

**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, we really appreciate your support.

Welcome to 2D Platformer – Run and Jump’ Abstract Jumper KIT Documentation. This will tell you everything you need to know about the KIT and how to start building your own game easy.

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 ‘run and jump’ game or something similar. This KIT provides you with simple codes, models, animations, menu, textures and everything to start your own game and including 2 example projects along with a 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:

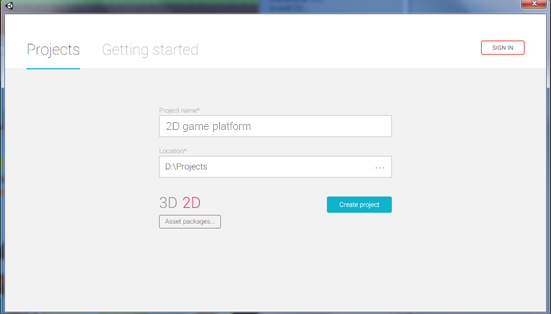
* 2 playable players with different sprites, color and animations which can be customized.
* Fully function player run, double jump, collect pickups, kill and get killed with animations.
* Score point system increase on every score pick up.
* Game over and game win screen GUIs which can be customized.
* Score display on game end and game win.
* Player and enemies animations for your game.
* Beautiful sounds effects and music.
* Loads of beautiful abstract style sprite and graphic packs by awesome **Kenney.**
* Ready to use prefabs to just drag and drop.
* Ready to use graphics sprites with collider ready.
* Different GUIs for your game and menu.
* Loads of different sprites including platforms, players, hills and more.
* Items which can be picked up (such as crystal scores).
* GUI speech bubble display to add simple text hints.
* Movable platforms and boxes.
* 2D Physics and rigibody.
* Locked doors with keys.
* Platforms that makes you jump higher.
* And many more.

What can I create with this?

This KIT is mostly used for creating a game such as 2D platform ‘run and jump’ games as you need to reach the end to win a game without getting killed, or something else. The models and prefabs use a style of 2D abstract platform game so it is easy to create your game and most of the prefabs are ready to use and you can also build your own easy with the sprites. The package includes sprites and prefabs to create your own environment easy along with objects. Everything can be changed to your needs. You have 2playable characters which you can start with including full features. You can create any type of 2D platform game and even though this is a 2D template kit, you can always edit it to your needs to create other games too.

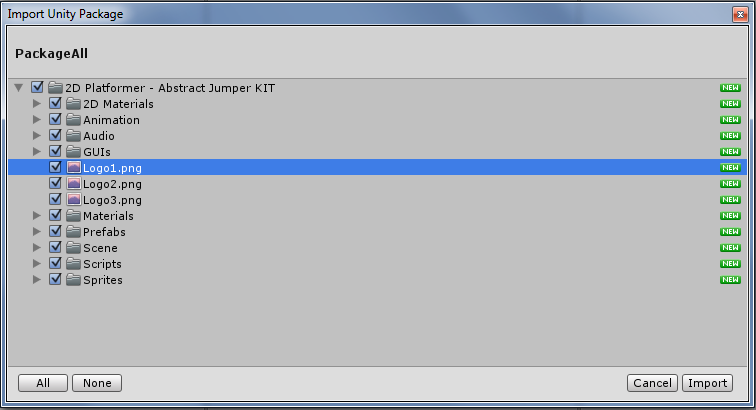
Create a new empty project

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

**2D Materials folder**

This is the folder that contains the 2D materials for jumping bounce on the jumper prefab, player and etc. You can edit the Frication and Bounciness to your needs.

**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.

**GUIs folder**

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

**Materials folder**

The materials folder contains simple material.

**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 or/and can walk on.

**Complete Tiles folder**

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

**Enemies’ folder**

Enemies for your game all comes ready to use and just to drag and drop. There are 3 types of enemies, two which can’t die and 1 which can die by being jumped on.

**Players’ folder**

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

**Settings folder**

The settings folder contains simple prefabs for your game such as the game camera to follow the player (which is not required if you drag and drop the player in the scene, but the camera is just in case), the win flag so when the player collides next to it you will win the game and the Collider to end the game when the player jumps off the scene or you want somewhere to end the game when the player collides with it.

