

Create a New Game

Welcome Center

When you first load up Stencyl, you'll see a screen that looks something like this.



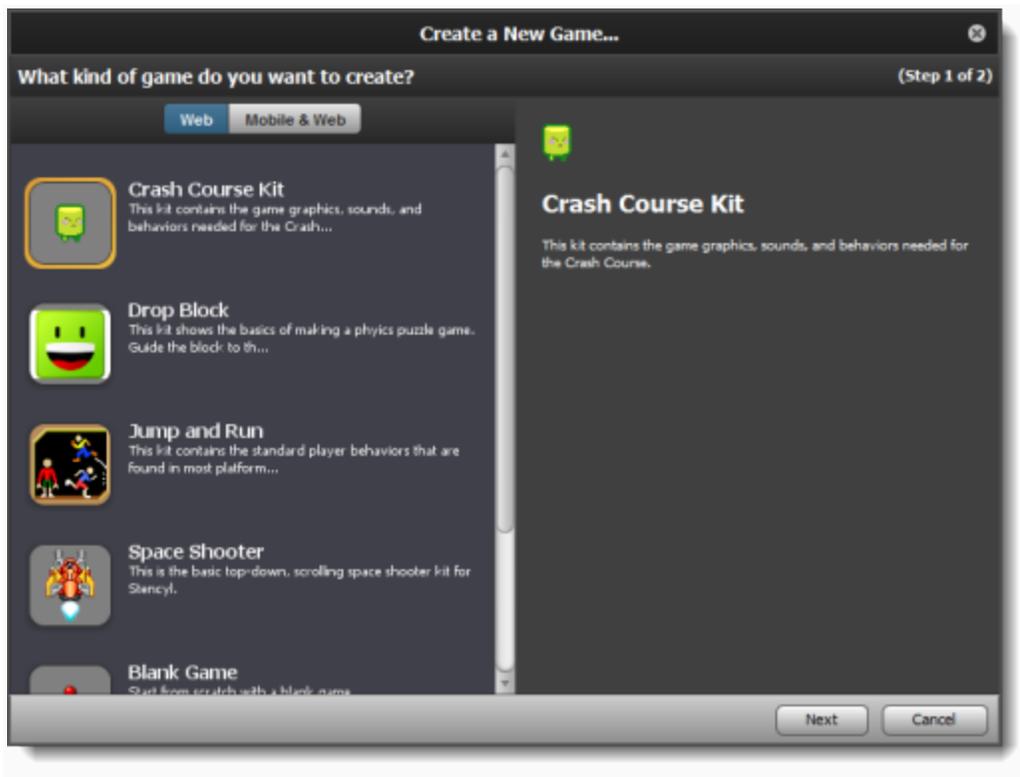
This is the **Welcome Center**. From here, you can create a new game, open an existing game, or browse games that other people have created.

Creating a New Game

1) Click the dotted square labeled **Click here to create new Game**.

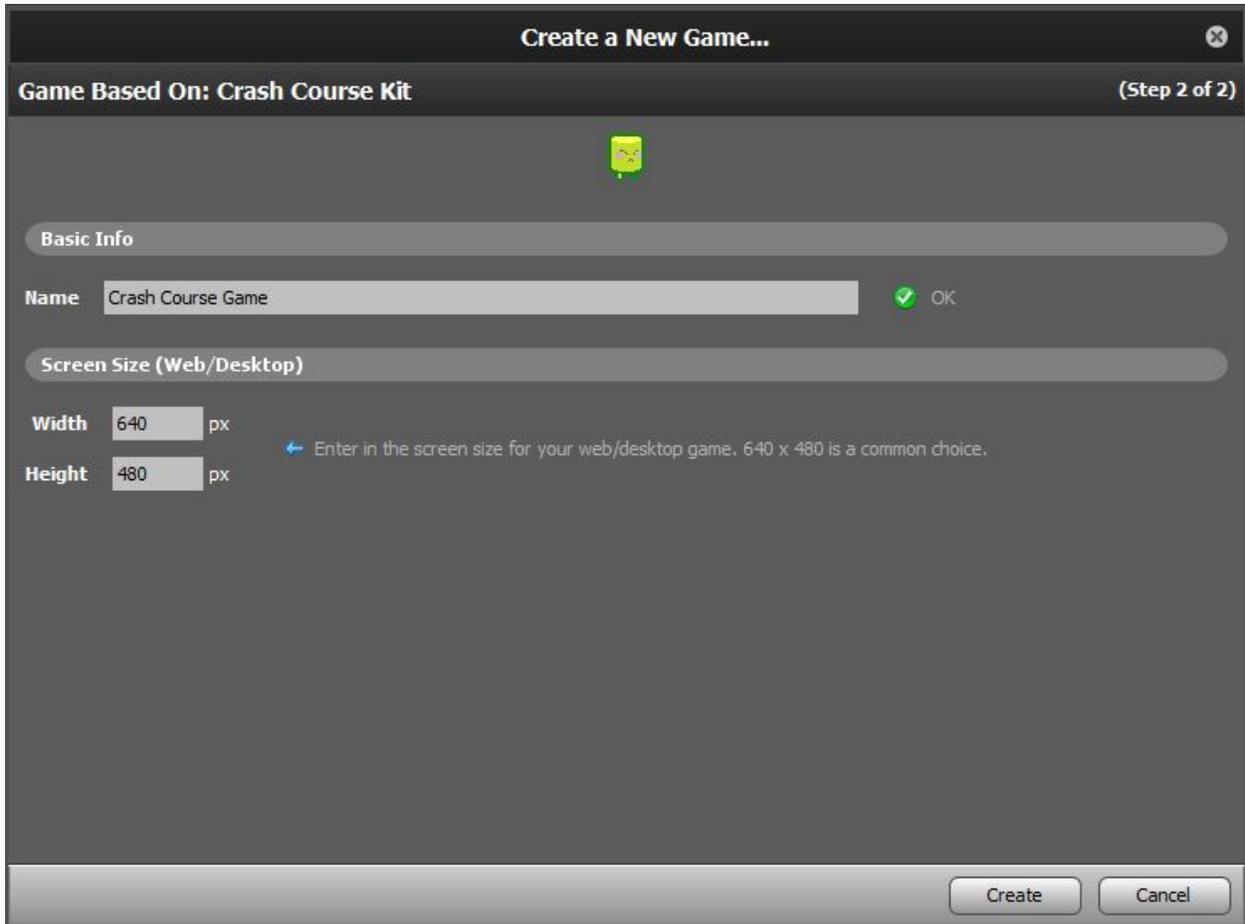


2) Click on the **Crash Course Kit**, then click the Next button at the bottom of the dialog.

