

**Version (See Asset Store):**

All updates will be featured on the Asset store version notes.

Are there any more features planned?

Depending on the kind of support we get from users and if they are more interested in seeing more, then why not 😊

.
. .
. .
. .
. .
. .
. .

Thank you for buying this KIT, please remember to leave a [review](#) so we can improve it more.

Welcome to 2D Platform – Hill Climb Racing Driving KIT Documentation. This will tell you everything you need to know about the KIT.

What is this KIT?

This is a kit which contains models, prefabs, items, player, and settings and scripts both JS and C# which was made for those who are new and want to learn more about creating a 2D platform racing or driving game. (Or any other similar type) This KIT provides you with simple codes, models, animations, menu, textures and everything to start your own game and including 3 example projects along with menu. You can simple drag and drop the prefabs to create your own game or just edit the example scenes itself for your needs easy in just a few minutes.

This pack contains:

- 3 playable players with different faces, cars and gender which can be customized.
- Score point system increase on every score pick up.
- Fuel system where your fuel will decrease and will only increase when you pick up fuel.
- Full playable player with acceleration, reverse and break.
- Game over and game win screen GUIs.
- Score display on game end and game win.
- Animations for your game.
- Sounds effects and music.
- Cloud sprites for your environment.
- 2 type of different environments for your game. (Desert and grass)
- 2 complete ready to use set of environment to just drag and drop in your scene.
- 8 different GUIs for your game and menu.
- 30+ different sprites including objects and hills.
- 2 items which can be picked up (fuel and score).
- And many more.

What can I create with this?

This KIT is mostly used for creating a game such as 2D platform driving or racing game as you need to race to the flag, drive and try not to run out of fuel or something else. The models and prefabs use a style of 2D platform game so it is easy to create your game and most of the prefabs and models are in silly and cartoon genre. The package includes sprites and prefabs to create your own environment easy along with objects. Everything can be changed to your needs. You have 3 playable characters which you can start with including full features. You can create any type of 2D platform driving racing game and even though this is a 2D template kit, you can always edit it to your needs to create other games too.

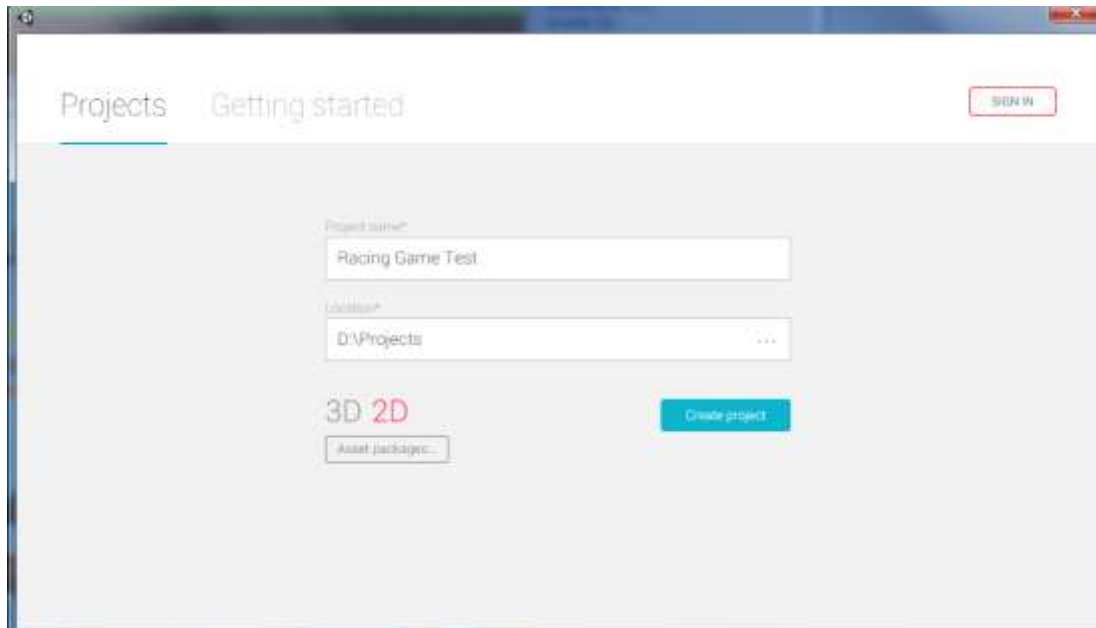
Below will be the step-by-step instructions to start creating your own game easily along with a F.A.Q and setting up; please follow the instructions before starting up the project and if you have any trouble, do not hesitate to contact us or use the scenes provided to learn more. Enjoy!

