Windows Game Workshop Hands-on Lab

Workbook 1: Build a classic Platformer game with Construct 2

Lab version: 1.0.0

Last updated: 10/19/2013

CONTENTS

Overview

1. Welcome to the **Windows Game Workshop**! This workbook will help you build useful Windows and Windows Phone games using Scirra's Construct 2 game development tool. You will use the tool's built-in Platformer template and assets from the XNA Platform Starter Kit to create a simple, but complete platform game.

2.

- **Workbook 1 – Build a classic Platformer game with Construct 2**
    - Choosing and API & Starter Kit
    - Create an API Developer Account
    - Understanding the Starter Kit
    - Understanding the API Call
    -
- **Workbook 2 – To the Windows Store!**
    - Get a Windows Developer Account
    - Using the Dashboard
    - Submitting an App
    - Addressing Certification Issues

As you work though these workbooks, you will learn how to work with various object types in Construct 2, and how to leverage existing assets to go from a functional, but unattractive, template, to a polished game. You'll also learn how to export your game for Windows 8 or Windows Phone using Construct 2's built-in export feature.

Objectives

3. This workbook will show you how to:

- Download and start Construct 2
- Open one of Construct 2's many helpful game templates
- Understand the structure of a Construct 2 game
- Add graphical and sound elements to produce a more polished result

- Export your game for publishing on the Windows 8 or Windows Phone platforms

---

System requirements

4. You must have the following to complete this workbook:

- Microsoft Windows 8
- Microsoft Visual Studio 2012 (any version that supports Windows 8/Windows Phone 8 development)
- Construct 2 (or access to the internet to download it)

Exercises

5. This workbook includes the following exercises:

1. Download and install Construct 2
2. Open and examine the Platformer template
3. Modify the template for a more polished game
4. Export your game for Windows 8 or Windows Phone 8

---

6. Estimated time to complete this workbook: **45 to 60 minutes**.

Exercise 1: Download and Install Construct 2

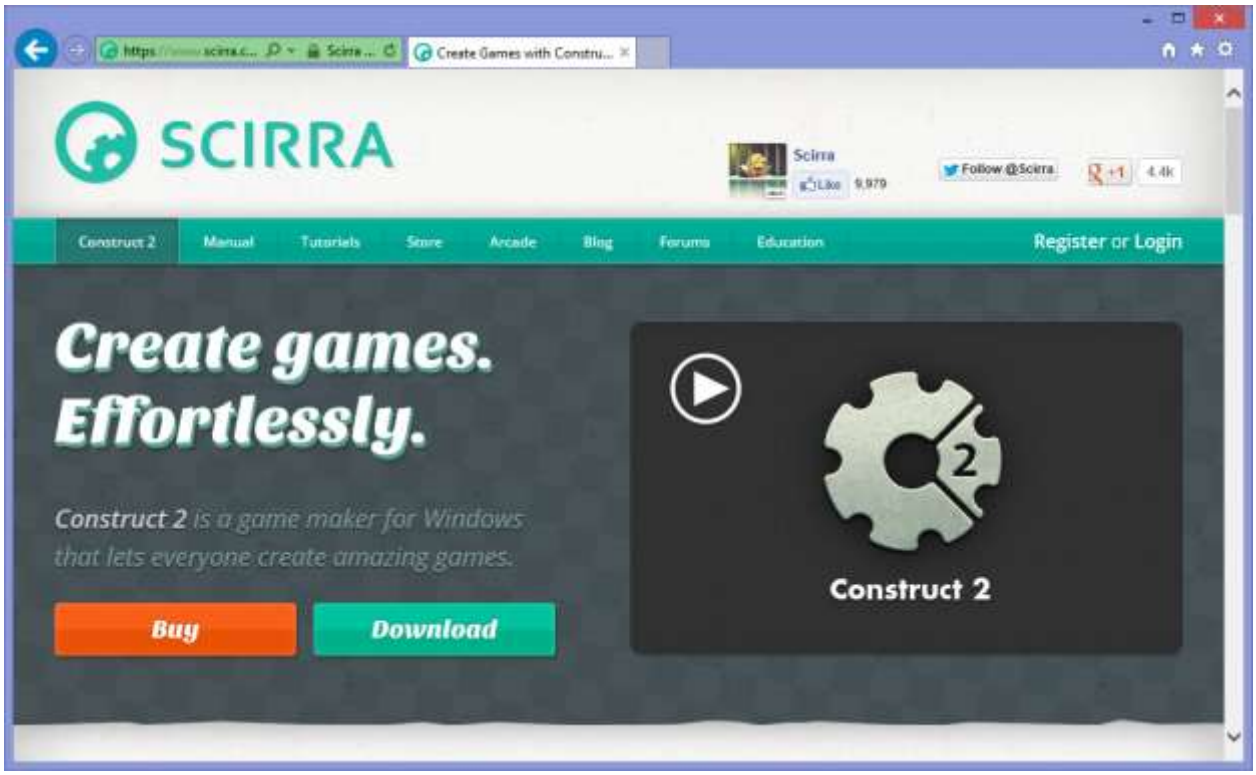
2. In the first exercise, you'll select the API to use, register for a developer account, and download the associated Starter Kit.

Task 1 – Download Construct 2

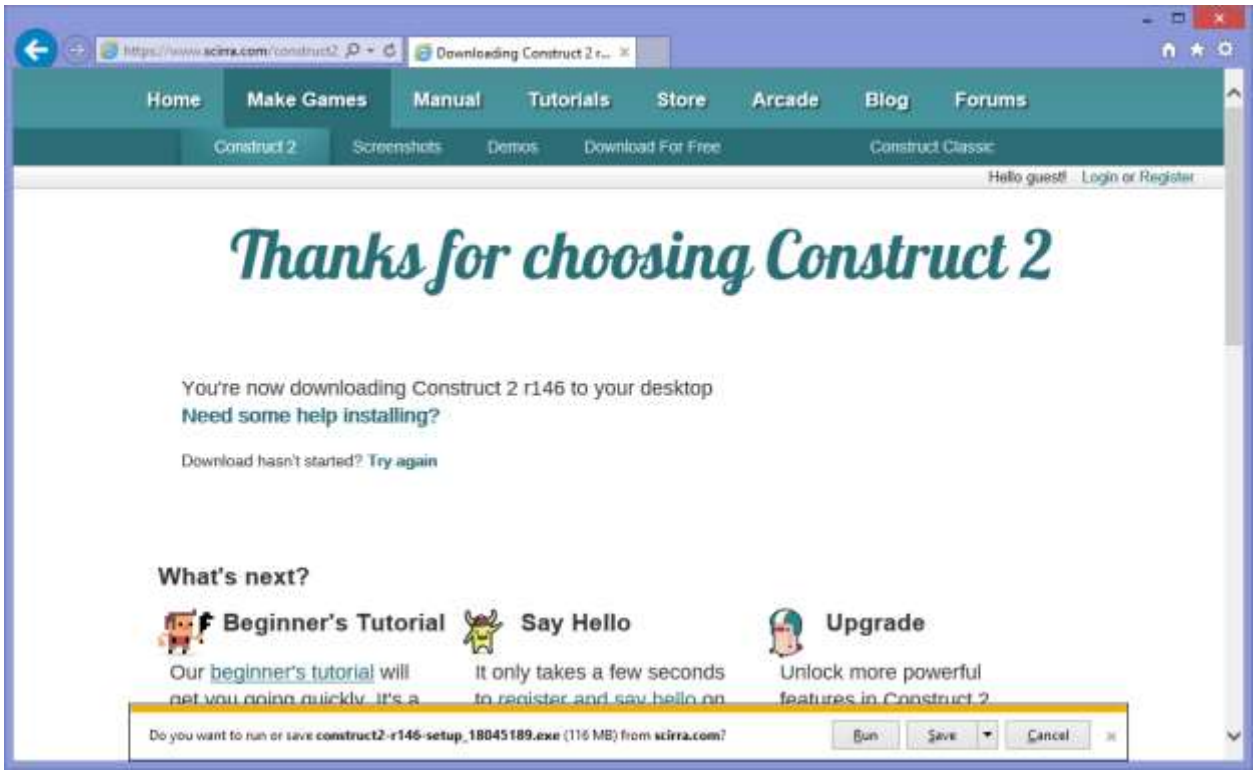
1.

1. Browse to http://www.scirra.com:

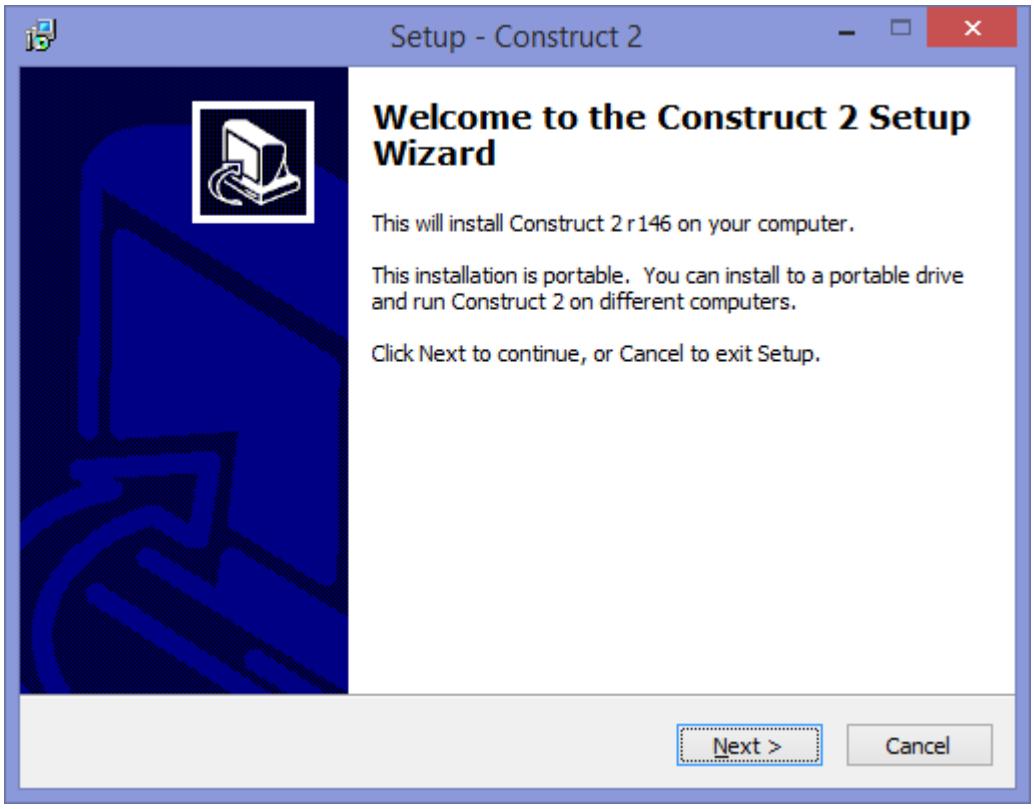
2. Look for the big Download button, and click it:



3. Click either the Run or Save button (depending on whether you want to save a copy of the installer on your local system):



4. Click Yes to the User Account Control security prompt.
5. Complete the Setup Wizard for Construct 2:



Exercise 2: Open and Examine the Platformer Template

3. Construct 2 features many useful templates and examples. Several templates demonstrate the basic concepts of a particular style of game. One such is the Platformer template, which sets up an environment with the basic elements of a platform style game similar to Mario Brothers, Prince of Persia, etc. By examining this template, we'll see how Construct 2 makes creating this type of game quite simple, through its use of game objects, including Sprites and Tiled Backgrounds, behaviors, including the Platform behavior, and events, which allow game developers to handle and respond to user input, among other things.