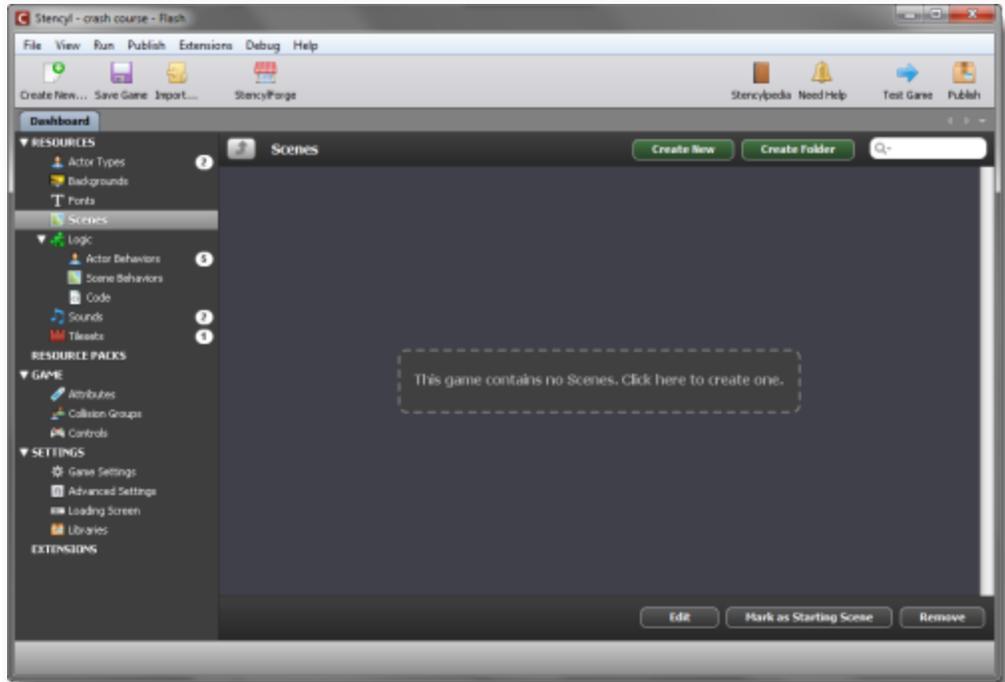


What are Kits? Often times when you create a game in Stencyl, you'll want to start with a Kit, a game template that comes with sample resources and has things like settings and game logic already configured. The kit you're starting with here has all the resources you'll need for the Crash Course.

3) Next, you'll see a dialog pop up (shown below) where you can set the game window's size (in pixels) and name your game. **We're going to name it Crash Course Game**, though feel free to name it something else if you'd like. By setting the **Width** and **Height** dimensions in the **Screen Size** section, you are determining the size of the window/view that the player will see when he or she plays your game. In this case, let's go with a Width of **640** and a Height of **480**.



4) You'll now be taken to the **Dashboard**, a central area where you can see your game's resources (graphics, sounds, game logic, etc.) and settings.



Game Resources

From the Dashboard, we can create new resources or import existing ones. For the Crash Course, we've included the resources you'll need to get started. Let's go over them.

Player Actor

We'll start by locating the **Actor** that will serve as our playable character.

Definition: In Stencyl, anything that can move or be interacted with is considered an **Actor**. This includes playable characters, enemies, user interface elements, etc. An **Actor Type** is a template for Actors, while Actor usually means a particular instance. Sometimes, we use the 2 interchangeably to keep our language simple.

1) First, click on the **Actor Types** entry in the Dashboard's left sidebar.