•
•
•
•
•

Create a new empty project

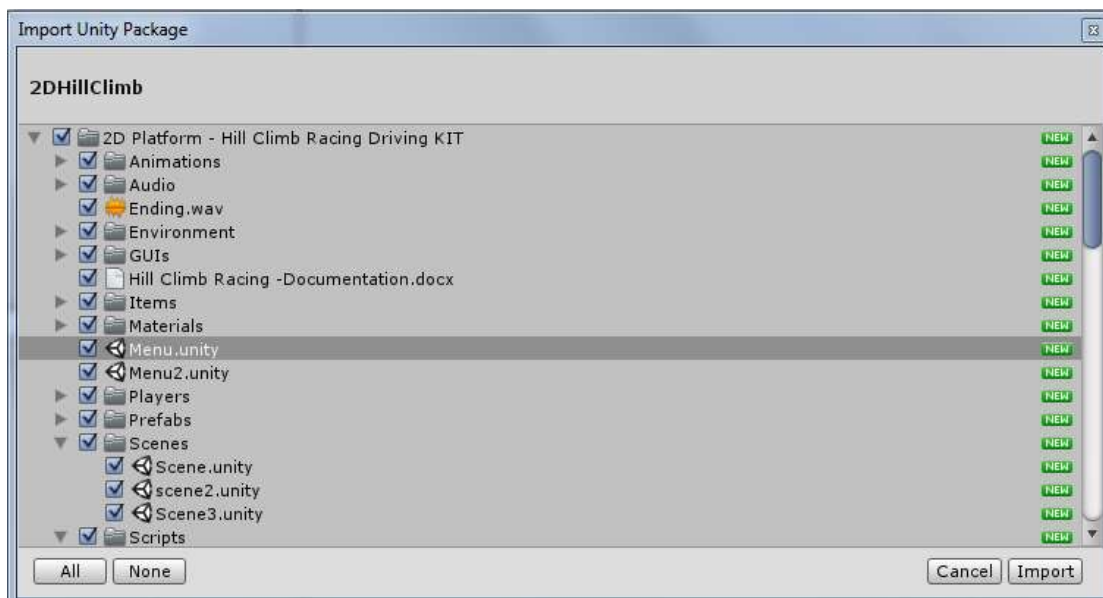
Create a new empty project, set defaults for 2D (3D works too, however 2D works best as 3D may trigger some errors since this is a 2D template) and Import this package into the NEW EMPTY PROJECT.

Please note: Importing into an OLD Project may result in damaging your old project.



Import Package to unity

Import the package inside Unity using the import button.



Explanation of files

Animations folder

This is the folder that contains all necessary animations. I advise not to touch these as they aren't required unless you want to edit the animations.

Audio folder

The audio folder contains all the KIT sounds and music including audio and mostly sounds FX.

Environment folder

Contains the environment as sprites to create your own 2D platform world. Please note that this is sprites and do not come with collider, so you would need to add the collider on the sprites by yourself.

Complete Sets folder

The complete sets folder is the whole environment ready to just drag and drop into your game.

GUIs folder

Contains the GUIs prefabs to use in game such as the game ending GUI, health huds, game win GUI and such.

Items folder

Items that can be picked up in your game. **FuelPickup** can be picked up to refill fuel and **ScorePickup** can be picked up to increase the player score.

Materials folder

The materials folder contains simple material and the 2D material for the tires to bounce. You can edit the Friction and Bounciness to your needs.

Players folder

The prefabs of the players comes in as ready to use and can be just dragged into the scene.

Prefabs folder

Prefabs folder are different then the sprites folder, because prefabs are ready to use game objects and they come with collider on, so each prefab you drag into the scene it will have a collider which the player will hit.

Scenes folder

Example scenes with 3 different levels.

Scripts folder

This folder contains IMPORTANT scripts for your game.

Settings folder

The settings folder contains simple prefabs for your game such as the game camera to follow the player, the win flag so when the player collides next to it you will win the game and the PlaySoundOnTrigger which will switch on a sound trigger when the player collides with this prefab.

Sprites folder

The rest of the sprites mostly of the models and objects, some can be used as sprites themselves however you can see the sprites such as the players, grounds and score etc.