**Scenes folder**

Example scenes with 2 different levels and the main menu.

**Scripts folder**

This folder contains IMPORTANT scripts for your game.

**Sprites folder**

The rest of the sprites mostly of the platforms and backgrounds, some can be used as sprites themselves (such as flowers) however you can see the sprites such as the players, items, platforms and environment etc

Planning your first game

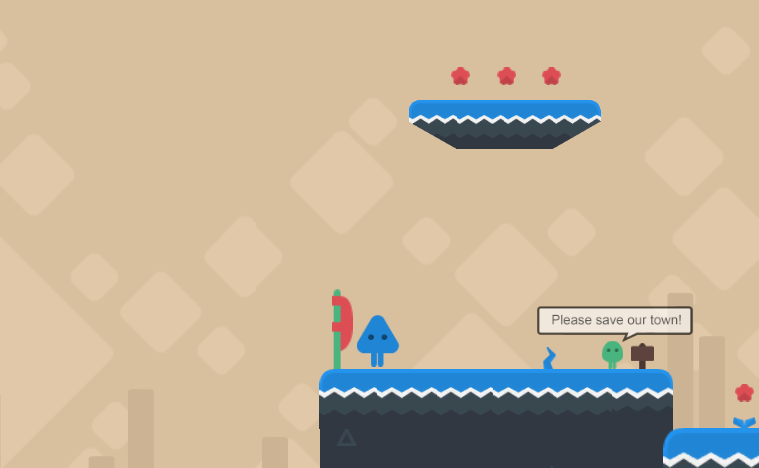
Now that you know what your folders are about, it’s time to start creating your own very first game!

So you want to make a simple run and jump game that you have to finish without getting killed.

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 enemies will there be? 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 enemies are all optional, but we will cover it too just like we are creating a full run and jump game.



.

.

.

.

.

Creating your first game

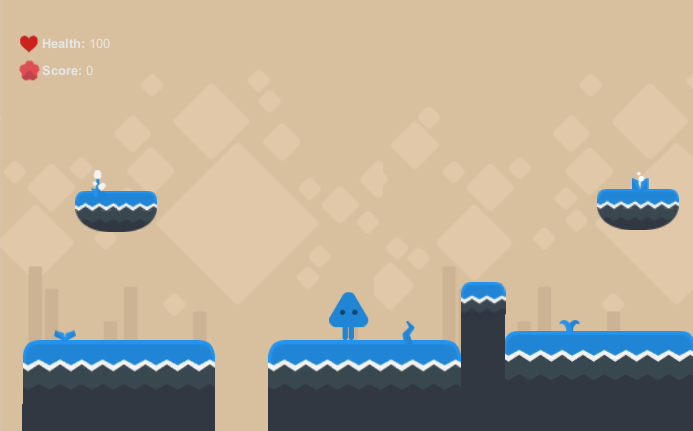
**Step 1**: Create a new scene, delete the **Main Camera** so your scene will be clean and new.

**Step 2**: Go to the **prefabs and to the players** folder, drag and drop **a player** into the scene.

**Step 3**: Go to the **prefabs again and go to complete tiles** folder, drag and drop **some platforms into your scene** and create an environment. You may need to scale the platforms a bit too.

**Step 4:** Basically you already have a game scene with everything ready into it. If you play the scene you can see that your camera will follow the player and the player can move and jump. Now you need to add some stuff into your scene. Go to the **sprites folder** and add some sprites into your scene. In this case, let’s add some plants!

**Step 5:** Once that is done we should add some background to make it better. Go to the **sprites folder** and go to **backgrounds** and add a background. Play around a little and make it look good. You may need to put the scene into 3D and move the background back so it don’t appear in front of your player and platforms. Backgrounds should always be on the back.

**

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 **A** and **D or arrow keys** and jump with the **spacebar** button. In your scene you should have about 3 prefabs; Main Camera, Player and Platforms. (Apart from the sprites which are not prefabs)

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. You can just complete the game by adding more platforms so the player can jump and making it harder etc. Now, let’s add some score points so the player can pick up score points!*

**Step 1**: Go to the **prefabs** folder and drag and drop the crystal prefab which is the score that the player will pick up and each time that is picked up, the player score will increase by +1 point score. Put some into the scene so the player can pick them up!

