

– THE SIMPLE PLATFORMER GAME –

TUTORIALS GUIDE

Developed by: ConstructG.com

Copyright Notice

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>

LESSON 1 – SETTING UP THE STAGE

Step 1

- Set the dimensions of the layout to: **(6976 x 480)**.
- Set the dimensions of the margin to: **(6976 x 480)**.

Step 2

- Rename the layout to: *“Main”*.
- Rename the events sheet to *“Main Events”*.

Step 3

- Set up the *“About”* section of the project:
 - Add a proper name.
 - Add a proper version.
 - Add an ID.
 - Add an Author.
 - Add an Email (If applicable).
 - Add a Website (if applicable).
- Set up the project settings:
 - Choose the first layout as *“Level 1”*.
 - Set Loader Layout to *“Yes”*.
 - Turn *“On”* pixel rounding.
 - Set the window size to: **(640 x 480)**.
- Set up the Configuration Settings:
 - Set up the preview browser.
 - Enable the pause on unfocus.
 - Set the *“Clear Background”* to *“Yes”*.

Step 4

- Enable *“Show Grid”*, *“Snap to Grid”* and *“Show Collision Polys”*.

LESSON 2 – ADDING A BACKGROUND

Step 1

- Rename the first layer to *“Background”*.
- Add a new sprite of type *“Tiled Background”*.
- Choose the right background from your folder.
- Click the option of *“Crop Transparent Edges”*.
- Resize your background as needed. [Make sure your background dimensions match respectively with your layout size].

Step 2

- Rename your background.
- Lock the background layer.

LESSON 3 – ADDING THE GROUND

Step 1

- Create a new layer.
- Rename the new layer to *“Ground”*.
- Select the new layer and lock all the layers beneath it.

Step 2

- Create a new sprite.
- Import your own graphics or draw one using the available tools. [Note the size of your imported graphics and make sure it matches with your layout blocks size (e.g. 32 x 32)].
- Click the option of *“Crop Transparent Edges”*.
- Rename your new sprite to *“Ground”*.

Step 3

- Align your new object on your layout depending on your game. [Note: Scaling your block or sprite might distort the art that is associated with it. Avoid scaling unless your sprite’s dimensions supports that].
- Add to your ground sprite the *“Solid”* behaviour.
- Copy and Paste your block or import new ones as needed.
- Lock the Layer that is associated with your ground.

LESSON 4 – CREATING THE PLATFORMER HERO

Step 1

- Create a new layer.
- Rename the new layer to *“Crabby_Spaceship”*.
- Select the new layer and lock all the layers beneath it.

Step 2

- Create a new sprite.
- Import the graphics of your hero.
- Click the option of *“Crop Transparent Edges”*.
- Click the *“Collision Polygon”* icon then select the option of *“Bounding Box”*. This will define a collision detector box surrounding the whole object.
- Click on the origin point icon then right click on the *“origin point”* select *“Quick assign”* from the menu then choose the option of *“Middle”*. This will assign the origin point to the middle of the object.
- Rename your new sprite to *“Crabby_SpaceShip”*.

Step 3

- Scale your hero properly and adjust it on your layout.
- Add to your hero the *“Scroll To”* behaviour.
- Add to your hero the *“Platform”* behaviour.
- Add to your hero the *“Destroy Outside Layout”* behaviour.
- Lock the hero’s sprite associated layer.

LESSON 5 – CREATING THE GAME’S FINAL TARGET

Step 1



- Create a new layer.
- Rename the new layer to *“Target”*.
- Select the new layer and lock all the layers beneath it.

Step 2

- Create a new sprite and Import the graphics of your target.
- Click the option of *“Crop Transparent Edges”*.
- Set the *“Collision Polygon”* of your sprite depending on the shape of it.
- Assign the *“Origin Point”* to the *“Middle”* of the object.
- Add new frames for your target sprite for some animation.
- Define the speed of your animation.
- Enable the *“Looping”* functionality by setting *“loop”* to *“Yes”*.
- Rename your new sprite to *“Target”*.
- Position your sprite properly on the stage and lock the layer that is associated with your target sprite.

LESSON 6 – DESTROYING THE HERO AND RESETTING THE GAME

- Go to the “*Main Events*” Event sheet.
- Define an event to restart the layout when the hero falls outside the layout. A demonstration goes as follows:

1	 Crabby_Spaceship	Is outside layout	 System	Restart layout
			Add action	

Note 1: If you know any kind of programming language, the logic behind it is a simple “*If Statement*”. The “*If Statement*” tests if the object’s current position from the layout is beyond the defined region, it resets the layout from the beginning.

Note 2: This is not only way to implement this. We choose this way as it is very simple to understand and match perfectly with the beginner level.