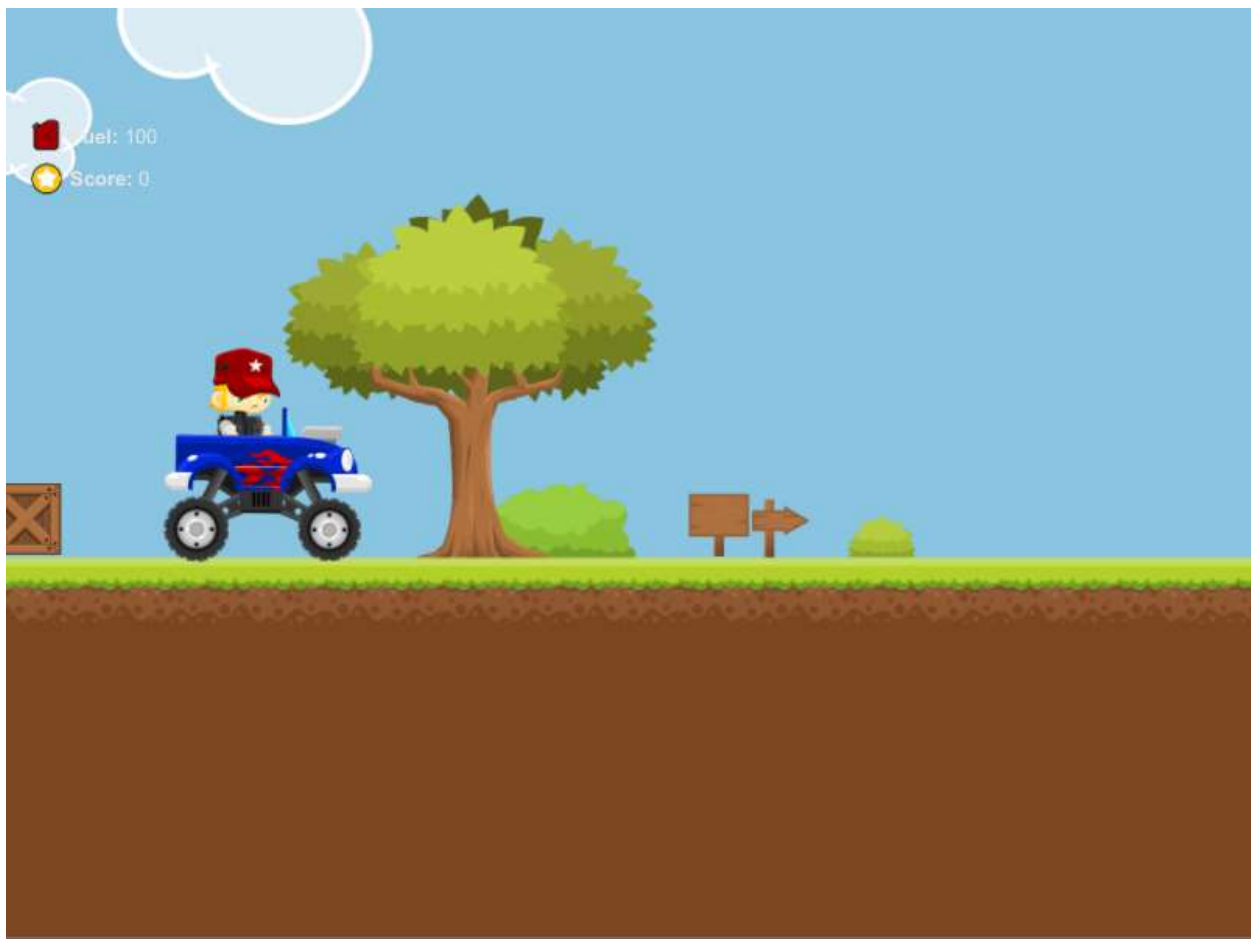
Planning your first game

Now that you know what your folders are about, it's time to start creating your own very first game! So you want to make a simple game racing or driving game that you have to finish and not fall on your body until the finish line, easy.

Start planning your own game and how it's going to be, how easy or hard is it going to be to the finish line? How many times will the fuel decrease from the gas? Start do some little planning and let's go!

Step-by-step ... Let's start! (Don't worry, this example scene is included too so you can also edit it)

Some steps are totally optional and no needed to complete the full game. For example adding score and fuel pickups are all optional, but we will cover it too just like we are creating a full hill climb racing game.