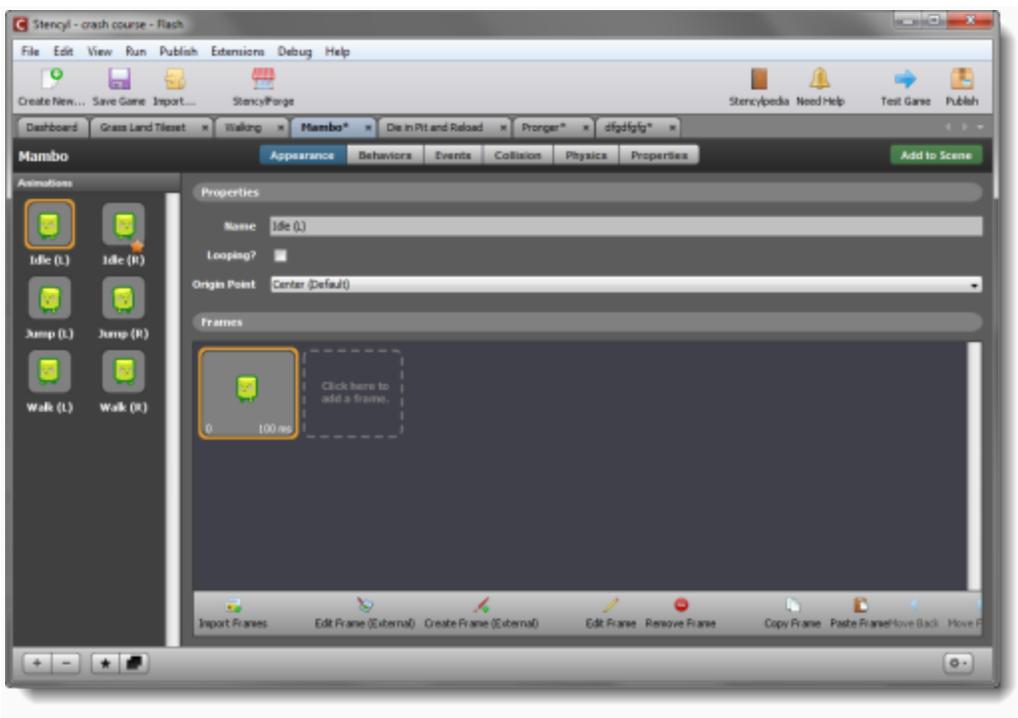


Note: You'll see a small number "2" next to the button. That number indicates the total number of actors in your game. Similarly, the numbers next to the other resources indicate the number of other types of resources we have.

2) Now, the **Actor Types listing** will appear. As expected, you'll see two Actor Types here, one called Mambo, the other called Pronger. Mambo is going to be our player's Actor.



3) If you **double-click** on the **Mambo** icon, Stencyl will open it inside the **Actor Editor**.

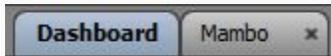


Definition: The Actor Editor is one of several resource editors inside Stencyl. It allows you to completely customize an Actor's appearance, behavior, and physical properties.

We'll come back to this editor soon, but for now, let's check out the rest of the resources we'll be using.

The Second Actor

1) Flip back over to the Dashboard by clicking on its tab.



2) Now open up Pronger, the actor we'll use as our enemy.



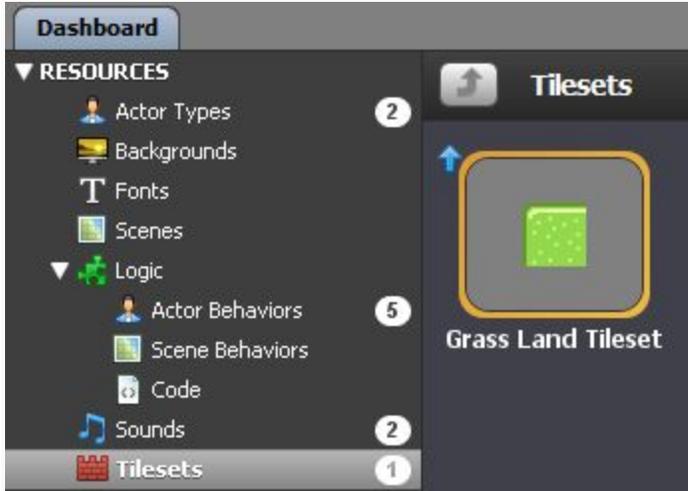
Tip: Prefer to open things up using the keyboard? Type Ctrl-O (or Command-O on Mac). This will bring up a dialog in which you can type the name of any resource. Use the arrow keys to fine-tune your selection and press Enter/Return to confirm your selection.

Tilesets

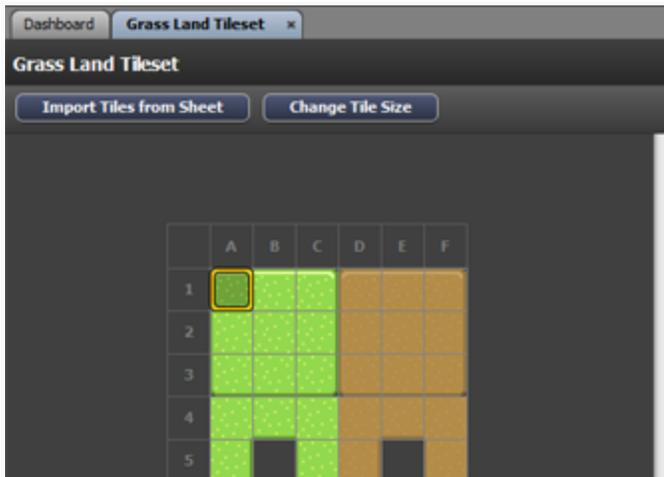
Now let's look at our Tileset.

Definition: A Tileset is a collection of rectangular tiles that can be used to build game levels (known as Scenes in Stencyl-speak).

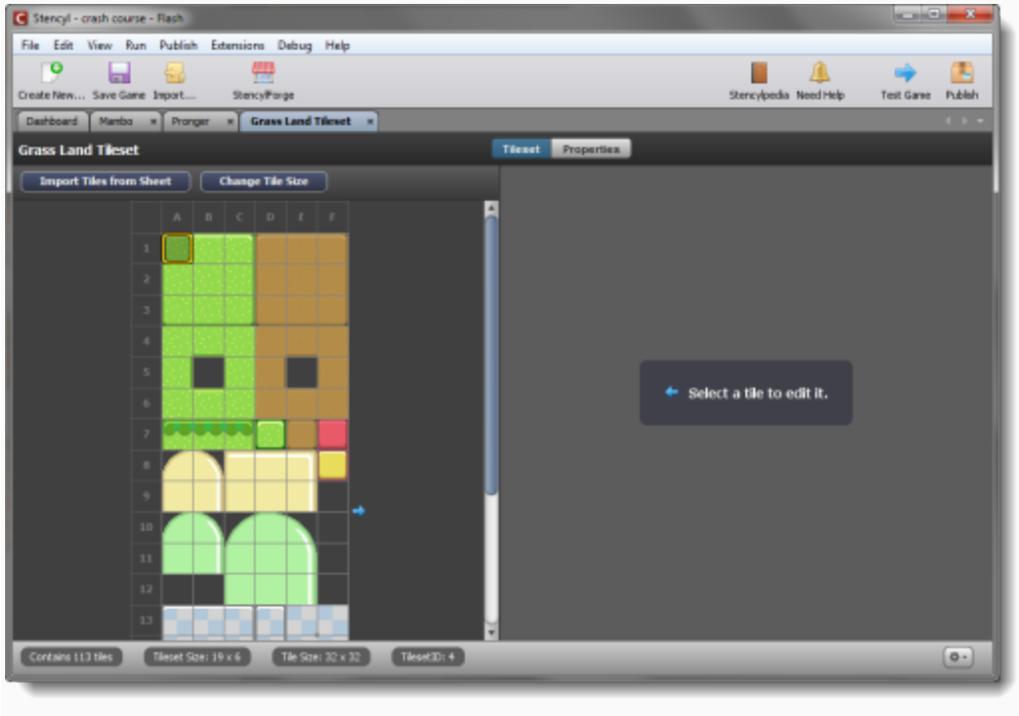
1) Click back to the Dashboard tab, and then click on the **Tilesets** category.



