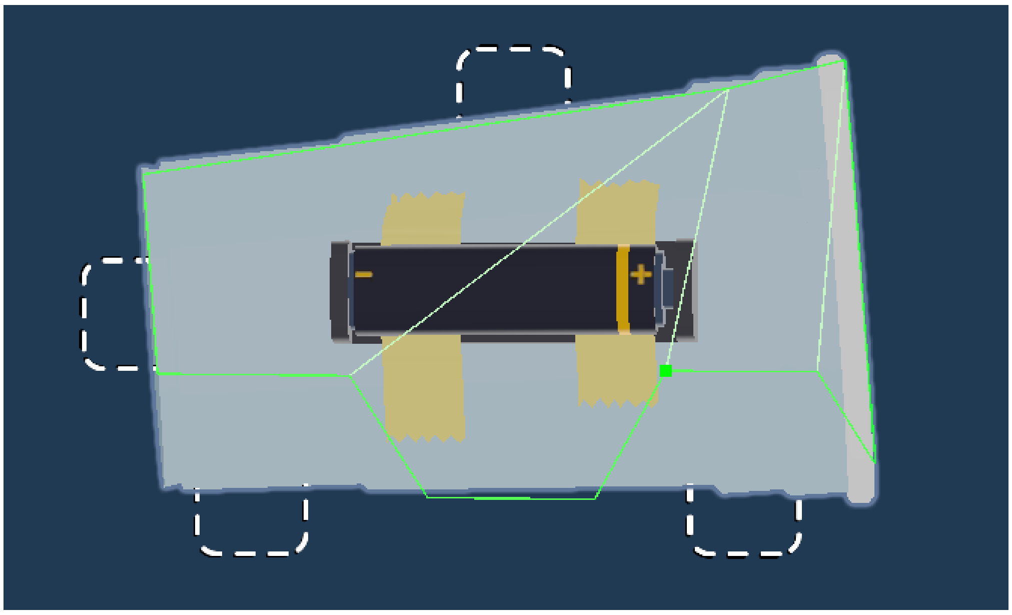


8. Select **Body** object and find **Polygon Collider 2D** script. Press **Edit Collider** button.

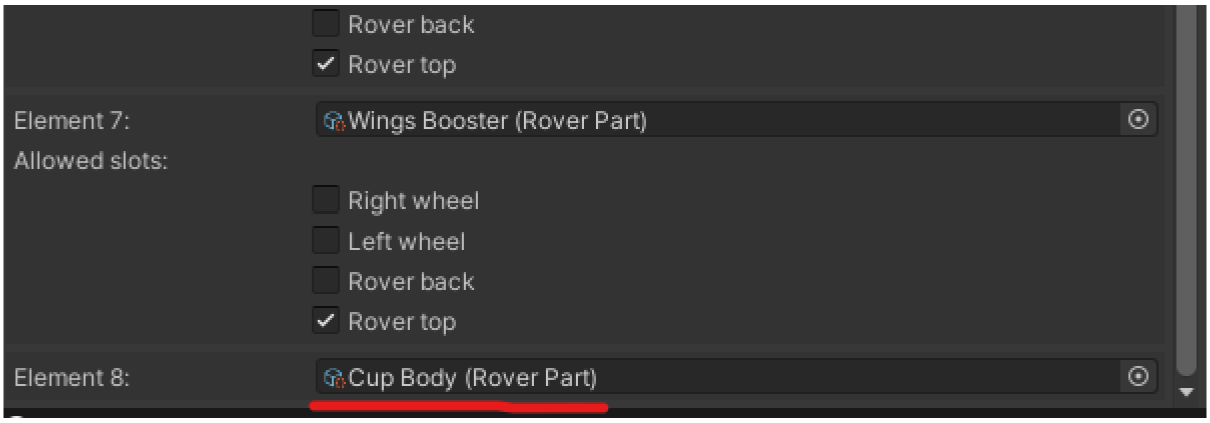

Polygon Collider 2D

Edit Collider	
Material	None (Physics Material 2D)
Is Trigger	☐
Used By Effector	☐
Used By Composite	☐
Auto Tiling	☐
Offset	X 0 Y 0
Points	
Info	