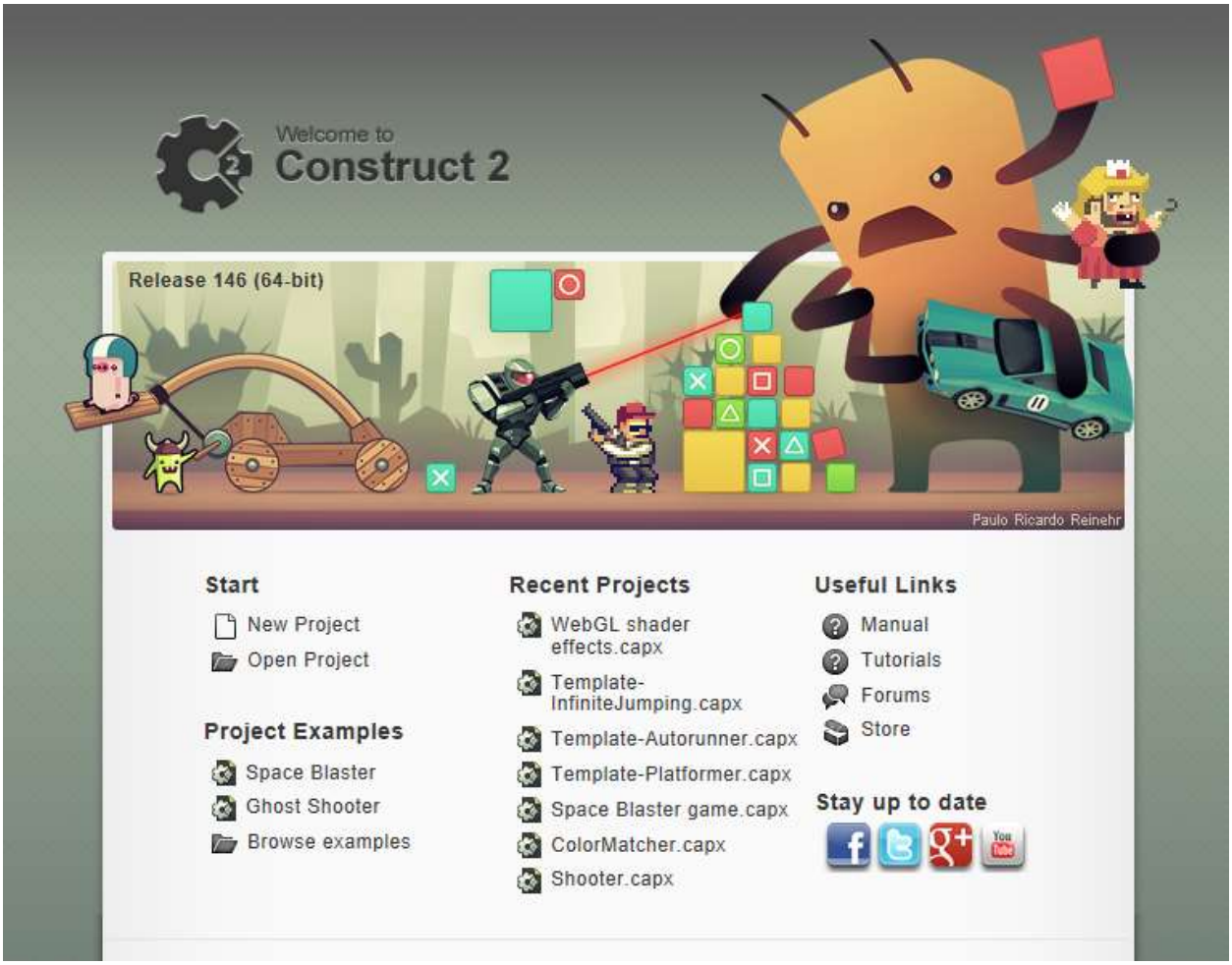
Task 1 – Open Construct 2

1. When the Setup Wizard for Construct 2 completed, you should have seen a checkbox offering to launch Construct 2. If you did not check that box on completing setup, you'll need to open Construct 2.

    6. Click the Start button on your keyboard, and start typing "Construct 2". You should see the list of programs filter down until you see the tile for Construct 2 (if you're on Windows 8.1, the filtered list will appear in the Search bar on the right side of the screen). Tap or click the tile to launch Construct 2 (or hit the Enter key, if the Construct 2 item is highlighted).

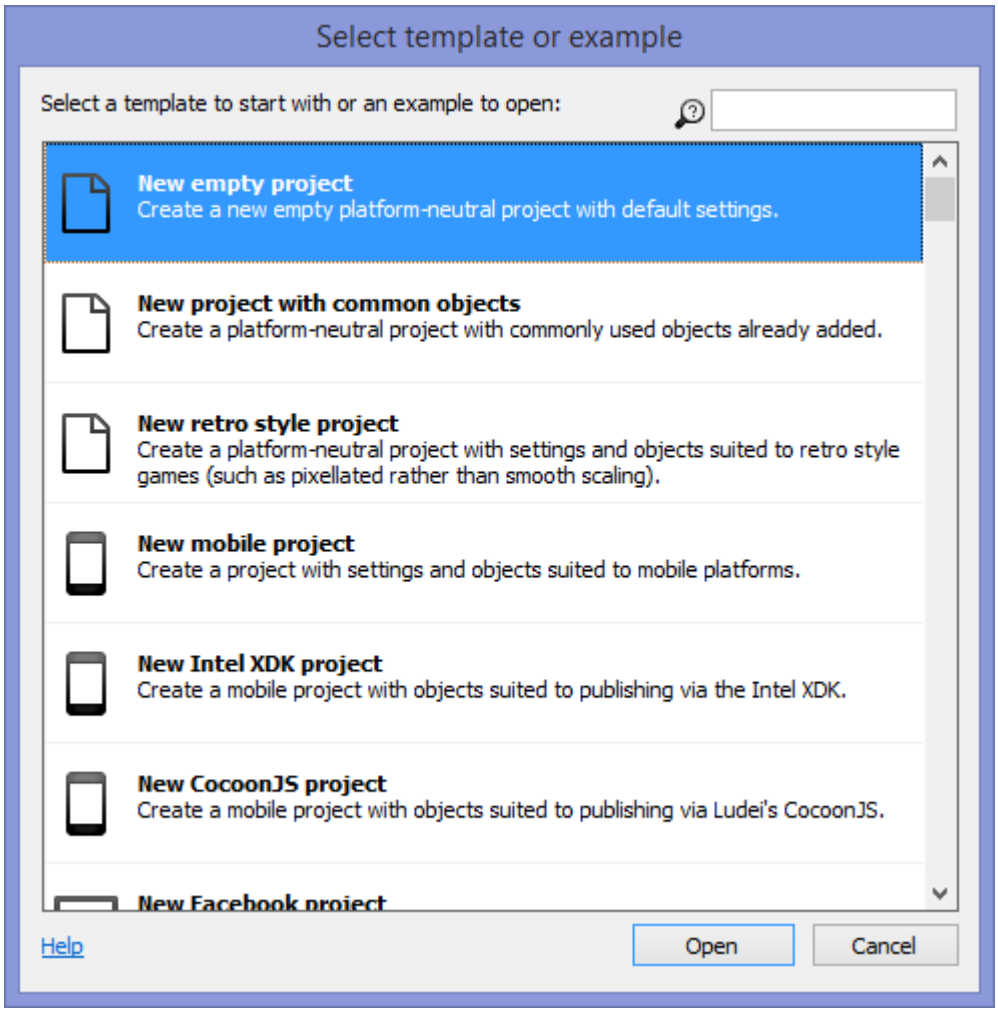
Task 2 – Open the Platformer Template

2. When Construct 2 launches, by default you'll see the Construct 2 Start Page, which loads into the Construct 2 editor pane:

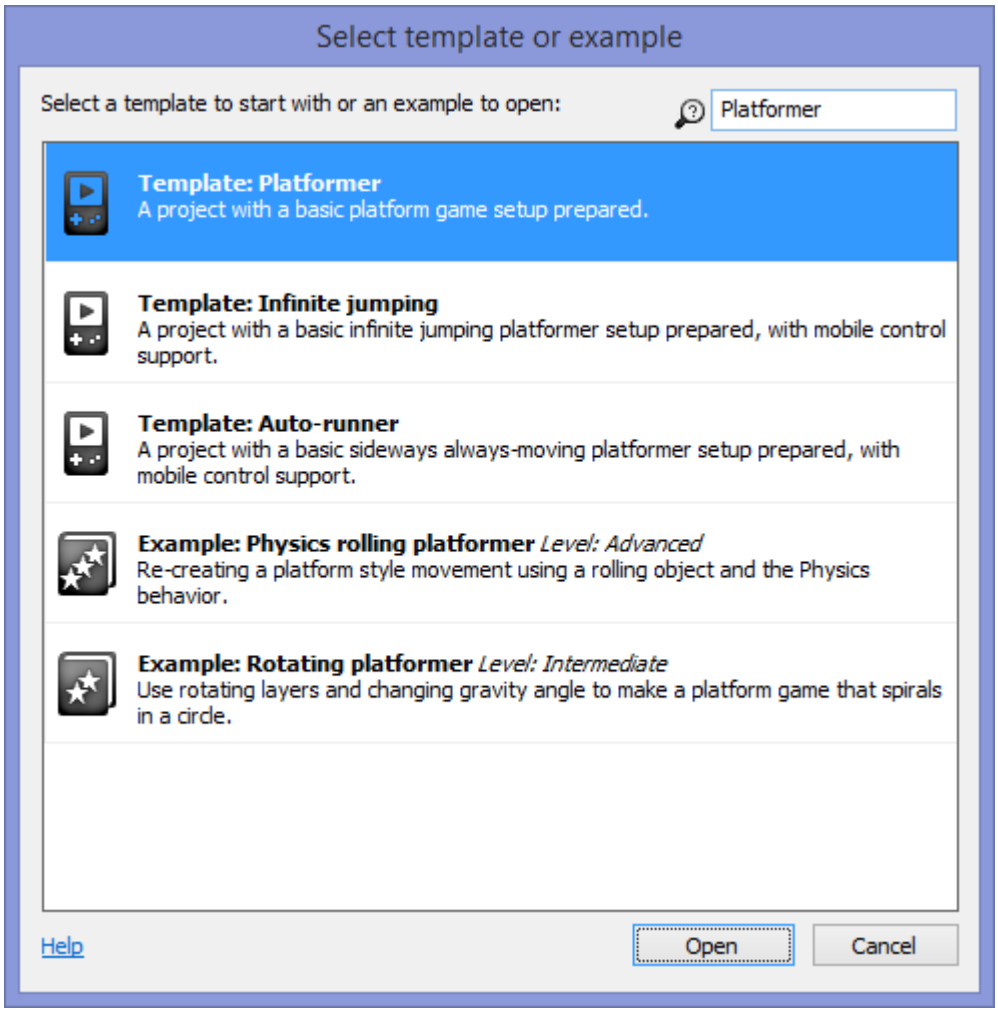


Notice that the Start Page includes many helpful items, including any recent projects you've worked on, links to the online manuals, tutorials, forums, and store, project examples, and the New Project and Open Project options.