2) Open up the **Grass Land Tileset**.



3) As you'd expect, an editor pops up in a new tab. This time, it's the **Tileset Editor**.



We'll return to this soon after opening up a couple more things.

Sounds

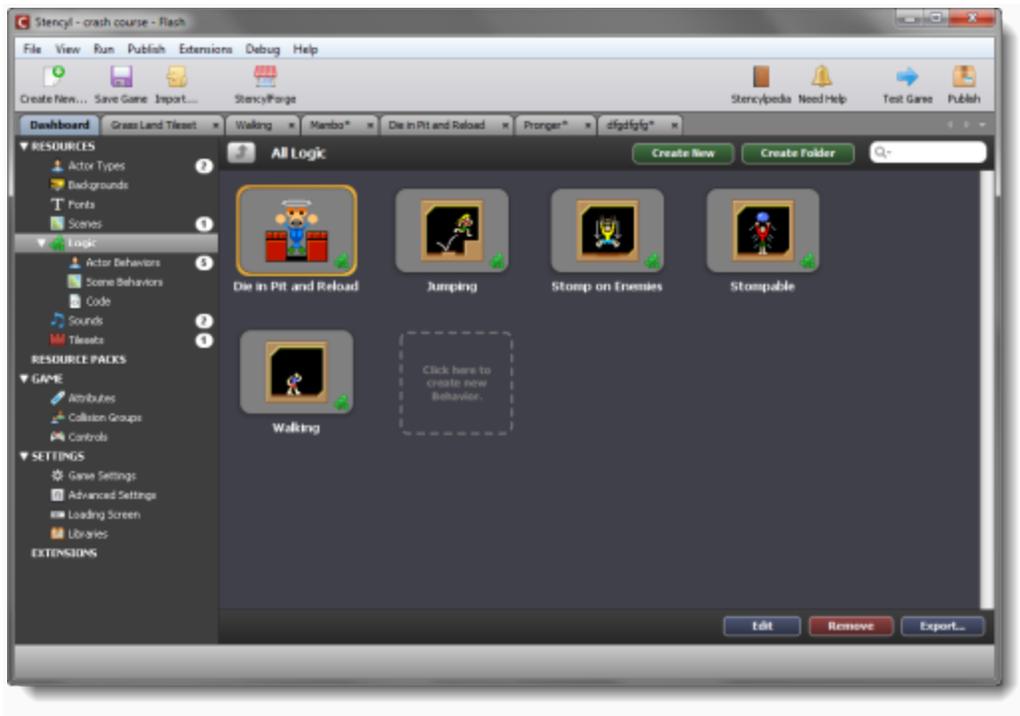
Let's look at our **Sounds** next. Click the Sounds button in the Dashboard. You'll see two Sounds are already there, Stomp and Jump.



Feel free to open up one of the sounds.

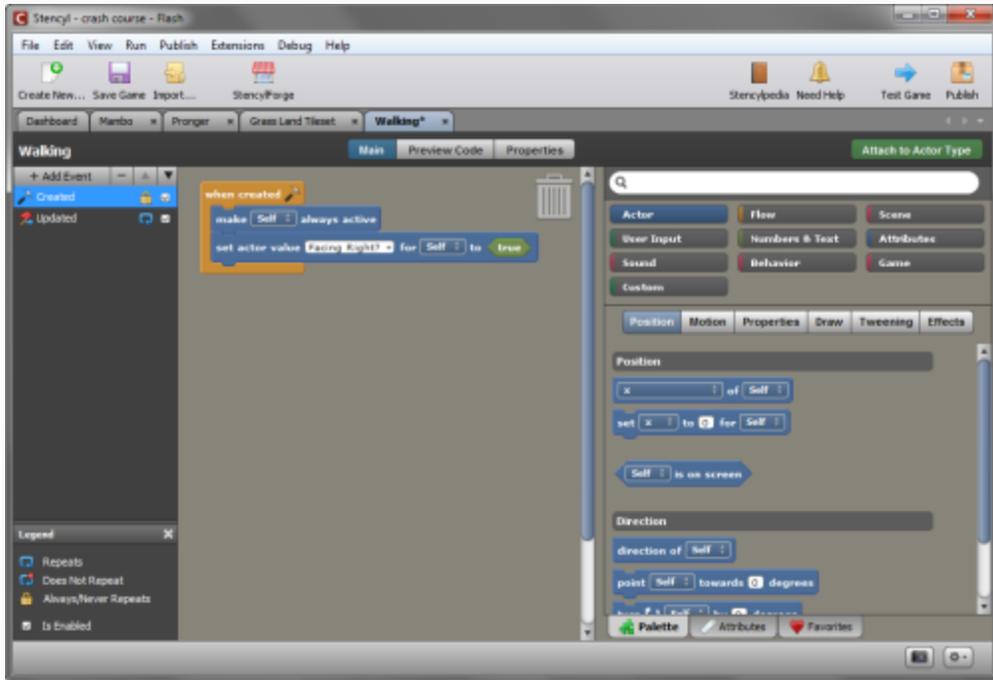
Behaviors

Last but not least, let's look at our **Behaviors**. Shown below are the Behaviors we'll be using in the Crash Course.