...

Creating your first game

Step 1: Create a new scene, delete the **Main Camera** so your scene will be clean and new.

Step 2: Go to the **SETTINGS** folder, drag and drop **Main Camera** into the scene.

Step 3: Go to the **PLAYERS** folder, drag and drop one of the player prefab into the scene. In this case we put a **Female** player prefab.

Step 4: Click on **Main Camera** in your scene and where it says *target*, drag and drop the **Controller** inside the player prefab, in this case inside the female prefab. (Do not drag and drop the entire player prefabs, otherwise the camera will not move. Instead open the player prefabs children and drag and drop the **Controller** only in the *target* of the main camera.

Step 5: Go to the **PREFABS** folder, click on ground prefabs and drag and drop any ground prefabs into the scene, in this case we put **DesertA** prefab.



If you done everything correctly you can see that the scene is already starting to build up. Play the scene and test it out the player should be able to move with **D**, reverse with **A** and break with the **spacebar** button. In your scene you should have about 3 prefabs; Main Camera, Player and Ground.

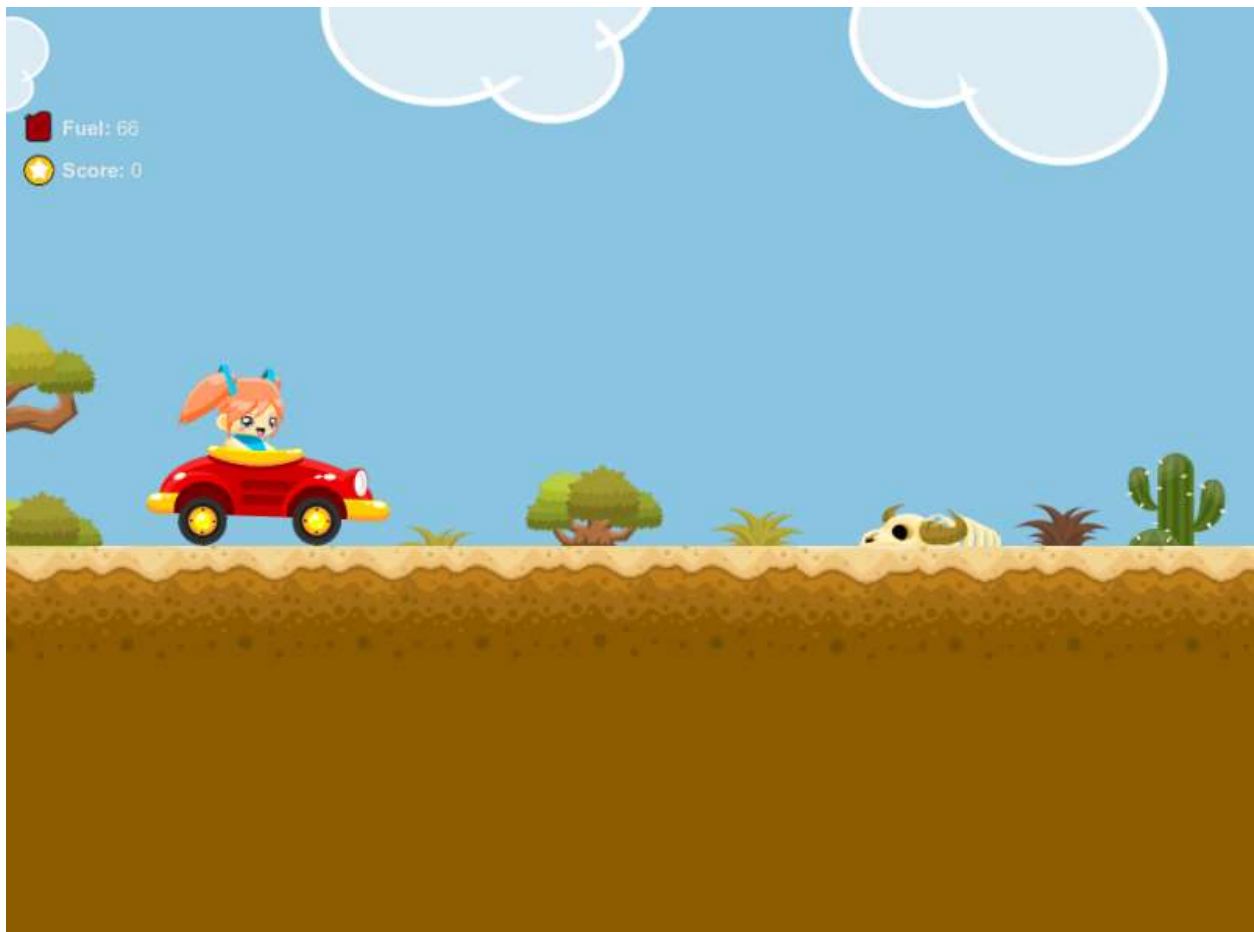
Detailing your first game

If you tested the scene out and everything is working correctly, then good job. As you can see you can move the player and everything is set up by itself. The fuel is decreasing and you need to hurry up to the finish line if you want to win, but we want to add some color into the world and add some scores for the player to pick up and even add some fuel to refill.

Step 1: Go to the **Environment** folder and choose a setting you like, in this case we will choose Rock Desert to stay with the theme. Add some bushes, cactus, grass and objects into the scene to make it feel more alive.