1. Click the New Project item on the Start Page, which results in the following dialog:



2. In the search box, type “Platformer” as shown below:



3. With the Platformer template selected, click the Open button. This will open the selected template.

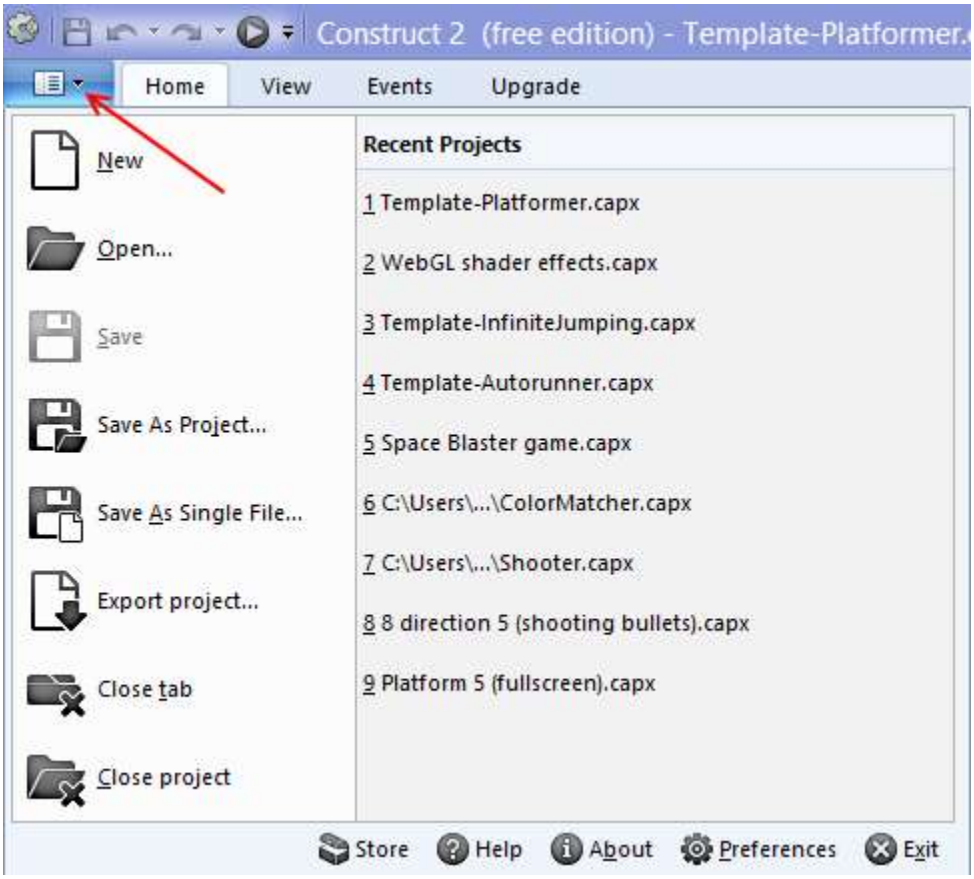
Task 3 – Explore the Construct 2 environment and the Platformer Template

When you’ve opened the platformer template, Construct 2 will add two tabs to the center pane of the environment (the one where the Start Page opened earlier), which include the default Layout for the template, and the default Event Sheet.

Layouts are used to arrange elements (Sprites, Text, etc.) to form the visual aspect of a game, while Event Sheets contain a set of one or more “if-then”-style rules that provide the logic behind the game. For example, the event sheet created as part of this template has a set of rules that handle keyboard input, and respond by moving the player sprite appropriately.

Start the exploration of the Construct 2 environment and the template from the top left, starting with the File menu and ribbon:

1. Click the button for the **File menu**, just to the left of the Home tab:



   Note that like the Start Page, this menu allows you to create or open projects, and access recently-used projects. It also contains commands for saving your project (you can save as a project, or as a single file with the .capx extension, which makes sharing code with others very easy), and exporting your project to a variety of platforms. Here is also where you can access the preferences for the application.

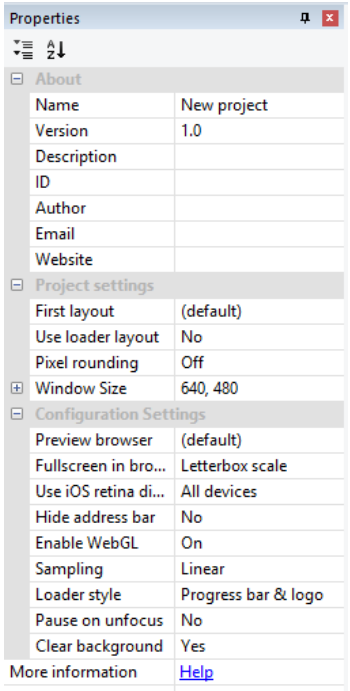
2. The **Home tab** on the ribbon is selected by default, an important section to take note of is the Preview section, which includes the Run layout and Debug layout commands. You will use those frequently for testing, so make sure you can find them easily.

3. Click the **View tab**. This tab allows you to configure various visual elements and windows in the environment. Note that some bars only work in the paid version of Construct 2, so if you attempt to enable them, you'll be asked to upgrade. The Zoom options are fairly self-explanatory, and are useful for viewing your entire layout, or zooming in to specific details (note that you can also use pinch-to-zoom on touch machines to control zoom). The Grid options and Grid size sections allow you to modify a grid that helps you more easily align game objects to one another. Lastly, the Display section allows you to view the collision polygons around game o objects, which can be useful when things aren't quite working as you expect.

4. Click the **Events tab**. Note that if you don't have an event sheet loaded, all the controls here are disabled. Switch to Event sheet 1, and you'll see that many of the controls are now enabled. Look over the available commands. You'll do more with these in a later part of the workbook. One thing to note is that all of the commands on the Events ribbon tab can also be accessed by right-clicking in the event sheet itself, either in a blank area, or on a specific event or action.
5. The remaining tab, **Upgrade**, contains a few buttons that pertain to upgrading to a higher edition of Construct 2. If you wish to charge for your game, you will need to upgrade to the Personal license at a minimum, but for the purposes of this workbook, the free edition will work fine.

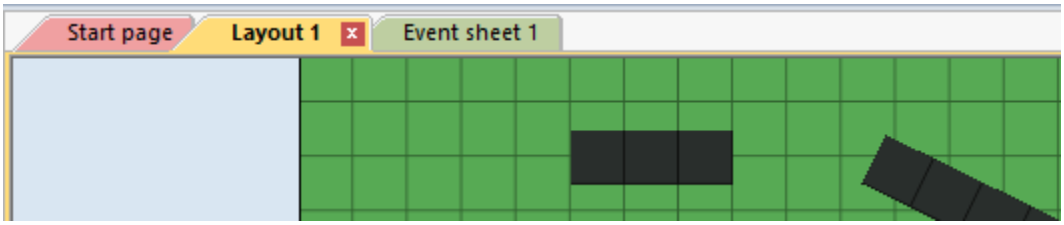
That completes the tour of the menu and ribbon tabs. Below these, you'll find the windows (Construct 2 calls them "bars") you'll use to build and modify your games. Starting from left to right:

1. On the far right, examine the **Properties bar**, which lists properties of the currently selected item, whether an object on the current layout, or an item from one of the other bars, such as the Projects bar or the Objects bar. By default, unless you've selected something else, this will display the properties of the project, as shown below:

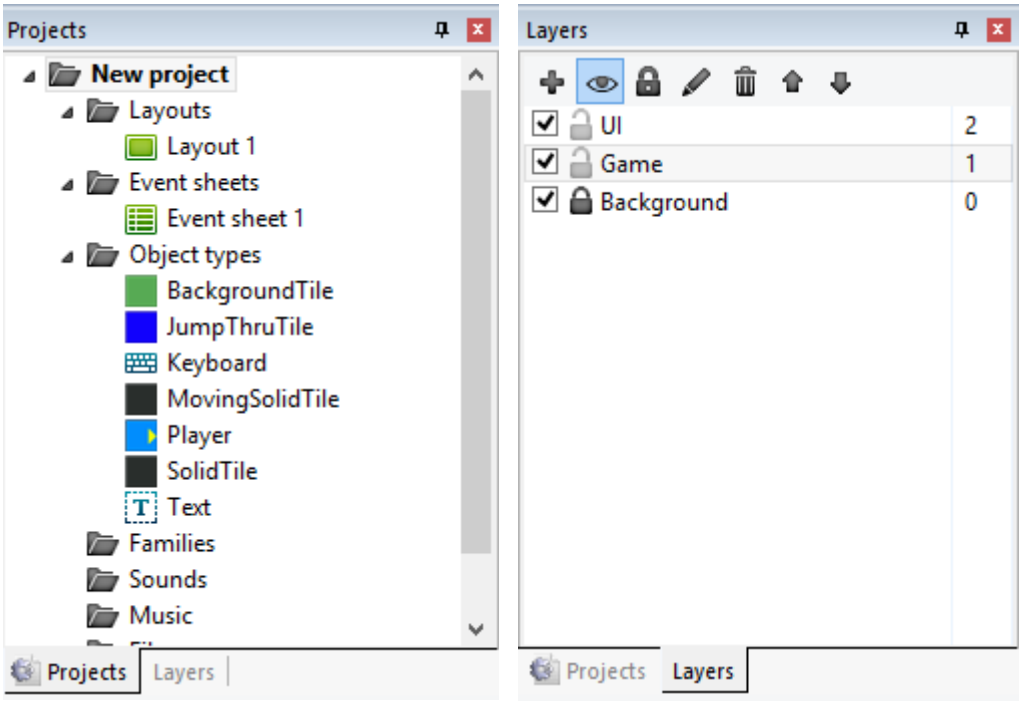


2. In the center, examine the **Layout View**, which is the visual designer for Construct 2, and is where you'll edit layouts and event sheets (in the **Event Sheet View**). For the Platformer template, the Layout View will include Layout 1, the only layout in the template, and Event sheet 1, the corresponding event sheet. If you left it open, you'll also see a tab for the Start

Page, as shown below:



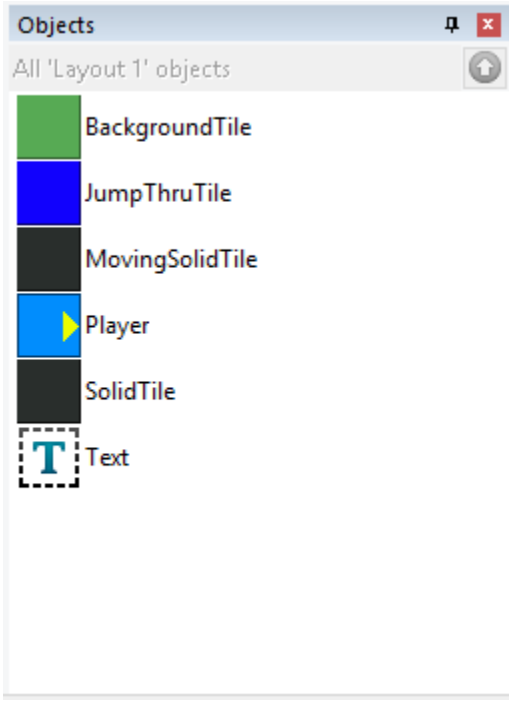
3. To the right of the Layout View are two sets of bars. At the top are the **Projects** and **Layers bars**, which are tabbed by default. Click into each to examine them:



The Projects bar lists all items within the project, including layouts, event sheets, object types, families (a feature of the paid editions, so we won't cover it here), sounds and music, and files. The Layout bar, meanwhile, lists all of the layers for a given layout. Layers allow you to organize the content in your layout both visually (lower number layers will render behind higher number layers) and also in terms of which objects can interact with other objects (only objects on the same layer will collide with one another, for example). The Layer bar also allows you to lock layers to prevent inadvertent changes, and to select the active layer, which is where new objects will be created, and will also enable selection of existing objects on that layer.