Note: Behaviors control all game logic and player interaction; they're what make every game "tick."

Let's just take a quick peek inside one of these Behaviors. Double-click on Walking to open it inside the **Behavior Editor**.

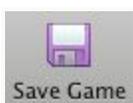


There's a lot going on here, and we'll talk about it further later on. The Behavior Editor is a powerful tool that makes designing complex logic quite straightforward. We have a whole tutorial dedicated to helping you learn the workings of this editor, though, so for now, just know that it exists.

Note: Prefer coding instead? We offer two ways to add code to your games, a dedicated Code Editor (or hooks to use your favorite text editor) and special Code Blocks that let you insert code into the Behavior Editor.

Saving

When you're working on your game, it's a good idea to save frequently. Just hit the Save Game button in the main toolbar to do so, or type Ctrl-S (or Command-S on Mac).



Customizing Actors

We've already imported some Actors into our game, but they aren't very interesting just yet; without Behaviors, Actors can't really do much at all.

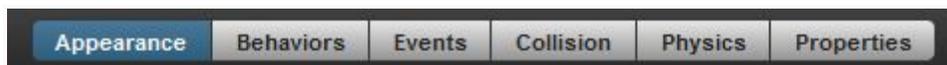
To **breathe some life** into Mambo and our enemy, Pronger, let's take a second look at the Actor Editor and **add the Behaviors** included in the Crash Course Kit.

Customizing Mambo

If you haven't closed the "Mambo" Actor yet, click its tab to select it. Otherwise, navigate to the Dashboard and double-click the Mambo Actor from the Actor Types part of the Library.



1) The familiar **Appearance** page of the Actor Editor appears. Skip over to the **Properties** tab by clicking its corresponding blue button at the top of the editor.



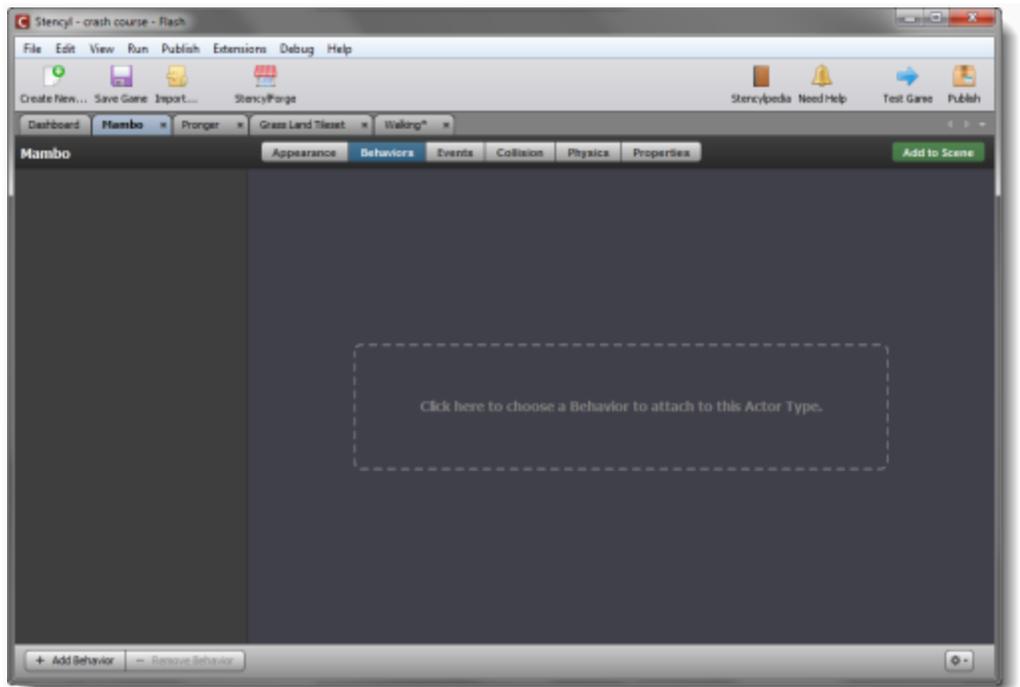
2) Check to see that Mambo is a member of the Players Group. This will ensure that Stencyl will handle collisions how we intend.



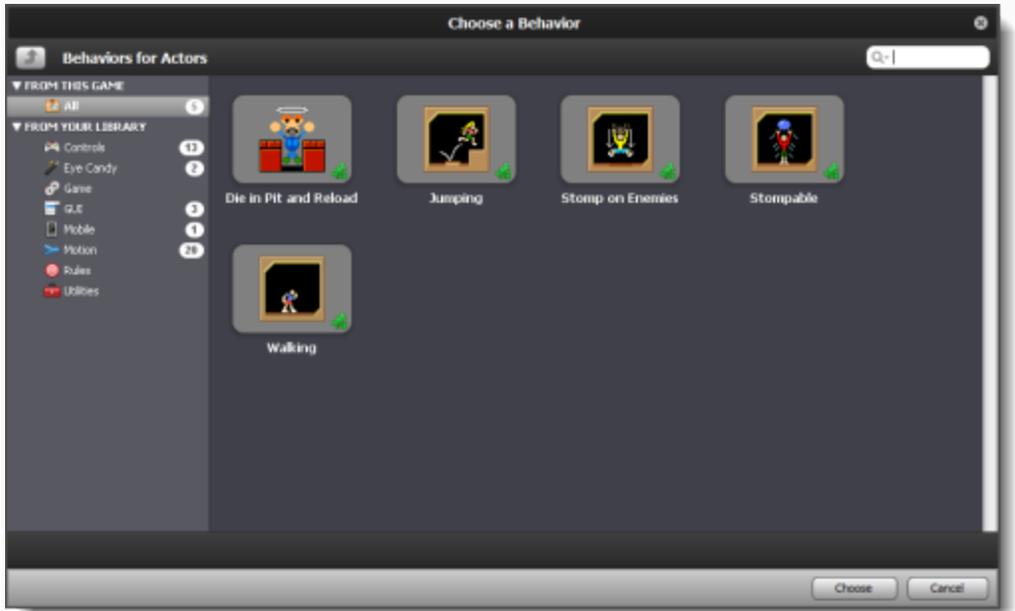
Definition: Groups let you categorize Actors in order to tell Stencyl what kinds of actors should collide with other. Groups can also let you treat different classes of Actors differently. [We've got an entire article devoted to Groups and how they affect collisions.](#)

3) Feel free to take a look at the Collision and Physics tabs as well. They contain additional settings that let you customize how Stencyl's physics engine treats the Actor, but for our purposes, the default settings will work just fine.