**Step 2**: Just to make the scene fully, let’s add a jumper so the player can jump high platforms by jumping on them. In the same folder, drag and drop a jumper prefab and put it wherever you want. (It should be near a platform where a player is too high to jump to)

Sometimes you need to make the scene cooler by making the platform sprites match extectly, as you can see this is the scene below that you have, which is nice but to make it better you may want to go to the last sprite which is under the jumper and change the sprite rendered to a box tile instead so it fits the scene better. You can also add your own prefabs by adding sprites and a collider to them.

**

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

Touching your first game

Once you done adding the grounds and everything we need to add some threats, spikes and even enemies to make the game more intense. Then all is left is a flag to win the game and that is it!

**Step 1**: Go to the **prefabs** folder and drag and drop an enemy from the enemies folder into the scene so we can make the game more challenging. There are 2 different type of enemies. These are; enemies which can die by being jumped on and enemies which cannot die so you will need to evade them.

**Step 2**: Go to the **prefabs** folder and add some spikes into the scene too.

**Step 3**: Go to the **prefabs** folder and in the settings folder drag and drop a Win Game Flag and put it anywhere in the scene where you want the player to get to and win the game.

**Step 4**: As optional, you can go to the **prefabs** folder and drag and drop a Locked Door and a Key Pickup and put it anywhere in the scene where you want the player to pick up the key and unlock the door to get where he wants to go.

**

Remember that you can add as many challenging stuff as you want such as enemies and more. You can also animate the enemies so they can move which we will cover in the next page. Also add some spikes and you can fill the scene by adding some locked doors, more score points, challenging platforms, key pickups, movable boxes and many more.

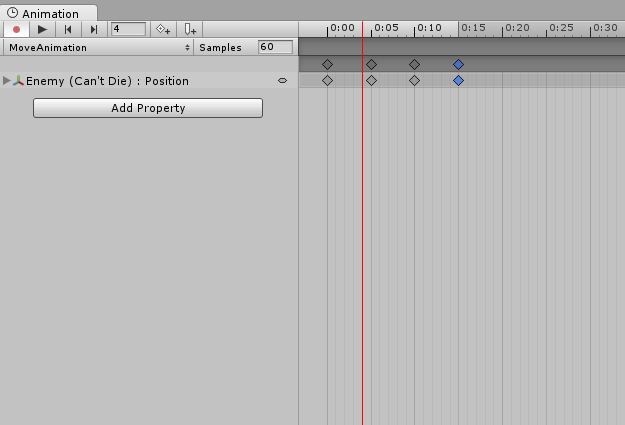
Animated enemies

Once you get everything done and you are ready from creating your environment and adding the items, you are now done creating your game. You may want to add some animated enemies into the scene, and this is super easy to do however you need to play around a little till you get it right. For this, we will be creating a new scene to show you off.

**Step 1**: First off, drop the enemy which you want to animate and move. Click on the enemy and go to windows tab and click on animation, then create a new animation and call it something like enemyanimation1 or anything you like.

**Step 2**: Now you will have the animation window, so go to 0:05 on the red marker and move the enemy slightly in the scene only for a short distance, you need to play around a little to get this right but just move the enemy and then leave the enemy and press on the red marker again and move it to 0:10 and do this again, and on the last dot place the enemy where it started off. Just like in the screenshot in the animation window. Each ‘dot’ is the enemy movement. Be sure to set the samples from 60 to about 20 or less for your enemy to move slowly. Play around a little and you will get it right and will have a moving enemy in no time!