NOTE: *To get help for any given part of the Construct 2 environment, simply right-click and select the help topic that appears at the bottom of the context menu, which will take you to the appropriate section of the online documentation.*

4. Finally, examine the **Objects bar**, which appears by default below the Project and Layers bars:

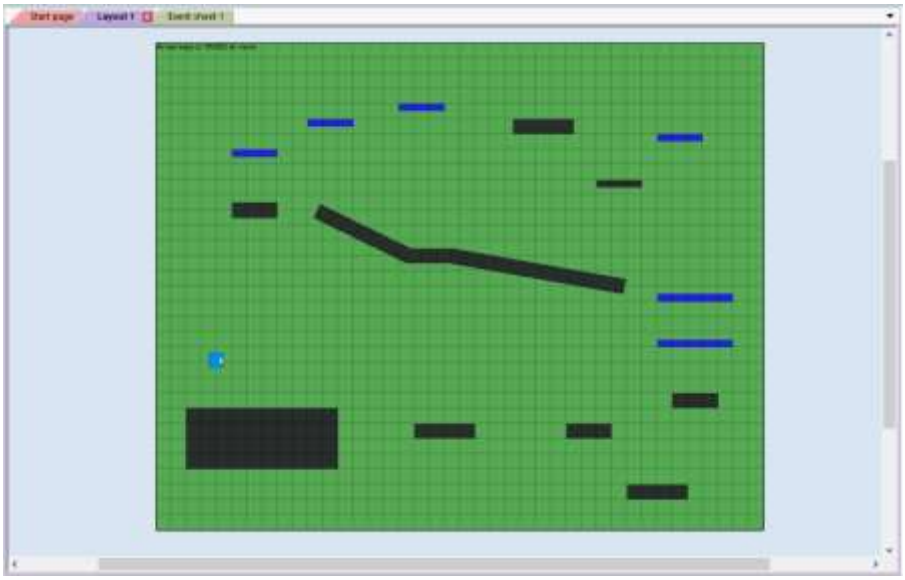


The Objects bar displays a list of all objects in the game. While not possible in the free edition, in the paid editions, you can organize these objects by folder for easier reference. If you select an object from this bar, you'll see its properties in the Properties bar. You can also easily drag an object from this bar onto a layout to quickly add it to the game.

This section has provided a cursory look at the Construct 2 environment. You'll learn more in subsequent exercises, but for more in-depth discussion, please refer to the Interface section of the Construct 2 online documentation. Note that the manual may also be searched, and can be downloaded for offline reference.

Now, let's examine the Platformer template itself:

1. If it's not already selected, click the Layout 1 tab in the Layout View. This will display the layout for the template (use the View tab's Zoom section to view the entire layout):



You're probably thinking that this game isn't going to win any design awards, and you'd be right. But the point of the template is to set up examples of the basic game mechanics, so that's what we're going to look at.

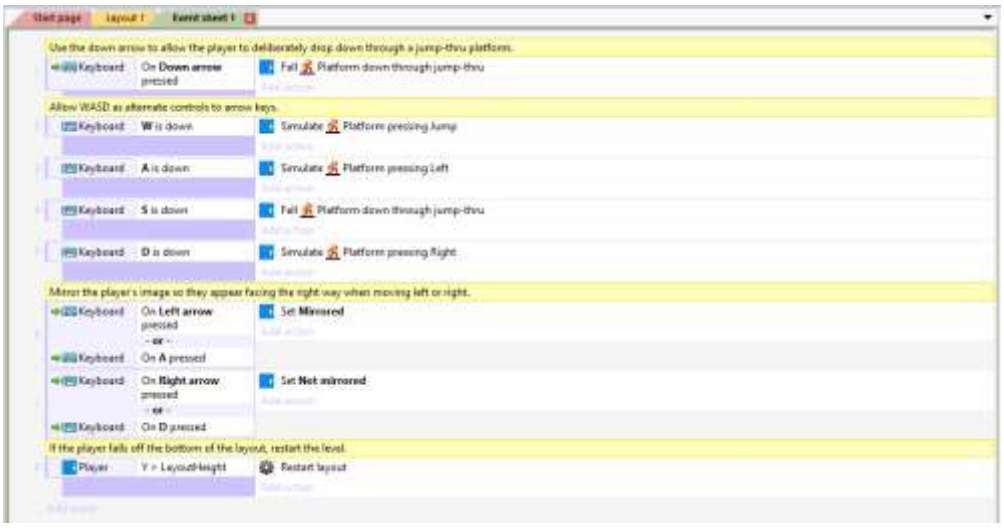
2. Starting with that lovely green background, you'll notice that it looks like a tiled wall. That's because it's essentially exactly what it is. The background is built using an object based on the Tiled Background plugin. This allows you to take any image and automatically tile it over the full extent of the layout. This makes it easy to use texture images like grass or dirt to quickly build top-down shooters, or build a basic sky backround with smaller images than would be required for a full-size background image. This can be more efficient than using larger, higher-resolution images (which can be particularly important when developing for mobile devices). If you select the **BackgroundTile** object from the Object bar, you can view its properties. Note that among other properties, objects support opacity, position, size, and many support one or more instance variables (variables associate with a specific instance of an object…great for storing things like an enemy's health), one or more behaviors (behaviors allow you to add sophisticated functionality to an object without writing code), one or more effects (effects require WebGL support, so are only available in Windows Store apps targeting Windows 8.1), etc.
3. Sticking with the Object bar, the next item is the **JumpThruTile** object. There are several instances of this object on the layout, all colored in blue. Note that these objects, along with the other "platform" type objects, are all located on the Game layer, as is the Player sprite. Select the JumpThruTile object in the Object bar, and you'll see that it has its Opacity property set to 80 (80% opaque), and is also the first object we've seen with a behavior attached, in this case the JumpThru behavior. This behavior allows other sprites to pass through it from below, but is solid from above. Objects with JumpThru applied may also allow other sprites to pass through from above when the down arrow is pressed (more on this later).
4. Next up is the **MovingSolidTile** object, which has two behaviors attached, the Solid behavior, which very simply makes the object solid and impassable, and the Sine behavior, which is used to create oscillations in an object's movement, size, opacity, or an arbitrary value which can be used in calculations. In addition to supporting a true sine oscillation, it also supports other

waveforms, including triangle, sawtooth, reverse sawtooth, and square. Note that the object itself does not include a value for the Movement property. If you select one of the object instances on the layout (they're highlighted when you select the MovingSolidTile object from the Objects bar), you'll see that some have this property set to "Horizontal" and some to "Vertical". Later, when you run the layout, you can play with these values to see how the other values affect the behavior of the object instances.

5. Next is the **Player** object, which uses the Sprite plugin. The Sprite is the most basic of objects, and is used for many purposes in games, including the player character, and any on-screen enemies. The Player object has the Platform behavior attached. This behavior provides a great deal of functionality very quickly, including mapping the arrow keys on the keyboard to move the Player left and right, jump, and drop through platforms using the JumpThru behavior. And the properties of the Platform behavior allow you to control such factors as how fast the Player moves, how high they can jump, and how quickly gravity accelerates the Player when falling. The Player object also has the ScrollTo behavior attached. This behavior has no properties, and simply ensures that if the Window Size property of the project is smaller than the Layout Size property of the layout, then the Layout will scroll to keep the Player object centered on-screen. If more than one character has this behavior, the scrolling will work to keep the center point between them centered on-screen.
6. The last object on the Game layer is the **SolidTile** object. As the name suggests, this object has the Solid behavior, and is based on the Tiled Background plugin. Most of the platforms in the template are instances of the SolidTile object.
7. Lastly, examine the **Text** object, which is used to display some simple instructions on how to play. In addition to Text objects for displaying text (scores and other HUD info, instructions, etc.), Construct 2 supports the use of Sprite Fonts, which can be great for retro-style games.

Our tour of the template is almost complete. To wrap things up, let's look at Event Sheet 1:

1. Select the "Event sheet 1" tab in the content area:



2. Exmine the existing rules.
   Nearly all of the existing rules in the event sheet are specific to Keyboard input. While the

Platform behavior attached to the Player object automatically enables control of the player via the arrow keys, the additional rules allow the use of the WASD keys as a substitute, which is a kindness for left-handed folks like myself. There's also a rule to allow the player to drop through any JumpThru platform, using down. Additionally, there's a rule that automatically mirrors the player sprite, so it's clear which direction you're moving/facing. And finally, there's a rule that resets the player location if the player falls off the layout.

3. Double-click on an existing Event or Action to see additional options for them. Also try right-clicking a given event or condition or action, to see the available options. We'll revisit these later in the workbook.

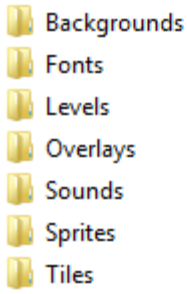
Exercise 3: Modify the Platformer Template

4. Now that you have a basic understanding of the Platformer template, it's time to update it with some more polished assets. To get started, you'll need some assets to work with.

Task 1 – Download the XNA Platform Starter Kit

To get started, it's helpful to have assets (sprite images, tiles, sounds, etc.) so you can focus on the development tasks, rather than worrying about the creative aspects. To make this easy, I suggest using the assets from the XNA Platformer Starter Kit, which is available with the Microsoft Permissive License, which provides broad latitude to reuse and create derivative works.