9. Adjust collider shape dragging points. Make sure to leave gaps for wheels. Like at an example:



10. Exit prefab mode.
11. Duplicate **Can Body** asset located at **Content/Levels System/Rover Parts/Bodies**. Name it accordingly.
12. Drag newly created prefab into **Part Object** and **Build Visuals** fields.
13. Open **Level Editor**: Tools - Level Editor
14. Select **Rover Parts** tab.
15. Find field called **rowerParts array size** increase its value by one
16. You'll see the additional element that appeared at the very bottom of the list below.
17. Drag the newly created asset here.

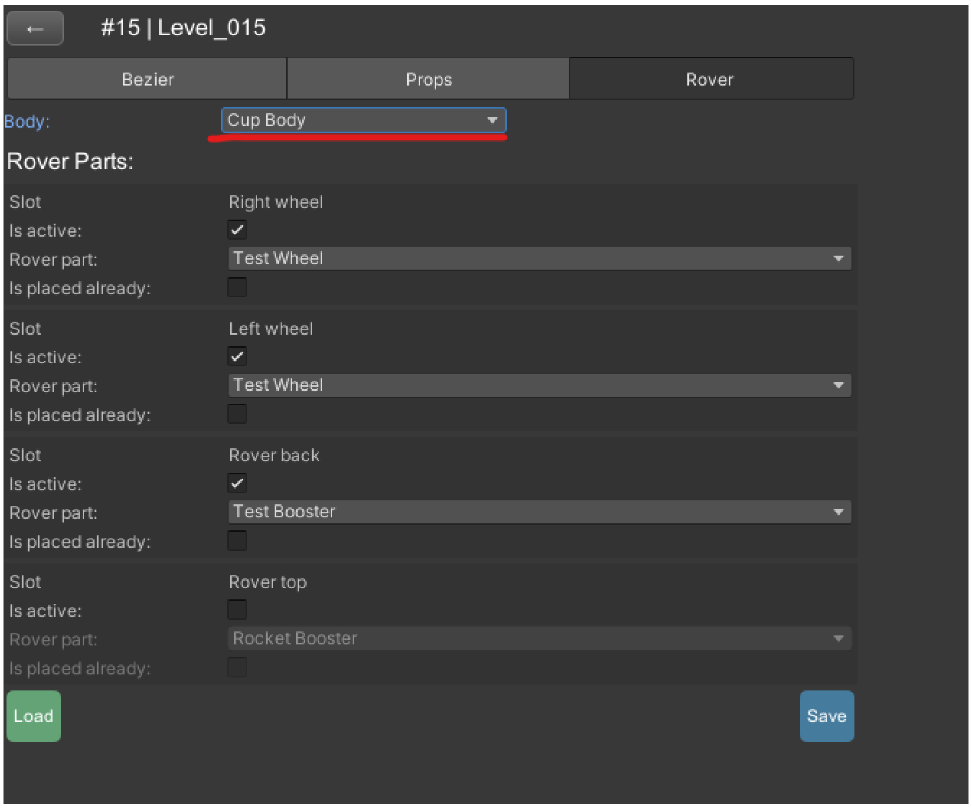


☐ Rover back
✓ Rover top

Element 7: Wings Booster (Rover Part)
Allowed slots:
☐ Right wheel
☐ Left wheel
☐ Rover back
✓ Rover top

Element 8: Cup Body (Rover Part)

18. Now you'll be able to select a new body type while creating or editing levels



#15 | Level_015

Bezier | Props | Rover

Body: Cup Body

Rover Parts:

Slot: Right wheel
Is active: ✓
Rover part: Test Wheel
Is placed already: ☐

Slot: Left wheel
Is active: ✓
Rover part: Test Wheel
Is placed already: ☐

Slot: Rover back
Is active: ✓
Rover part: Test Booster
Is placed already: ☐

Slot: Rover top
Is active: ☐
Rover part: Rocket Booster
Is placed already: ☐

Load | Save