**



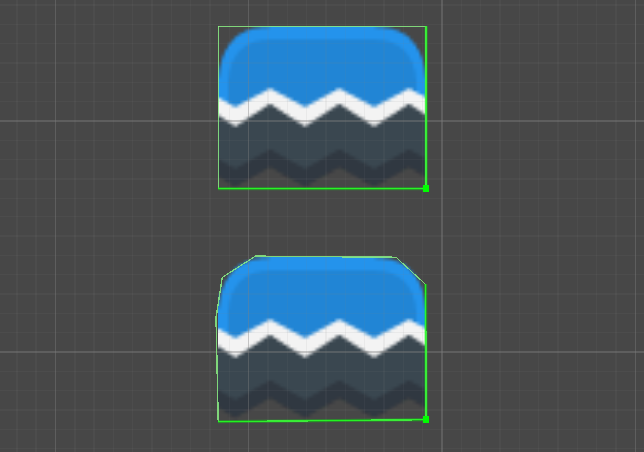
Creating your prefab platforms from sprites

You may need to create your own sprites with colliders and we will guide you through on how to add a collider from a simple in your game. This is very important part on creating the platform for your game along with the collider so the player can move on. Let’s start step by step.

**Step 1**: First off, go to the **sprites** folder and navigate to tiles and drag and drop a sprite into the scene. In this case we will drop a tile which have round edges to edit its collider. A sprite is a simple image which includes nothing but just the image in the scene, while a prefab is a complete game object where it contains collider and such (where the player can move on it, so for example if you put a sprite into the scene and a player on to it the player will fall since it have no collider. If instead you put a prefab the player will not fall since it will have a collider) etc.

**Step 2**: Click on the sprite and go to component > 2D physics and add a polygon collider. Sometimes the polygon collider detects the sprite however in some cases it don’t and if this is the case, you have to edit the collider yourself.

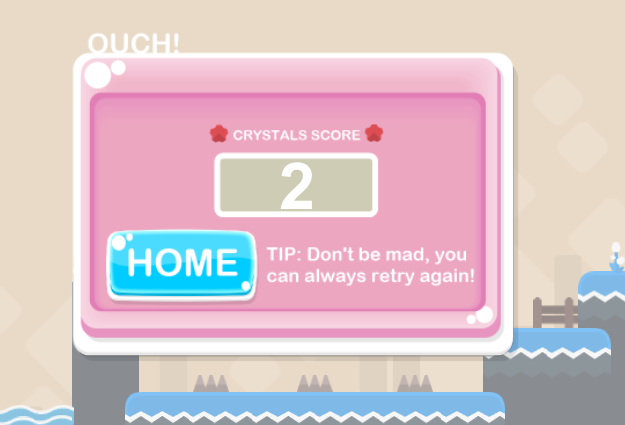
**Step 3**: Now go to the inspector and in the Polygon Collider 2D press on edit collider. Play around with the collider to make it like something below. You can add as much sprites with colliders as you want but be sure to make them all fit and that a collider is not out of its place from a sprite otherwise your player with get stuck in some sprites. If you add two sprite platforms next to each other, do not add colliders on both of them instead put only one polygon collider and edit it. Do not add one collider on each one.



Playing your first game

Congratulations! You just created your own very first game. You can control the player with arrow keys or A and D to move and jump with spacebar button. You can move on the ground which is with the polygon collider. You can pick up score to add score points you need to not to let the enemy near you or touch any spikes and try to finish by getting to the flag. Once you finish to the flag when colliding, the game is won and your score is displayed.

If 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!

**

# Be sure to check the F.A.Q below and have fun! ☺

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 run and jump games.

**3. Can I add more sprites?**

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 and use the ‘edit collider’ feature to edit the colliders exactly for maximum exposure, otherwise with no 2D collider your player will fall of ground.

**4. How do I change the background music?**

To change the background music to something else, go to the player and go to sprite and open its children. There will be a background music under trial where you can change the music to anything you want.