1. Download the starter kit from: http://aka.ms/PlatformerSK
2. Be sure to read the license in the root of the Zip file.
3. The content you need is in the Platformer\Content folder:



Task 2 – Update the Player

Since it's the central part of the game, let's start by improving the player sprite. The good news is that the existing Player object already does what we need it to do, behavior-wise. But it needs some improvement in the visual arena, which is where the Platformer Starter Kit assets come in. You might

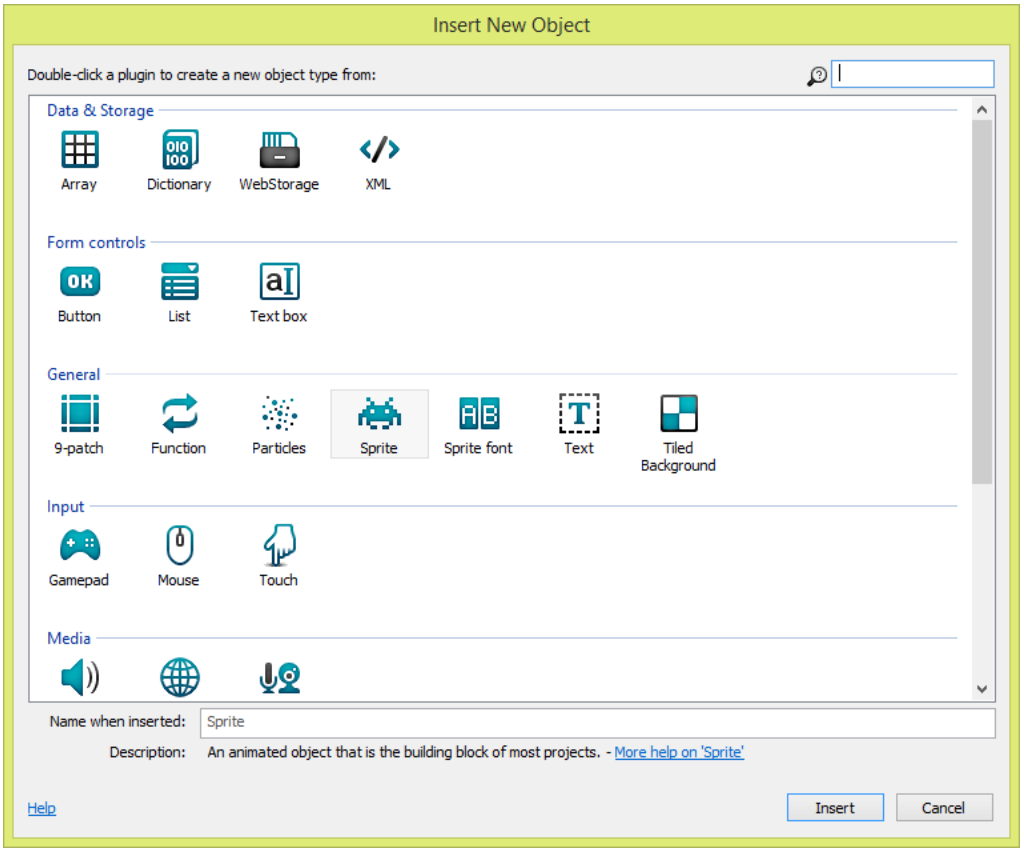
think it would make sense to simply replace the existing Player sprite's image with the images from the assets, but while that might seem the intuitive way, it can introduce unexpected results, because many of the assets are used for animations, such as running, and each frame of the animation may be slightly different in overall shape. Depending on how the collision polygon for the frame is drawn, this can produce strange results, such as the player hanging from a platform by their arm or nose. Instead, what you need to do is use the existing Player object to handle the movement aspects, and add another sprite that will handle the visuals, and track the location of the Player object.

5. Before starting modifications, use the File menu to save the game as a single file, choosing whatever location and name you prefer. You should save your progress often in case you run into any issues.

6. Since the current Player object will serve as an invisible guide for our visual representation, rename the current Player object to "PlayerBox".

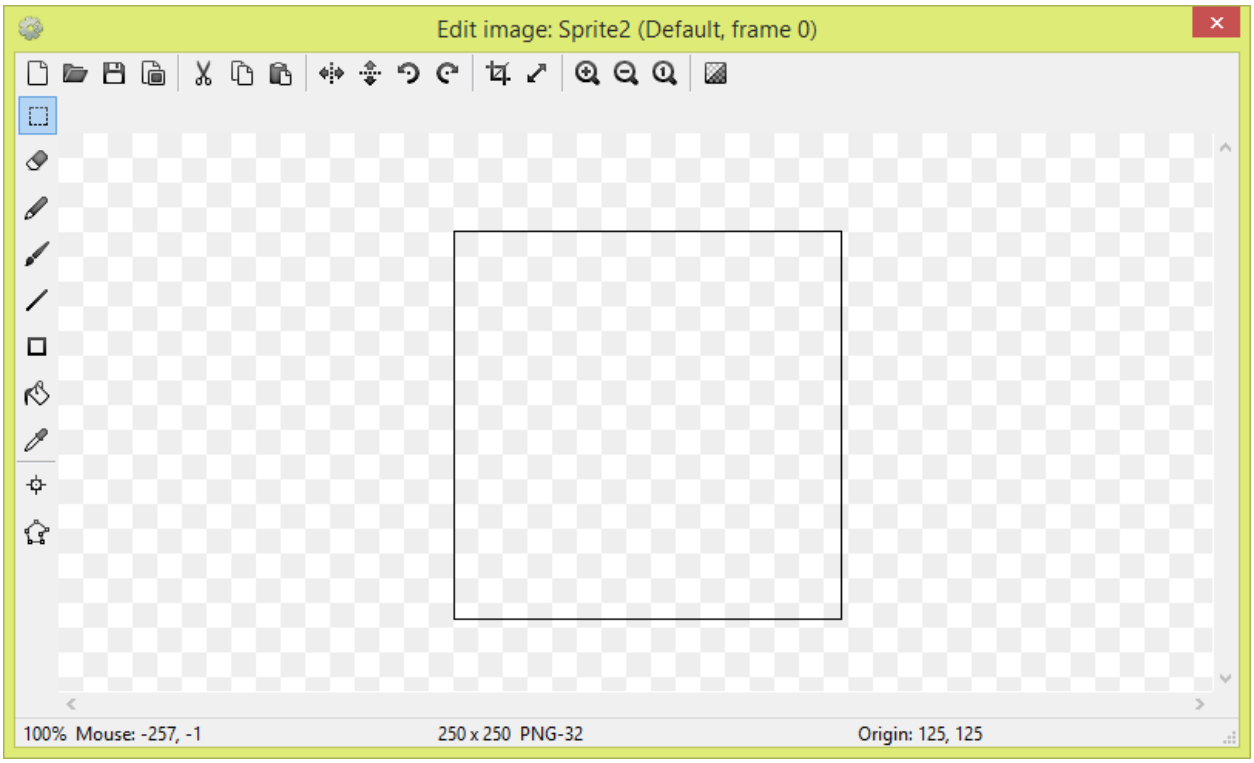
7. With the PlayerBox object selected, look for the Initial visibility property in the Properties bar, and set its value to Invisible. This will only affect the visibility of the object at runtime.

8. Double-click on the layout, and in the Insert New Object dialog, select Sprite, and click the Insert button (note that you can also simply double-click an object to insert it directly:



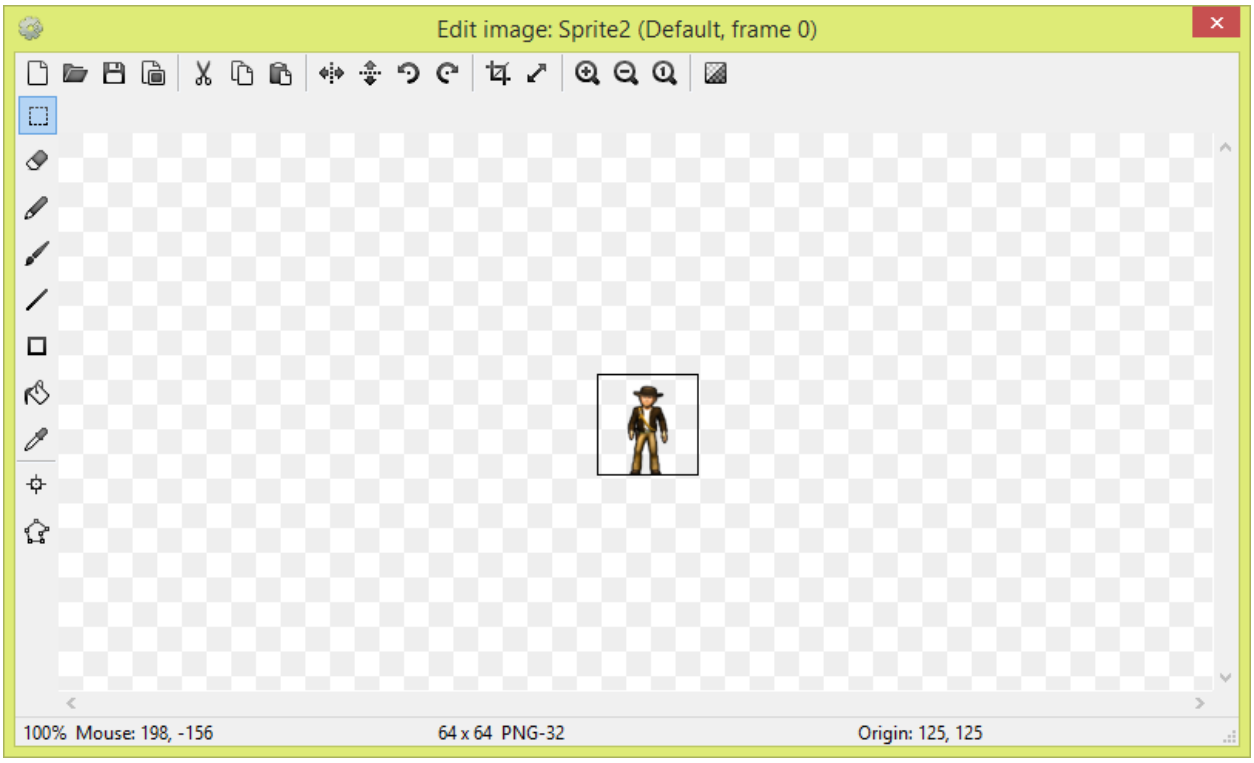
9. The cursor will change to a crosshair, with the text "Game" next to it, indicating that you will be inserting the new Sprite object in the Game layer…if the text does not read "Game", hit the escape key, and go to the Layers bar and ensure that the Game layer is selected and not locked, and try the previous

step again. When you're ready, click over the center of the PlayerBox object to insert a new object. This will open the Edit image dialog (along with the Animations and Animation frames dialogs) for the new Sprite object:



10. Click the Load Image button from the toolbar of the Edit image dialog, and browse to the Platformer\Content\Sprites\Player folder, select the Idle.png image, and click Open. This will load the

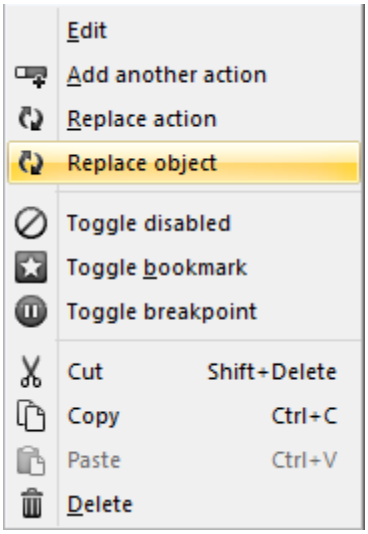
first image for our player:



11. Close the Edit image dialog, and rename the sprite you just added to “Player”.

Note: The platformer template is set up with the assumption that the Player starts out facing right. The image you just imported starts out facing left. There are two ways to fix this…one is to click the Mirror button (which can be applied to a single frame, or with the Shift key, to an entire animation), which appears just the the right of the clipboard icon. The other is to update the event sheet to reverse the SetMirrored command from its default. The latter is probably easier, since all of the animations will have this issue, and it will take extra time to mirror them all.