Step 2: Go to the **Environment** folder and click on clouds folder and add some clouds in your scene.

Sometimes when you add sprites they will appear before your player in 2D which will make the sprites on front, to prevent this go into 3D and grab all the sprites you put in and drop them back into the scene so it seems like they are in the background.



As you can see, your game is already done in just five easy steps. Now add some more grounds and obstacles, add some wood and be creative. Play around a little and make it look awesome!

Touching your first game

Once you done adding the grounds and all you need to add some score and maybe even some fuel for the player to pick up. To do this, go to the **Items** folder and drag and drop the *ScorePickup* and *FuelPickup* prefabs into the scene anywhere you want above the ground. Make sure you make them into a level where the player can pick them up, this is so the player can pick up score points to add to the score and fuel so he can refill his tank otherwise he will lose if he run out of fuel!

Step 1: Go to the **Items** folder and drag and drop some score points prefabs into the scene so the player can pick them up to add to his score, this is optional.

Step 1: Go to the **Items** folder and drag and drop the fuel prefab anywhere you want so the player can pick fuel up because if he runs out of fuel, the game ends. This is also optional.

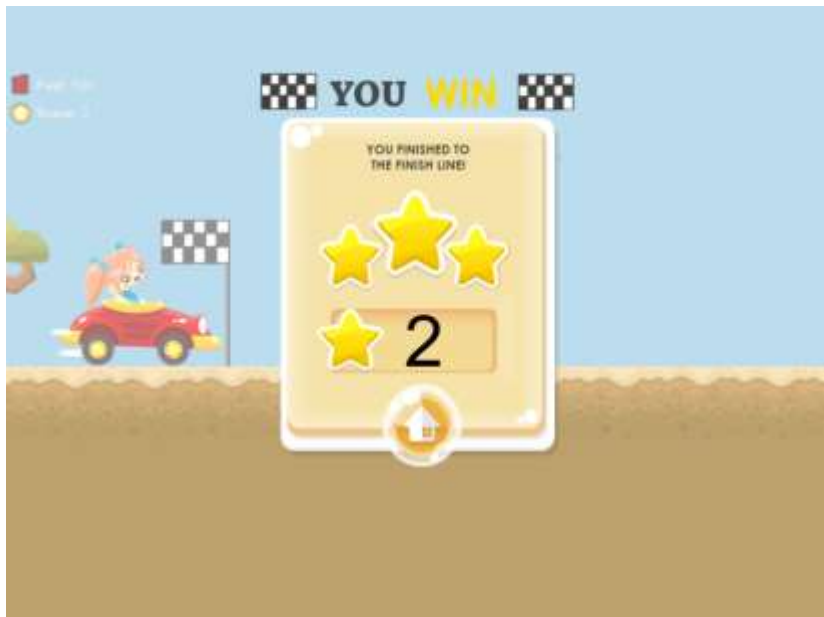


The game is now finished, the only that is left is the finish line to win and end the game and display the win game screen once the player reaches the final destination. Let's add that finish line flag so we can complete the project!

