



Welcome to the GameSalad Hour of Code activity! If you're familiar with GameSalad, feel free to jump right in to the Hour of Code steps on page 12.

If you're not familiar with GameSalad (or want a quick refresher), work your way through the tutorial from the beginning.



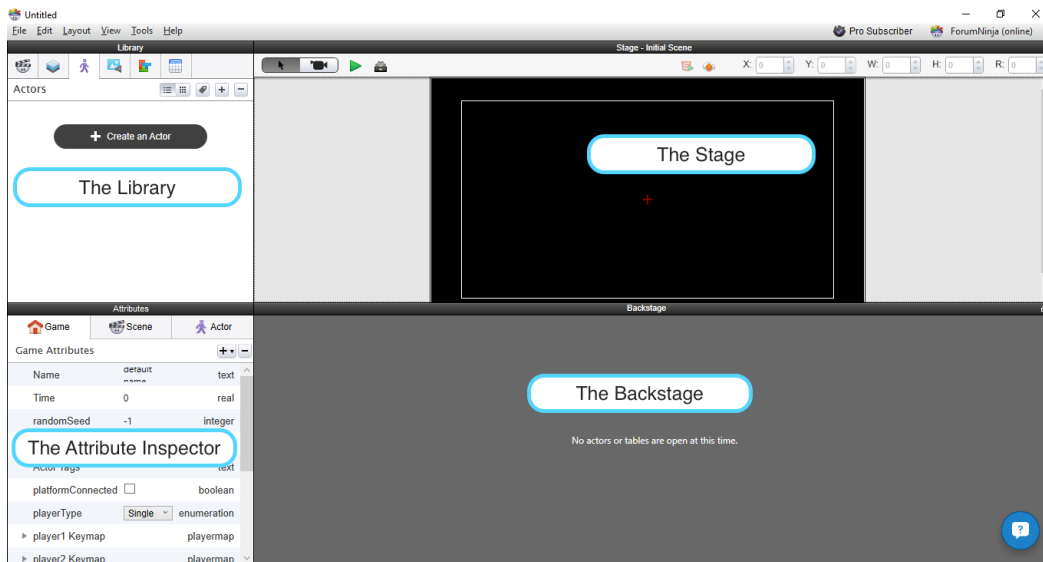
## Table of Contents

The GameSalad Interface .....	3
Key Concepts .....	4
Actors .....	4
Behaviors .....	5
Attributes .....	6
Scenes .....	8
Provided Actors .....	9
Hour of Code Steps .....	12
Adding Your Name to the Project .....	12
Previewing the Game.....	13
Creating Your Own Level!.....	15
Challenges.....	18
Publishing to the GameSalad Arcade and Sharing Your Game .....	19



## The GameSalad Interface

When you first open GameSalad, you will see the **GameSalad Game Editor**:



The Game Editor is split into four major sections:

- **Library**, where you can add actors, scenes, layers, assets, and tables, as well as see the list of behaviors GameSalad supports.
- **Stage**, where you can add actors to your scenes and position them.
- **Backstage**, where you will see and edit all the behaviors (logic) for your actors.
- **Attribute Inspector**, where you will create and modify attributes (variables) for your actors.



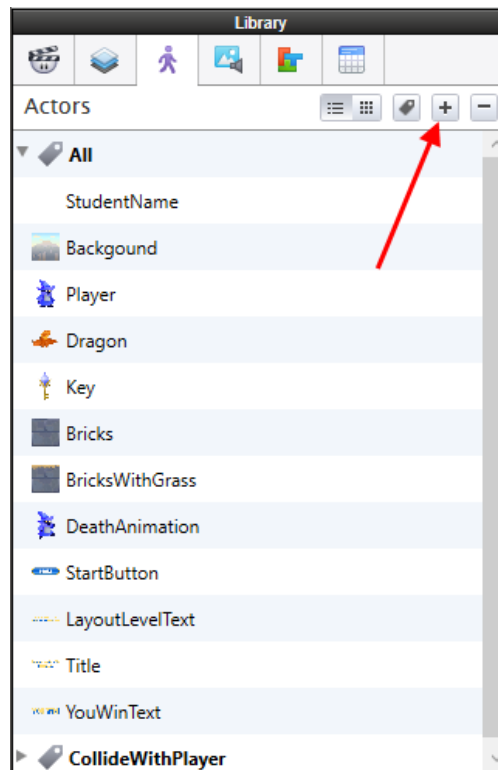
## Key Concepts

There's a few key concepts you'll need to understand to use GameSalad. Most notably, you'll need to know what actors, scenes, attributes, and behaviors are.

### Actors

Simply put, actors represent every object inside your game that the player can (or can't) see and interact with. In this hour of code game there's some already made actors such as the wizard that the player controls, the bricks that the wizard stands on, and the dragon.