12. To fix the mirroring issue, switch to Event sheet 1, and locate the section that handles mirroring the player’s image. We need two fixes here…one fix is that we no longer want this event to apply to the PlayerBox object, since it’s now invisible. Instead, we want the mirroring to apply to the Player object. Right-click the Set Mirrored action, and select Replace object:

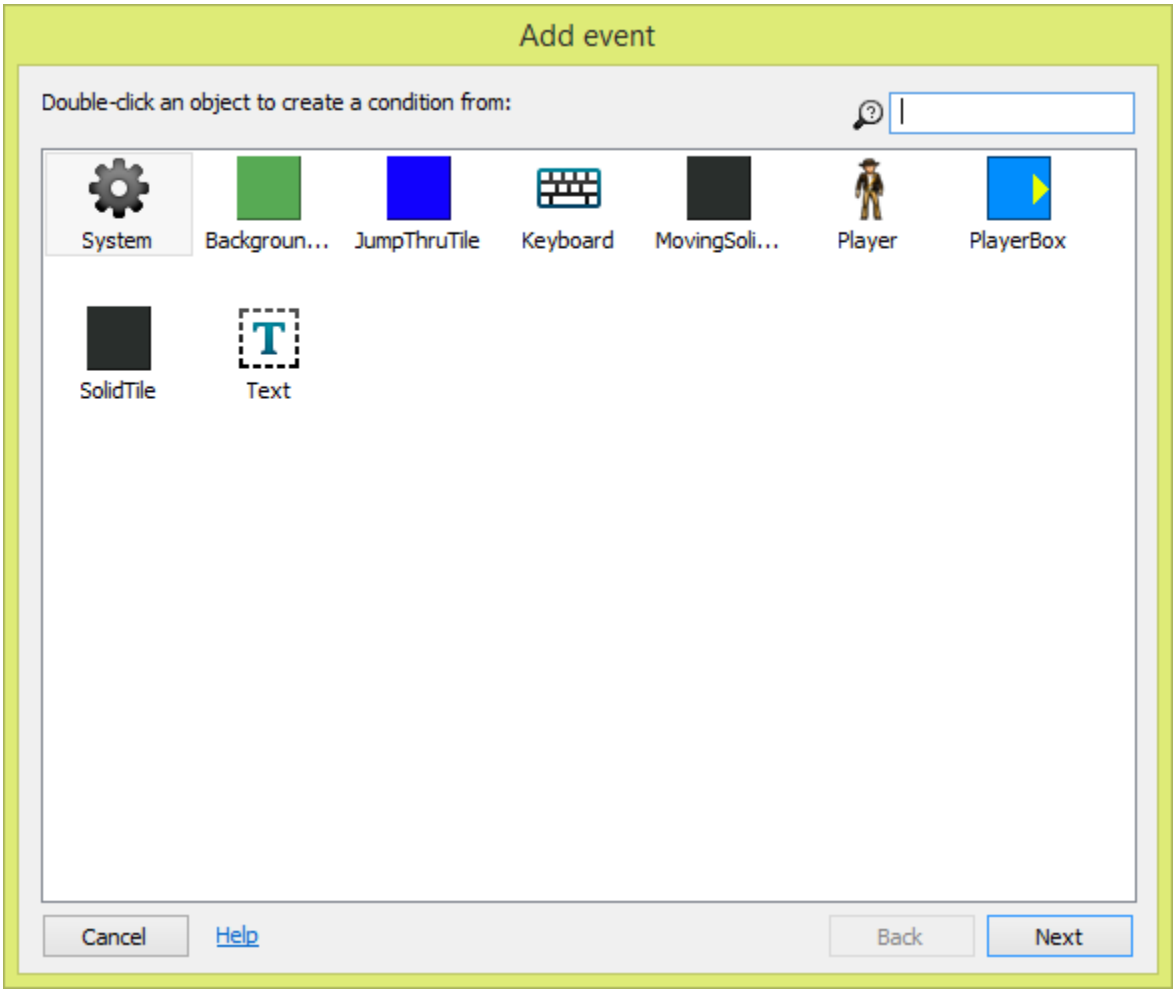


In the subsequent dialog, select the newly-added Player object, and click OK. Do the same with the Set Not Mirrored action.

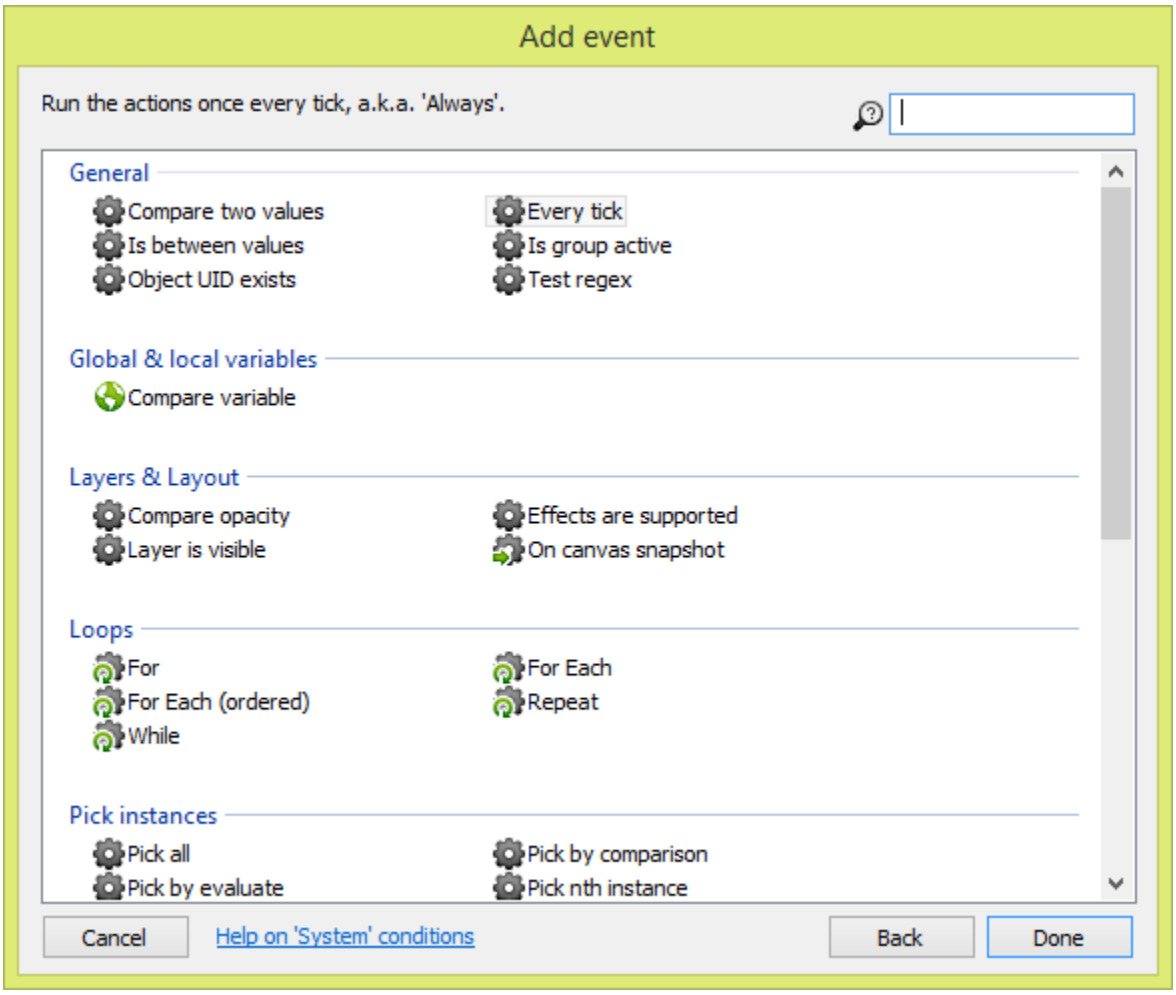
13. Next, double-click the Set Mirrored action, and change the State from Mirrored to Not Mirrored, and click Done. Do the opposite for the other command. Save your progress (Ctrl+S), and click on the Run layout button in the Home tab to test out your current progress.

You probably noticed pretty quickly that while the player is in fact mirroring properly, it's not moving. This is because we need to add a rule to make the Player sprite follow the PlayerBox object.