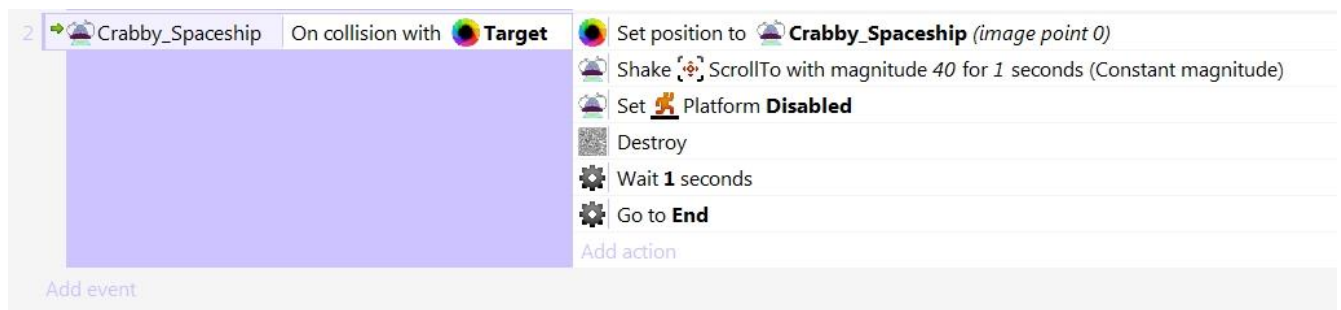
LESSON 7 – MOVING TO LAYOUTS AND WINNING THE GAME

Step 1

- Create a new layout.
- Rename the new layout as “End”.
- Select the new layout.
- Set the size of the new layout to “640 x 480” or any size of your choice depending on your game and requirements.
- Build a new level in the new layout or end your game with a friendly message by creating a new background or sprite.
- Make sure you organize your layers properly when adding objects to your new layout.

Step 2

- Select the events sheet of your main layout (“Main Events”) and define an event when your hero is colliding with your target.
- The event will test the condition of collusion between your two objects before executing the specified list of actions.
- Once the event is defined, add the set of statements to be executed as a result. In this case, we will move from our “Main” layout to the “End” layout.
- To spice up our animation a bit, include more events for generating effects. Set the “Target” to move towards the position of the hero “Crabby_SpaceShip” in order to give a visual feeling that this spaceship is getting absorbed within the “Target”.
- Assign a “Shake” event to shake the focus camera of the stage.
- In order to prevent any unexpected movement from the player, set the platform to “disabled”. This way, the player won’t be able to use any navigational keys and will simply be forced to view the animation till the end.



LESSON 8 – TESTING AND PUBLISHING YOUR GAME

Stage 1 - Testing

Test your game properly and play it several times while aiming to spot as many mistakes as possible before publishing it. Quoting from the official website of Scirra, to test your game during development you can preview it by clicking the "play" icon in the quick access toolbar or ribbon, pressing F5, or by right-clicking a layout in the Project Bar and selecting Preview. This uses a local server to preview your game on. You can change the browser that you preview with using the Preview browser setting in Project Properties.

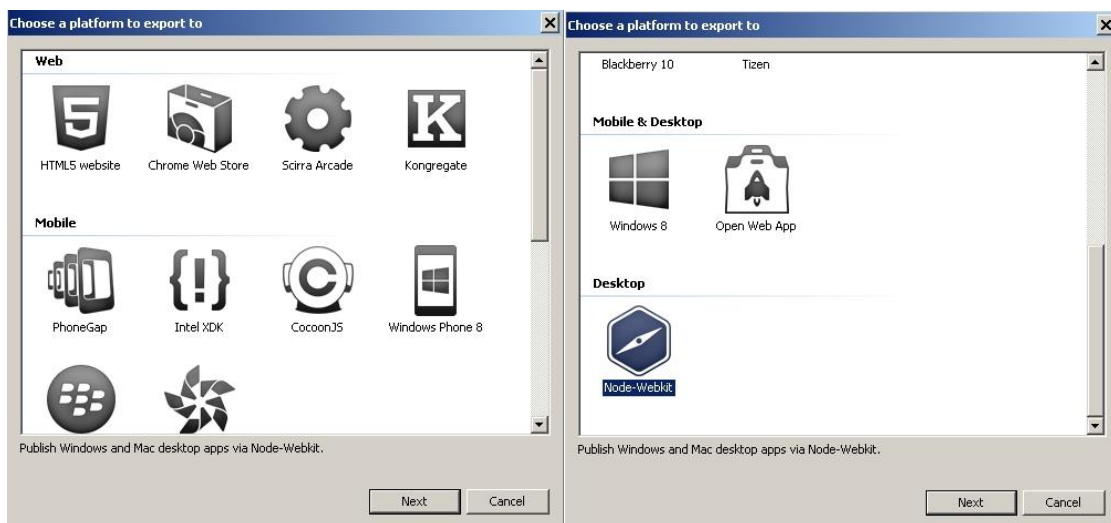
If you have mobiles, tablets or other laptops/PCs you want to test on, previewing over your local network (e.g. Wifi) can make testing considerably easier. For more information, see the tutorial [How to preview on a local network](#). Preview-over-Wifi is also supported with the CocoonJS Launcher for quick testing in CocoonJS.

You can also hold shift and preview to do a continuous preview. This saves the game, updates it, then loads the previous state again.

You must export then publish (i.e. upload) the project before anyone else can play it. You cannot share the URL during preview over the internet.

Stage 2 - Publishing

To export your project, use the File menu, ribbon or Project Bar to open the Export Project dialog. Construct 2 supports many platforms. (Check the two screenshots for the available types by the time of writing this manual).



For this tutorial, we will choose the **"HTML5 Website"** and **"Node-Webkit"**.

"Construct 2 allows exporting projects with a technology called node-webkit. This is effectively the Google Chrome desktop browser packaged up to stand alone with your project. Therefore its performance and features match those of the Google Chrome desktop browser, which is a mature and high-quality browser with excellent performance. Construct 2 can export to Windows, Mac OS X, and both 32-bit and 64-bit Linux with node-webkit." – From the Official Scirra Website.