You can add new actors by clicking the "+" button in the upper right of the Actors tab in the Library (the 3<sup>rd</sup> tab).





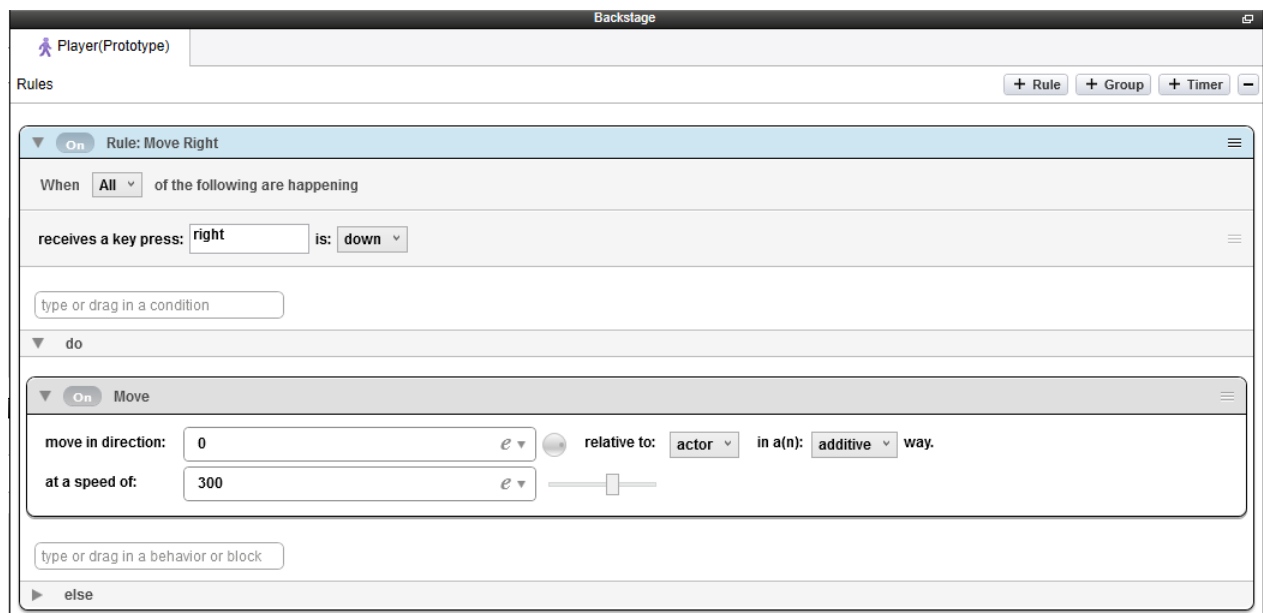
## Behaviors

Actors by themselves don't do anything! Behaviors are bits of logic that you can add to actors to make them do what you want. For example, if you wanted to have an actor move, you could add a Move behavior to it!



Rules are a special kind of behavior that hold other behaviors inside of it. Rules also have conditions (that you give them) that stop the behaviors inside them from running until something specific happens.

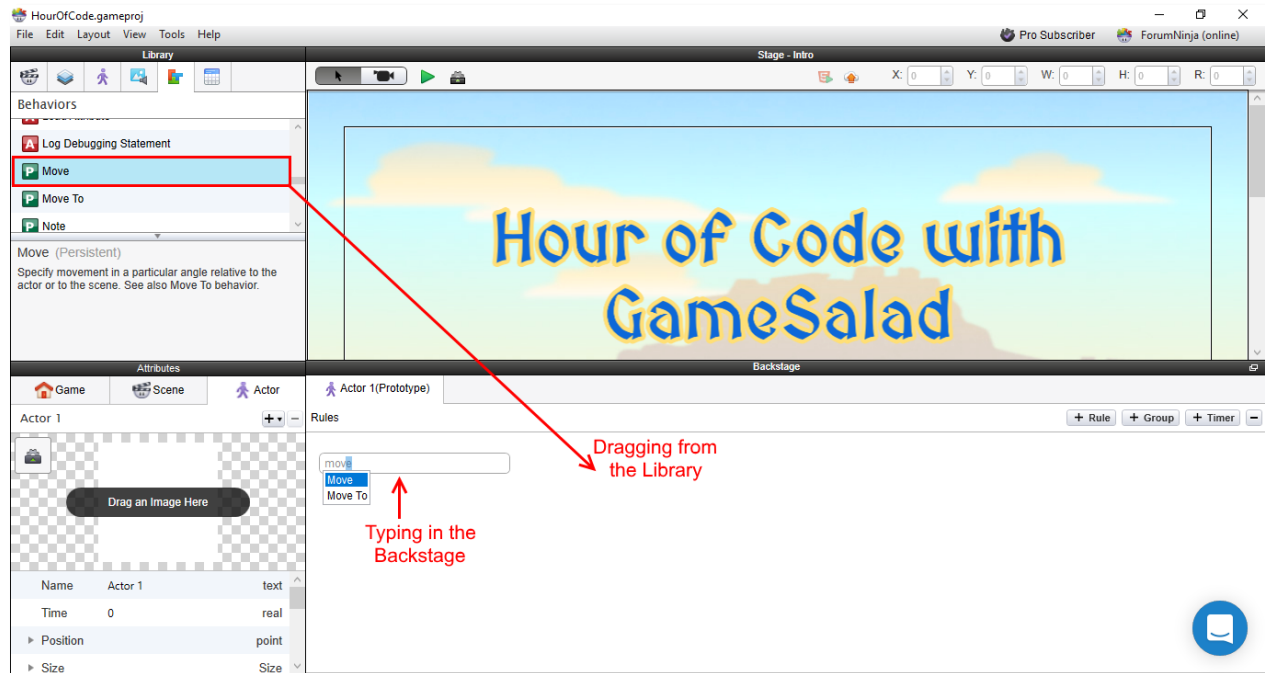
Using the same example as above, if you wanted to have an actor move only when the right arrow key is down, you'd need a Move behavior inside of a Rule with a "key" condition with the right key selected!