14. In Event sheet 1, right-click in an empty area and select Add Event. Choose the System object and click Next:

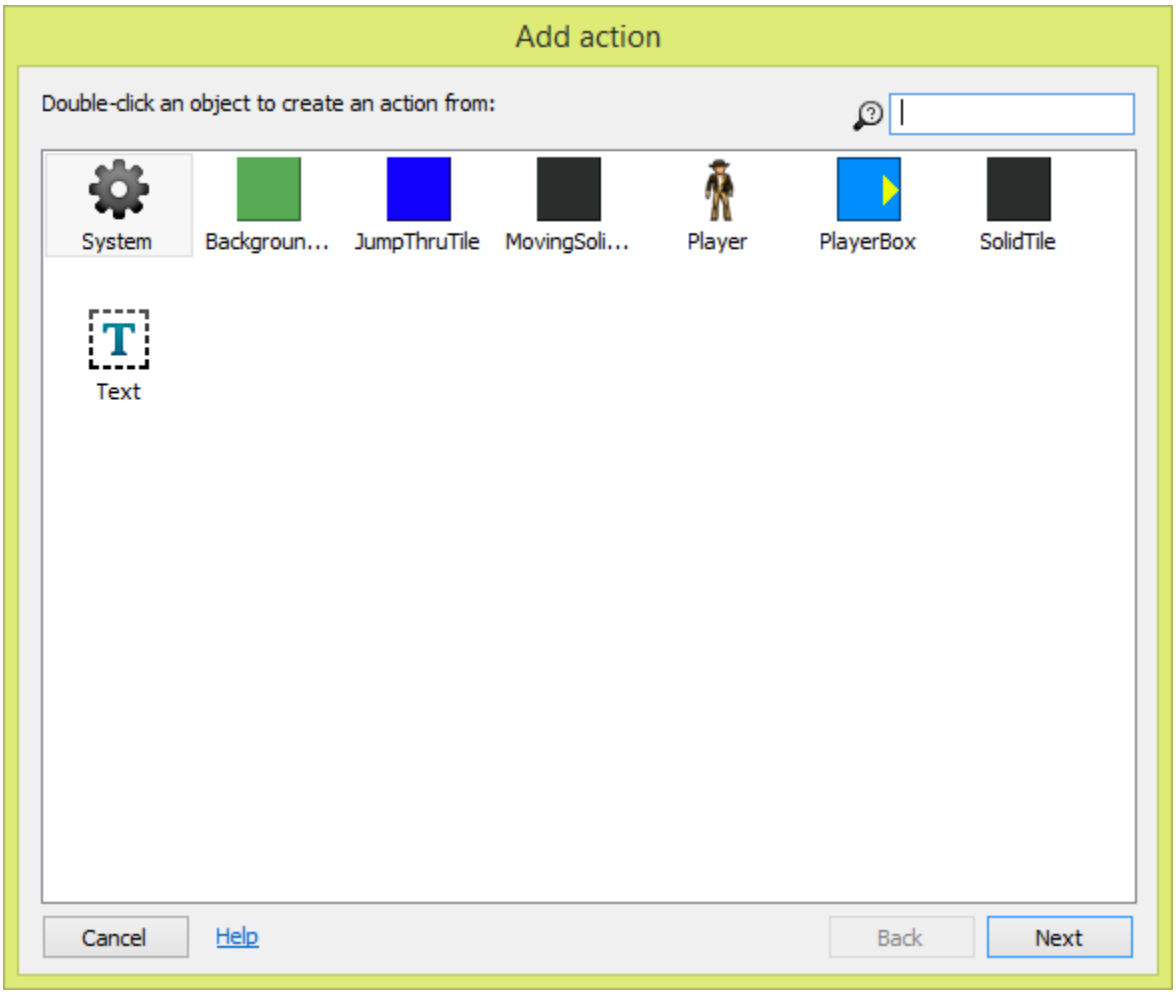


15. Near the top of the next dialog, select the Every tick item, and click Done:

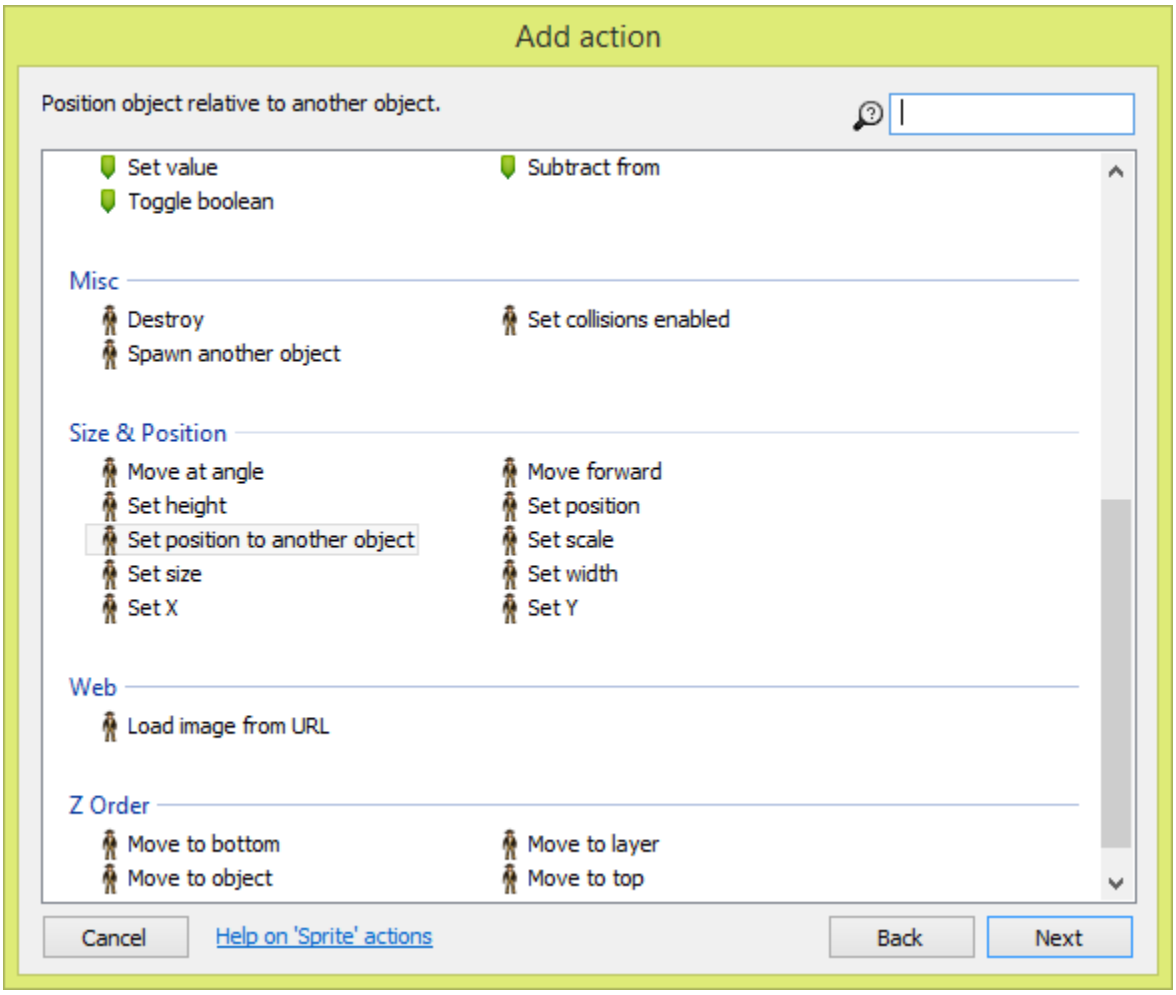


This will add the event to the event sheet. Now you need to add the desired action.

16. Click the Add action text directly to the right of the new event:



17. Click the Player object, and click Next:



18. Scroll down to the Size & Position section, select the Set position to another object item, and click Next:

19. In the resulting dialog, click the <click to choose> button, click the PlayerBox object, and click OK, then click Done on the Parameters dialog. Save and run to test the new event (you may need to refresh the browser to get the latest version of the game code).

The player sprites from the XNA Platformer Starter Kit includes animations for running, jumping, death, and victory. Adding these animations will be left as a separate exercise for the reader, and can be accomplished using the "Adding more animations" section of the following tutorial on the Construct 2 website: https://www.scirra.com/tutorials/253/how-to-make-a-platform-game/page-6

Now that the player is looking better, you can focus on the rest of the layout, starting with the background. If you run the current layout again, and pay close attention, you'll notice that in addition to the player moving on the screen, the platforms actually move relative to the background. This is referred to as parallax, and it's a commonly-used feature in platform games. It's achieved by setting the Parallax property of any given level to a value that differs from the level above or below it. Layers that should not move at all are given a Parallax value of 0, 0. We're mentioning this, because it's likely that part of the reason for using such an obvious tiled background in the template is to emphasize this effect.

That said, the current background is not terribly attractive, so it should be replaced.

1. In File Explorer, navigate to the Platformer\Content\Backgrounds folder of the XNA Platformer Starter Kit. This folder contains three different backgrounds consisting of three layers each.

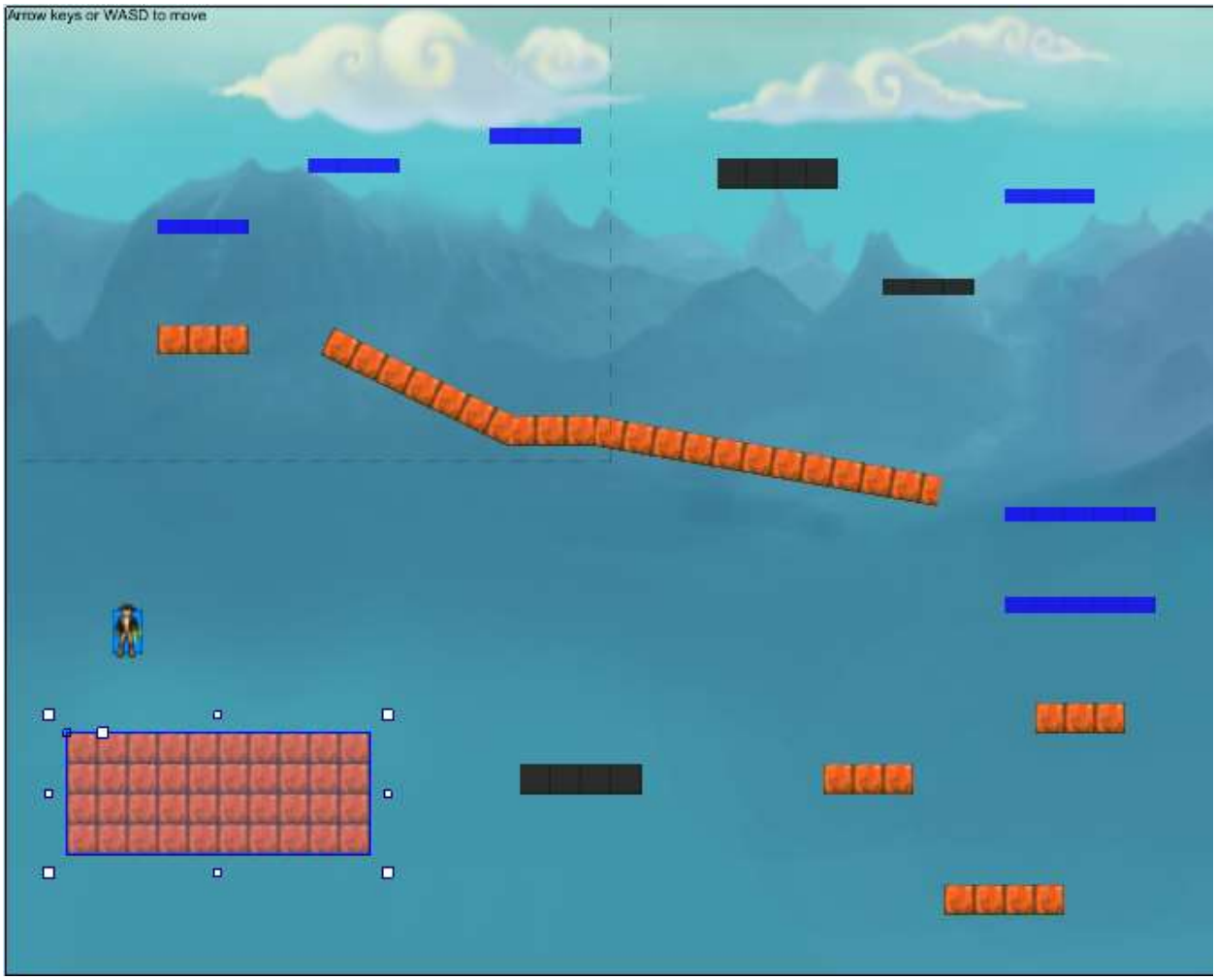
2. Open Layout 1, unlock and select the Background layer from the Layers bar, and select and delete the BackgroundTile object from the layout. Note that this will not delete the object from the project, only from the layer.

3. Double-click on the layout, select the Sprite object, and click Insert. As before, use the Load image button to browse to where the backgrounds are located, and select the desired image. If you want to use all three images, you should add the rear-most image first (you can adjust the z-order later, but this is harder to do in the free version of Construct 2). The images are not large enough by default to fill the entire layout, so move the image to align with the top-left corner, then resize to fill the layout.

4. After adding the desired background(s), to avoid unintentional changes lock the Background layer, then save, then run the layout to see how things are looking (remember to refresh the browser if needed). The game is looking better already!

The last visual element that needs updating is the platforms. The XNA Platformer Starter Kit contains a Tiles folder with assets we can use for a nicer look for the platforms.