4) We're going to move on to the **Behaviors** tab, where the real customization begins. Start out by clicking the Behaviors button (right next to the Appearance button). The following screen will appear:

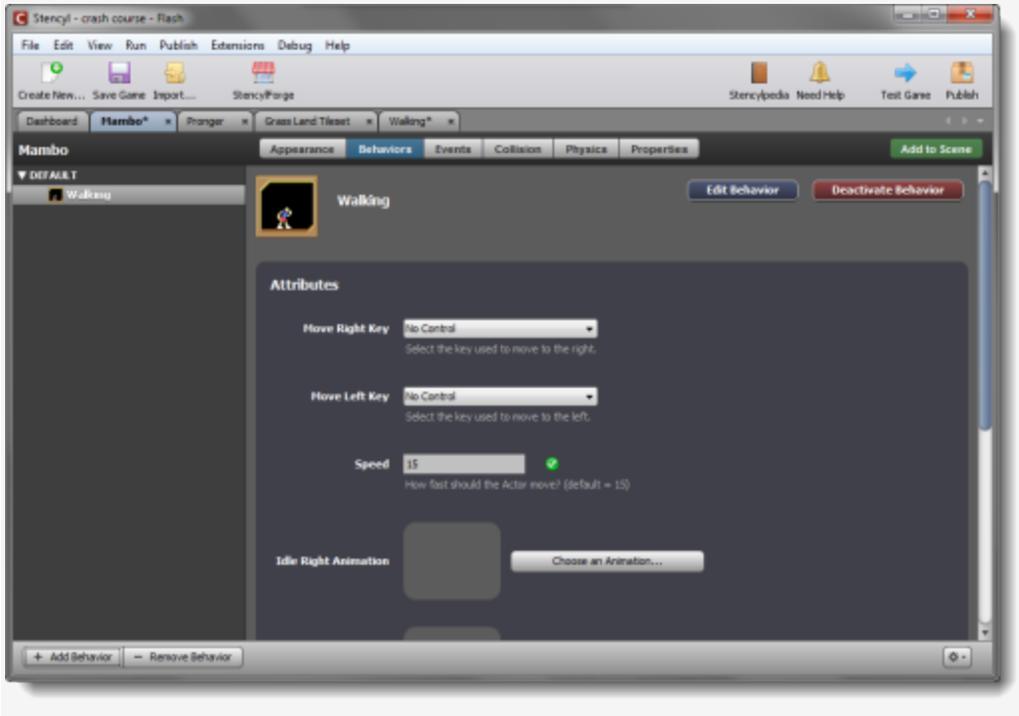


5) **Click on the Add Behavior button in the lower-left hand corner.** When the dialog appears, **select the Walking Behavior**, and finally click the Choose button in the lower right of the dialog to confirm your selection.



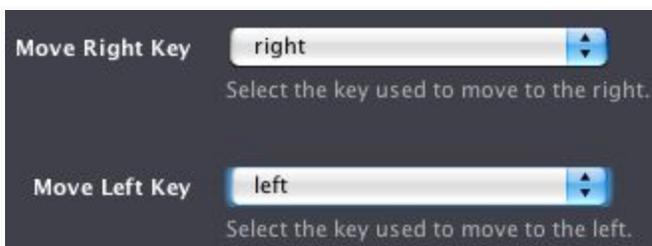
6) We're taken back to the Actor Editor. Notice the addition of the **Walking Behavior** in the list at the left, as well as all of its **Attributes** in the main window.

Definition: Attributes are customizable values that make Behaviors reusable and easy to modify. For more about Attributes, see our [introductory article about them](#).



7) Let's start **customizing these Attributes**. Some of the Attributes have default values that we can use, like **Speed**, whereas others we will have to set ourselves.

First, change Move Right Key and Move Left Key to the right and left **Controls**, respectively.



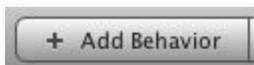
Definition: Controls map physical keys on the keyboard to names you can refer to in your behaviors. If you ever decide to change the actual key, you just have to change it once place. See our [Controls article](#) for more information.

8) Then choose the desired animations by clicking on the "**Choose an Animation**" button and selecting the animation sequences you want.

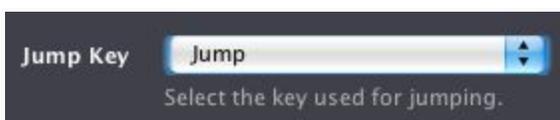


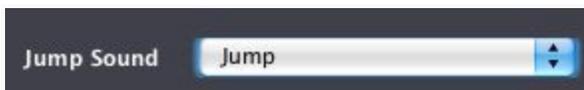
That's it for the first Behavior!

Let's add the rest and customize them in a similar way. To add more Behaviors, click the Add Behavior button at the lower left-hand corner of the editor.



- **Jumping:** Make the following modifications to the default values: Select the Jump key for jumping, and choose Jump (R) and Jump (L) for the Jump Right and Jump Left animations, respectively. Finally, add the Jump sound, so Stencyl will play a sound effect when Mambo jumps.





- **Stomp on Enemies:** Set the stompable group to Enemies and the jump key to Jump.



- **Die in Pit and Reload:** Nothing to configure.

Customizing Pronger

Pronger will be simpler to set up than Mambo. Switch over to his tab (or open him from the Dashboard, if need be).