**5. How do I make the game end when the player jumps off screen or off the ground?**

Go to the prefabs folder and drag and drop the FallDownCollider into the scene. This will active the game end once your player touches it, FallDownCollider, traps and enemies which will damage and kill the player are all tagged with with ‘**trap**’. (Which is the tag to decrease the player health and end the game)

**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 WinGameFlag into the scene. Once the player collides with the prefab, the game will be won and display the game win GUI and score. The win game flag prefab can be customized to whatever you want, just be sure to tag it with the tag ‘**win**’. (Which is the tag to win the game once the player touches it)

**7. How do I change hurt sound and score pickup sound?**

Go to the main player prefab in the player’s folder and go to the inspector. There will be the settings of the player under **Player Scripts** which you can change, settings include Hurt Sound and Score Sound, and these are the sounds that will play when you pick up 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 player jump features to ‘only jump when on ground’?**

The player uses a jump cooldown script which will make him able to double jump only after 0.7 seconds from the last jump (which is recommended to use) but we think for everybody since some awesome people don’t like this, we also created a script which will make the player only able to double jump IF the player is touching the ground. However, in this case you have to tag all the GROUNDS with collider with ‘**ground**’ tag and jump replace the JumpCooldown.cs script in the Player to DoubleJump.cs Script.

**9. How can I change the player jump features to just one jump instead of a double jump?**

Simple. Go to the player inspector and replace the Jump Cooldown script to Jump Cooldown Once script. This will make the player only jump once.

**10. How do I change the win game sound and the end/win game sound and HUD GUIs?**

Go to the main player prefab and you will see a script in the player called player.cs and there you will find the sounds and HUDs of the end game and win game which you can change those to your needs.

**11. 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 that will be displayed.

**12. How do I create a speech hint GUI in game?**

Go to GUIs and drag a speech that you want to be displayed when the player hit a collider. In the speech bubble’s inspector drag and drop the Display Text script and add a 2D box collider but be sure to tag ‘Is Trigger’ to active. Once the player gets near the speech it will be displayed and you can also play around with the offset to change where the collider will hit to display the speech GUI. Simple, and you will have some small-talk in your game! (P.S an empty speech bubble is located in your folder which can be customized with your text, which I suggest using a simple software to just add text and save it to display it in your game)

**13. My camera won’t follow the player?**

Delete the main camera in your scene and place the Player prefab into your scene, a camera should already be attached to the player in this case. If not, you have to just simply attach the main camera to the player prefab to follow the player.

**14. 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 stuck. 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 add sprites only in the scene, and then just add a box collider or polygon collider directly on the whole sprites so you will be sure that the player will not bounce off on any colliders.

**15. 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.

**16.** **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.

**17.** **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.

**18.** **Which tag should I use to make something hurt the player?**

The tag you need to use is always called **trap** which if for example you tag any sprite with the tag **trap** then when the player gets near it he will die and the game will end.

**19.** **How can I make the player jump higher and run faster?**

The simple way to make this possible is to edit the mass in Rigidbody 2D of the player. You can change the mass from 0.11 (default) to something like 0.05 and the player will run faster and jump higher.

**20.** **Why don’t enemy move when I place them in the scene?**

Enemies do not move when you place them into the scene, instead you need to animate them so they can move where you want them to move. This can be done simple with animations, see page 9 for more information on how to animate enemies.

**21.** **How can I make the player jump higher and adjust the jump rate of each jump?**

Go to the script JumpCooldown.cs and open it up. Find the line;

gameObject.GetComponent<Rigidbody2D> ().AddForce (Vector2.up \* 35);

You can change the 35 to higher (or lower) to change the jump speed. The same goes for the other script below it for the double jump.

**22.** **How can I change the jump rate cooldown where a player can jump after he finish jumping?**

Go to the main player prefab in the player’s folder and go to the inspector. Find the Jump Cooldown script in the inspector of the player. There you will see a Jump Rate (which is by default 0.7) and you can change this to whatever you want. (Ex: If you change it to 2, that means the player can only jump after 2 seconds after his last jump) Also, if you change this then you need to change the jump animation to fit the player jump too. Find the JumpAnimation.cs script and open it up. Find the first line of script and change the jump rate to the same one you done with the other script (So if you change it to 2 for JumpCooldown.cs, then you need to change it to 2 for the JumpAnimation.cs too)

**22.** **How can I change the speed of the player via script?**

Go to the main player prefab in the player’s folder and go to the inspector. Open the player.cs script and find these lines;

GetComponent<Rigidbody2D>().AddRelativeForce (Vector3.left\*1.9f);

You can change the 1.9f to higher to increase the speed or lower to decrease the speed. Do changes to script lines below and left and right.

***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.*

***Happy game making!***