Finishing your first game

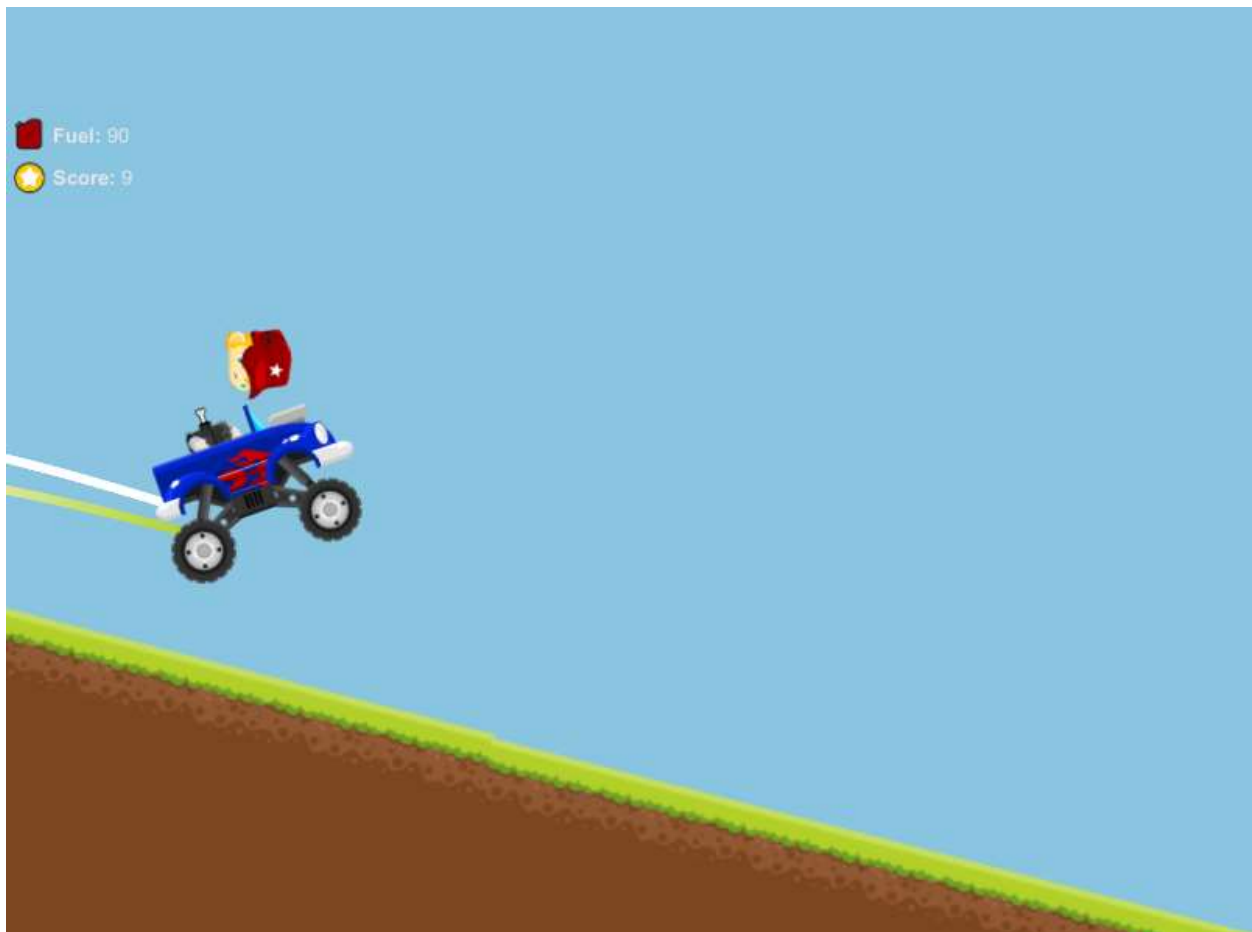
Once you get everything done and you are ready from creating your environment and adding the items, you now want to create a finish line to finish game and display the winner screen. This is mega easy, all you need to do is go to the **Settings** folder, drag and drop the *Win Flag* into your scene. When the player reaches the flag, the game is won and the screen is displayed along with your score. Easy right?

Step 1: Go to the **Settings** folder and drag and drop the *Win Flag* prefab into the scene where you want the game to end once the player reach the flag destination.



Playing your first game

Congratulations! You just created your own very first game. You can control the player with D to accelerate, A to reverse and spacebar to break. You can move on the ground which is with the polygon collider and you have to hurry because if the fuel runs out, you lose and also if you player gets upside down, you lose too. The fuel decreases every 3 seconds and (if you wish to change that scroll down to the F.A.Q to see how) you can pick up score to add score points and fuel to refill. Once you finish the race when colliding with the win flag, the game is won and your score is displayed. f you are in doubt or stuck, be sure to check out the F.A.Q and be free to use the example scenes provided to learn from them. Enjoy!