- 1) Click on the **Properties** button as we did with Mambo, and find the **Group** dropdown.
- 2) Go ahead and change Pronger's group to Enemies.



- 3) All that's left to do with Pronger is to add a single Behavior. Click the Behaviors button and add the **Stompable** Behavior (located under the Collisions category). Customize the following two Attributes and leave the others as they are:



Note: As you can tell, this Behavior makes Pronger "stompable" like a Goomba in Super Mario Bros. Pronger will "die" when hit from above and play a sound when this happens. If you click the **Edit Behavior** button, you can peek at the "code" behind this behavior.

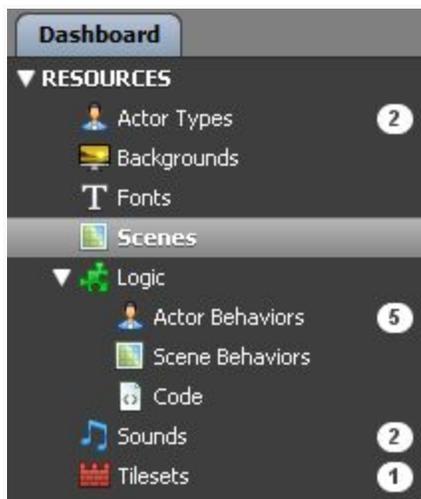
Creating a Scene

Now that we have our resources in place and our Actors configured, we can create a Scene.

Definition: Scenes are game levels that get populated with the tiles and Actors we've created. You can even attach Behaviors to Scenes, although we won't be doing so in this tutorial.

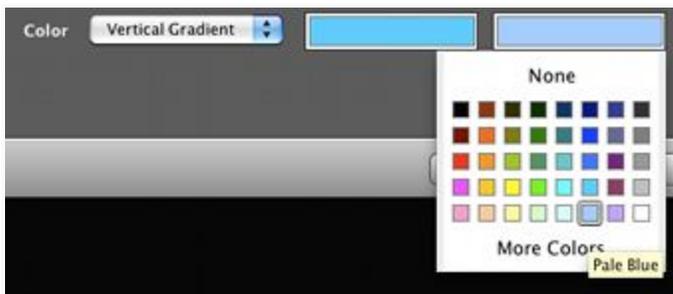
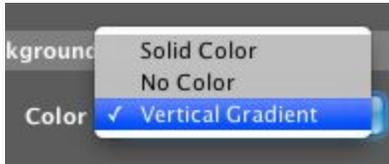
Creating a New Scene

1) From the Dashboard, click the **Scenes** category, followed by the large dashed rectangle.

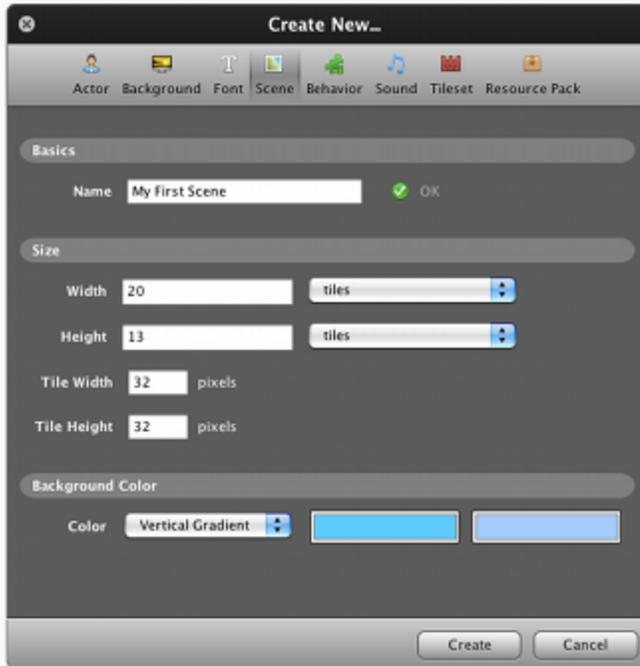


2) The Create New Scene dialog will appear. **Enter a name** of your choosing.

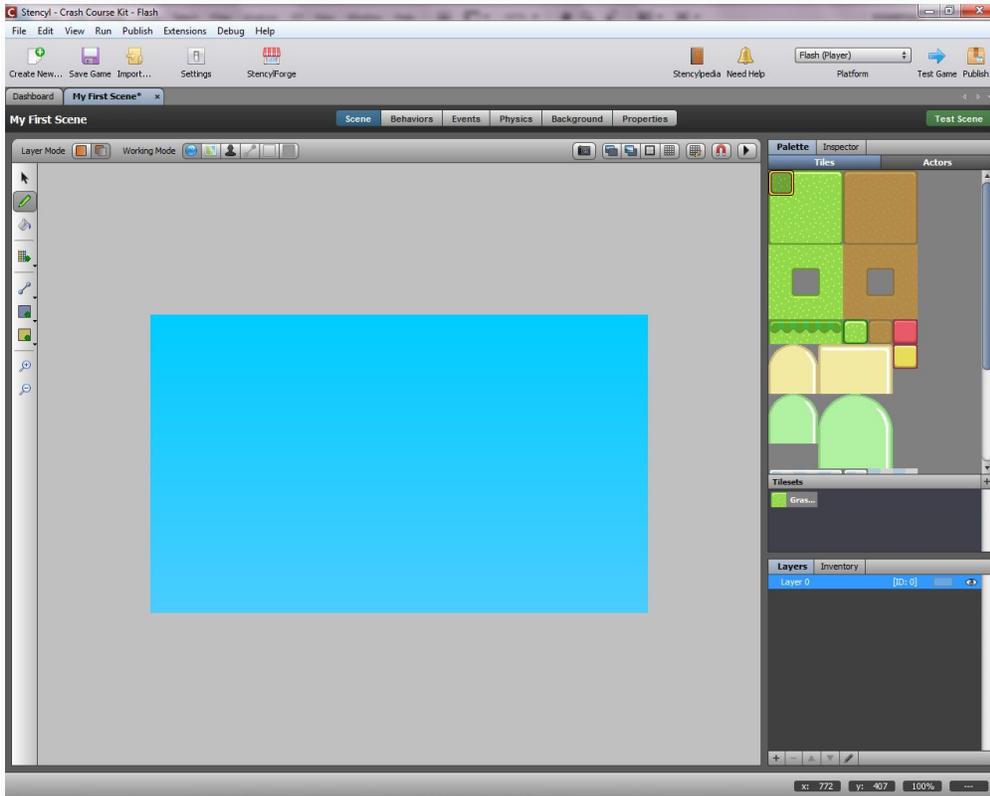
3) Let's have Stencyl generate a pretty sky background for us by clicking in the Color dropdown and selecting "Vertical Gradient." **Select two colors** by clicking inside the white boxes and then clicking on the desired colors. Note that the first color box on the left selects which color the gradient starts with at the top of the screen, and the second color box on the right selects the color that the gradient shifts to, moving toward the bottom of the screen.