1. In the Layers bar, make sure the Game layer is selected, then choose the platform you wish to update, and double-click it to open the Edit image dialog. Using the Load image button, browse to the Tiles folder of the assets we've been using, and select a tile to replace the default. On selecting a desired tile, you'll get a message about the size of these tiles not being the preferred size for tiling. If you click the indicated resize button, and set the Width to 32, that will make the tile work just fine. Close the Edit image dialog to see the result:



In the image above, the SolidTile image was replaced by the BlockA0.png image.

2. Using the same technique, replace the remaining platform objects' images.

3. Again, save and run the layout to see your progress.

Now that you've got the basics covered, as an additional challenge, use the techniques described in the "How to make a Platform game" tutorial to add enemies using the sprites from the XNA Platform Starter Kit. You can also use the Gem.png file to create collectibles in the game.

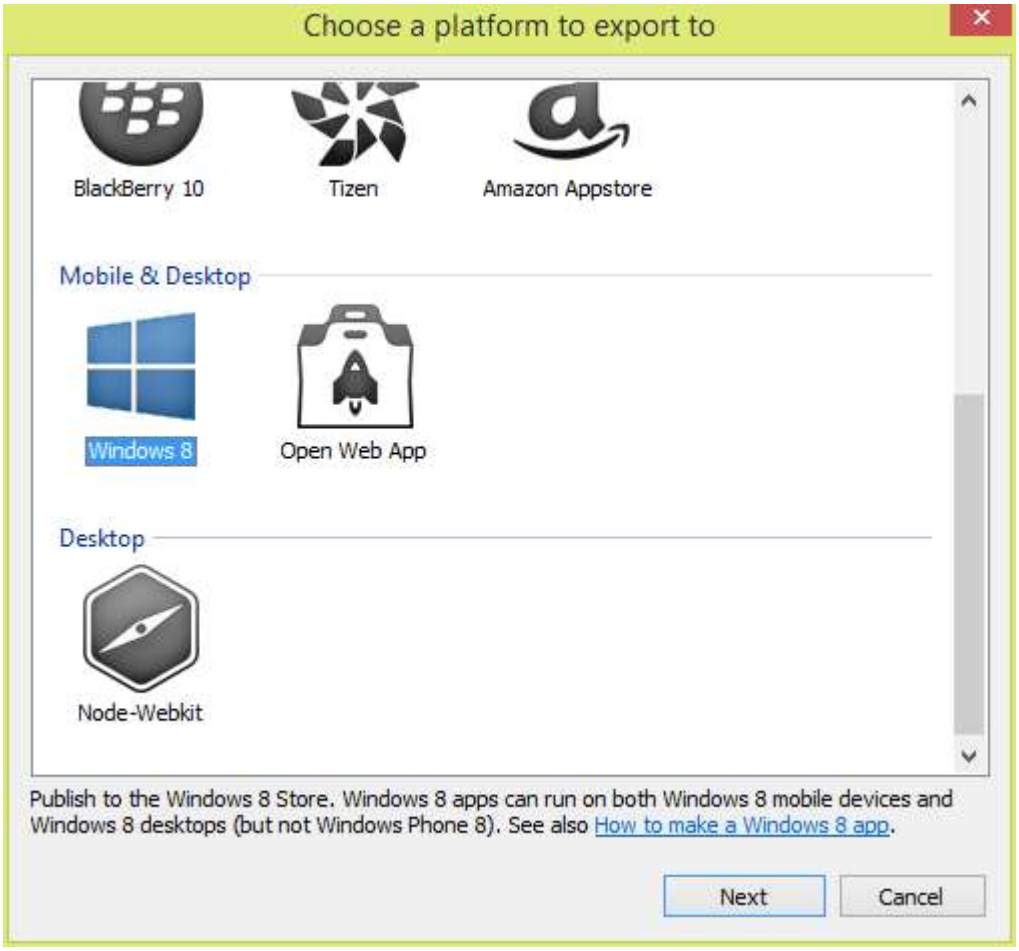
Exercise 4: Exporting Your Game

Once you've got your game in better shape, it's time to start thinking about publishing. When it comes to publishing your game for the Windows Store, one of the first steps is to export your game as a Visual Studio Windows Store app project.

Task 1 – Export Your Game

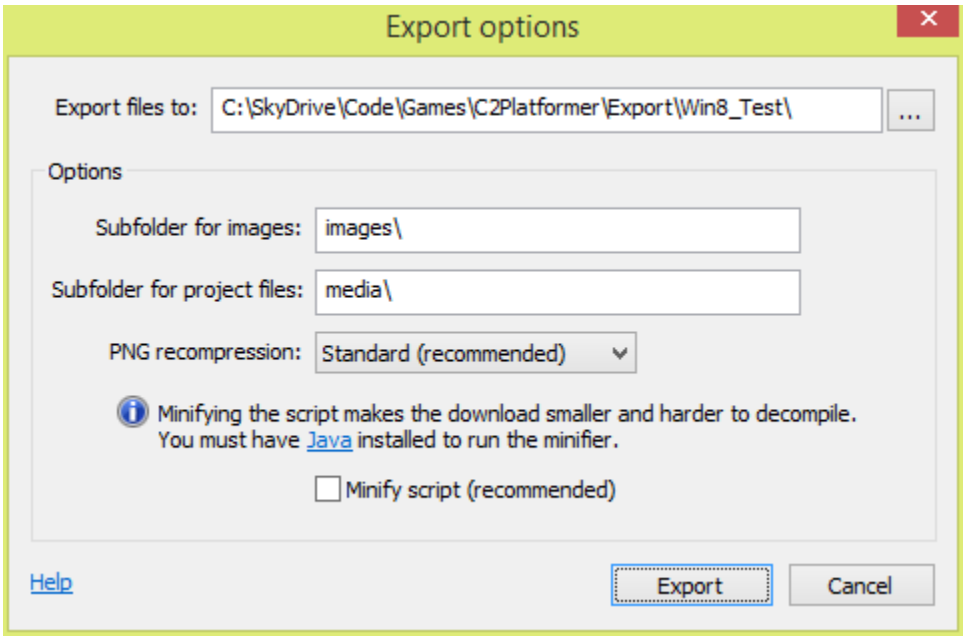
1. Save your .capx file before proceeding.

2. On the Home tab, click the Export project button, and scroll down and select the Windows 8 option, then click Next (if you have not set a description, version and author in the Project properties, you'll be prompted to do so before you can export):

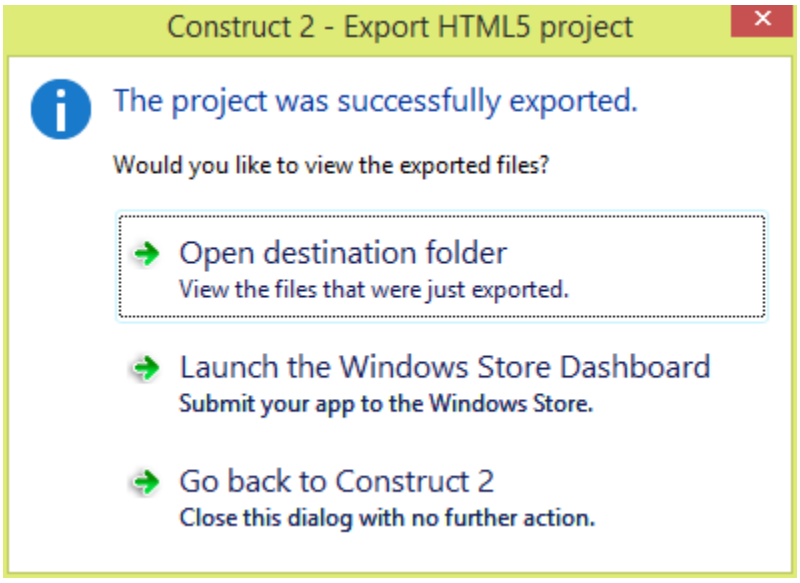


3. In the Export options dialog, select the location to export to, and make any desired modifications to the naming conventions for the subfolders. Note that if you do not have Java installed, you will not be able to minify scripts as a part of the export process (though you should minify your

scripts before publishing):

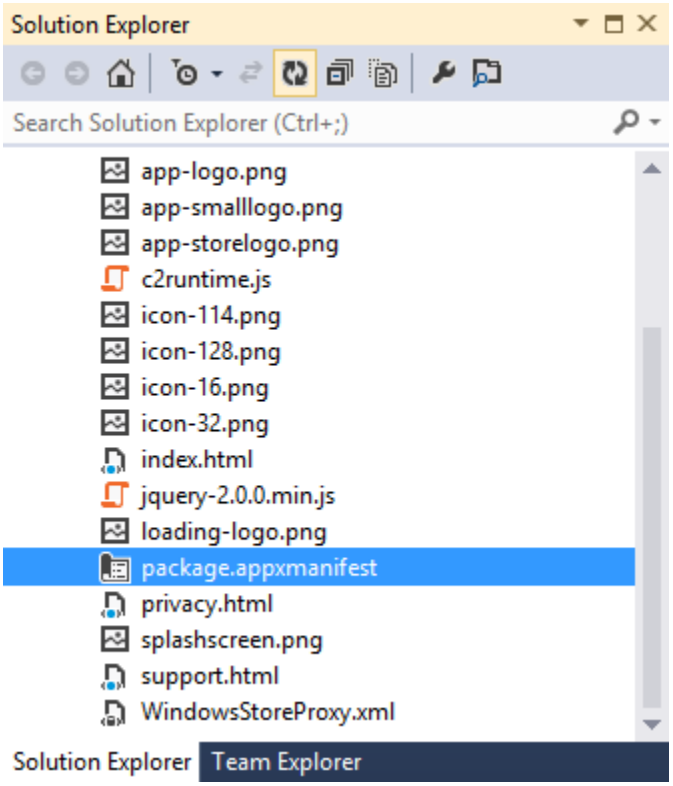


4. When you're done with the settings click Export, and the export will begin. When the export is complete, you'll be prompted with the following dialog:



5. It's a good idea to test the app, so go ahead and choose the first option, then open the .sln file created by Construct 2. If you have Visual Studio 2012 installed, this will open the project.
If you try to run the project now, it will fail with two errors related to the manifest file, which does not yet have all the information needed.

6. Open the package.appxmanifest file by double-clicking it in Solution Explorer:



7. Note that the Packaging tab has a red X indicating errors. Select that tab. Enter 0 in the Build and Revision boxes.

8. Click the Choose Certificate… button, then click the Configure Certificate… drop-down, and select Create test certificate…

9. Click OK in the Create Test Certificate dialog, then click OK to close the Choose Certificate dialog.

10. Save package.appxmanifest.

11. Now you can test the app by clicking the Local Machine button in the toolbar. The app should function just as it did in the browser, but running full-screen. Note that because the walkthrough did not update the game's Window Size property or its layout to match the usual 16x9 aspect ratio of a Windows 8 machine, the content will be letterboxed. Ideally, you should adapt your game UI to better fit the screen(s) it will be running on, and avoid letterboxing as much as possible.

Summary

In this workbook, you reviewed and modified the Construct 2 Platformer template, using assets from the freely-available XNA Platformer Starter Kit. You also learned how to export your project for publication in the Windows Store.

In the next Workbook, you'll take the next step by learning how to get a developer account and publish your game to the Windows Store!