Try not to lose your head though, remember to take breaks! ☹️

F.A.Q

1. Who is this made for?

This KIT is made for anyone who would like to experiment, learn and create a similar or any other game. This KIT will provide you the scripts you need to create it. Yes, the KIT may be messy at first (The scripts are both in C# and JS) and also they are separated scripts and the reason for this is to be made easier to understand for the new users who are willing to learn more.

2. What is this KIT focused on?

The main reason this KIT was release was for creating 2D Platform driving racing game.

3. Can I add more sprites?

Yes! You can add as many sprites as you like and also create your own grounds, just be sure to add a polygon collider when you are creating your own grounds.

4. How do I set player fuel to decrease faster/slower?

Find the Playerscript.cs located in the scripts folder. Find the line;

```
fuel--;  
fuel--;  
m_Timer = 3.0f;
```

The `fuel--;` is how many times will the fuel decrease and the `m_Timer = 3.0f;` is in how many seconds will the fuel decrease in, so in this case its 2 times ($100 - 2 = 98$) every 3 seconds. You can change that to your needs, for example if you add an extra `fuel--;` and increase the time to `m_Timer = 5.0f;` then it will be 3 times in 5 seconds.

5. How do I remove that the players head fall off?

Go to the player prefab and navigate to the controller and find the last Wheel Joint 2D and check that in the Connected Rigid Body there is the Head as a target. If so, go to Break Force and instead of the default to drop his head on a force change it to Infinity. Your player head will not fall off now.

6. Done creating my game, now what do I have to do to win the game?

Go to the settings folder and drag and drop the Win Flag into the scene. Once the player collides with the prefab, the game will be won and display the game win GUI and score.

7. How do I change fuel pickup and score pickup sound?

Go to the main player prefab (Male, Female, Orc) in the players folder and go to the inspector. There will be the settings of the player under **Player Scripts** which you can change, settings include Fuel Sound and Score Sound, these are the sounds that will play when you pickup the score or fuel. You can change them to your needs. There you can also change the other settings such as the GUIs and HUDs.

8. How can I change the max speed to faster/slower?

Go to the main player prefab (Male, Female, Orc) in the players folder and go to the inspector. There will be the settings of the player under **Car Movement**, there you can change the Max Fwd Speed (Forward speed) and Max Bwd Speed (backwards speed).

9. How do I change the win game sound and the end game sound?

Go to the main player prefab (Male, Female, Orc) in the players folder and go to the inspector. Click on Controller and there are two prefabs called GameEndCollider and GameWinCollider. Click on one of them and on the script you can set which audio to play on End Game Sound or Win Game Sound.

10. How do I change the pause HUD?

The pause GUI can be found in Main Camera, go to Main Camera and you can change the texture on Pause Texture.

11. My camera won't follow the player?

This can happen if the Target of the camera is not selected to follow the player. Go to your scene and click on the Main Camera, there is a script called CameraFollow2D and near Target, drag and drop your player prefab from the scene and into the Target. This will make the camera follow the Target player.

12. Why sometimes my player comes out flying and get stuck on grounds?

This can happen if you have a polygon collider on the sprite grounds and you have not set them up correctly, so the polygon is bouncing out of the ground making the player hit it and get stucky movements. Be sure to take your time with the grounds and make them fit so there are no colliders coming out of the original ground. You can also edit your own sprite ground, put it into the scene and add a polygon collider to make it more smoothly as the ground sprites are separated prefabs.

13. Can I add more to this KIT?

Once you buy it, the KIT is yours to edit. This KIT was actually made as a “learning project” for others to learn on how to create a simple game. Yes, you can modify the code and add your own to improve this KIT for your project.

14. Can I change the HUDs of the game?

Go to the player settings (as instructed in question 7) and in the inspector there will be the settings of the HUDs for health, ammo and score. There you can change which GUI you want too.

15. I have my own ground sprite, how can I use that?

To use your own ground sprite, you have to import it into Unity and then drag it into the scene. Be sure to add a 2D polygon collider to the sprite and you are good to go.

If you have any trouble, you can always edit the example scenes in the project.

P.S: Always create an empty project and NEVER import it into an old project because it might (and will) damage your old project!

P.S.S: Note, this KIT will be updated so please see the version notes in the asset store for a more updated version of the KIT. Also, if you like please do leave a review so we can improve!

Happy game making!