4) When you're all finished, the dialog should look something like the picture below. **Click Create** when you're finished.



5) The Scene Designer will open. The interface may remind you of some popular painting programs and is just as intuitive to use.



Placing Tiles

Let's add some tiles to our Scene. They'll form the ground that our characters will stand on.

1) First, make sure that the **Pencil tool** is selected from the toolbar on the left-hand side of the editor.



2) **Click** on the *upper left-hand tile* in the Tiles section of the **Palette** (on the right side of the screen).



3) **Place** it in the bottom left-hand corner of the Scene by **left-clicking**.



4) Next, **select** the *middle tile* in the top row of the Palette.



5) With the left mouse button held down, **click and drag to fill** in the bottom row, leaving just a single spot in the corner.



6) Now **select** the *right-most* tile in the same row of the Palette.



7) **Place** it in the remaining empty spot of the Scene's bottom row.



Tip: When you select tiles to place, if you choose more than one tile in the Palette at a time, you can place the group of tiles you selected. See the screenshot below.



Placing Actors

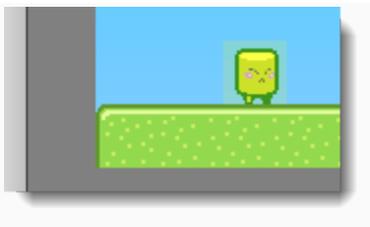
Now we get to add Mambo and Pronger to the Scene.

1) **Click on the Actors tab** in the Palette, and make sure Mambo is selected.



If you move your mouse over the Scene, you'll notice that Mambo follows the cursor.

Left-click near the ground to place him in the Scene.



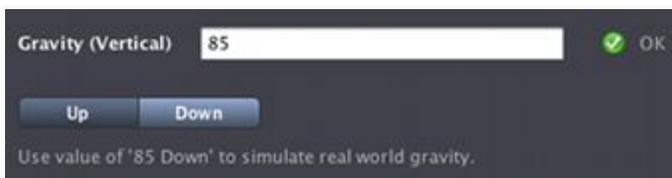
Tip: If you hold down the Shift button, you can snap the Actor to a grid.

2) Now select Pronger and place a few of him in the Scene as well. Your final Scene might look something like this:



Adding Gravity

Press the Physics button between the Events and Background buttons at the top of the screen, and in the **Vertical Gravity** section, type in "85" to simulate real-world gravity.



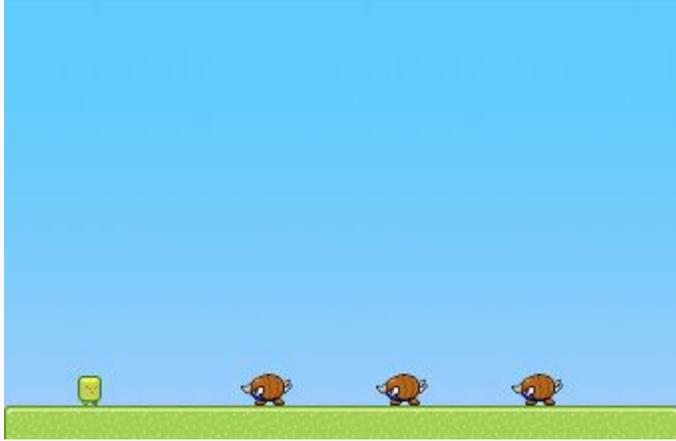
Test Your Game!

Click the **Test Game** button in the top toolbar to play your game.

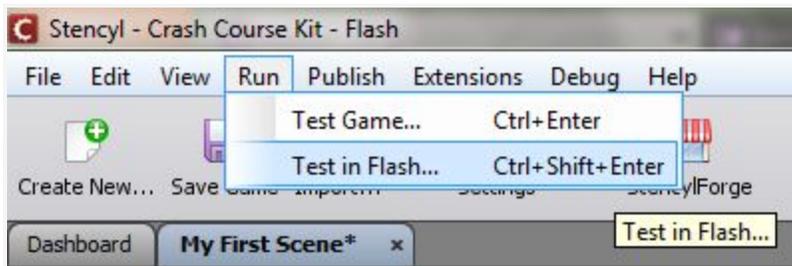


You can **walk** with the **left** and **right** arrow keys and **jump** with the **Spacebar**.

Try pushing the Pronger Actors around and jumping on them. If you fall off screen, the Scene will reload.



Note: You can also test your game inside of a web browser by choosing Test Game inside Browser from the File menu or pressing Ctrl-Enter (or Command-Enter on Mac).



Congratulations!

You've just created and tested your first game in Stencyl. That's quite an accomplishment already, and we've barely even scratched the surface of Stencyl's capabilities.

More Information!

For more information check out www.roughscience.com.au. Or drop a line to sean.m.elliott@gmail.com for future Rough Science and Making Games workshops.