You can add behaviors to actors in two ways:

1. Typing the name of the behavior inside the field that's located inside the Backstage / rules.
2. Locating the behavior you'd like to add in the Behaviors tab of the Library (the 5<sup>th</sup> tab) and then dragging said behavior onto the Backstage.



## Attributes

Attributes inside GameSalad store all the data for your game just like variables do for computer programs! There's 6 different types of attributes that hold different types of values:

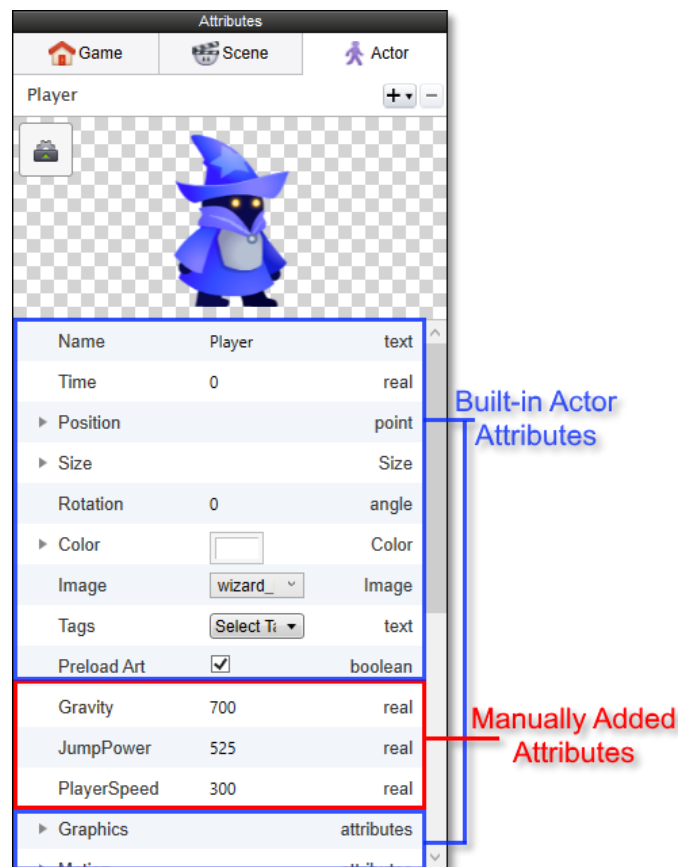
1. **Integer** – Integer attributes can hold all positive and negative, whole number values. Example integer values: 7, -23, 10.
2. **Real** – Real attributes can hold all positive and negative, whole and decimal values. Example real values: 2.6, -17, 3.14159265
3. **Boolean** – Boolean attributes can only hold the values true/false. You can think of boolean attributes as a switch that can only be on or off.
4. **Text** – Text attributes can hold text values. Example text values: "Hello World!", "h", "h-i-5".
5. **Index** – Index attributes can hold only positive, whole number values. Example index values: 5, 0, 200.



6. **Angle** – Angle attributes can hold values between 0 and 360. This attribute is best used for representing angles. Example angle values: 240, 90, 183.6.

Each actor inside GameSalad has a lot of built-in attributes that you can access, but you can create your own attributes too so the possibilities are endless! Some of the built-in attributes are an actor's width, height, and even physics attributes like an actor's density or velocity.

The last thing that's important to know about attributes is that there's both game attributes and actor attributes. Game attributes can be viewed by and changed by every actor inside your game, while actor attributes can only be viewed and changed by the actor they're on.



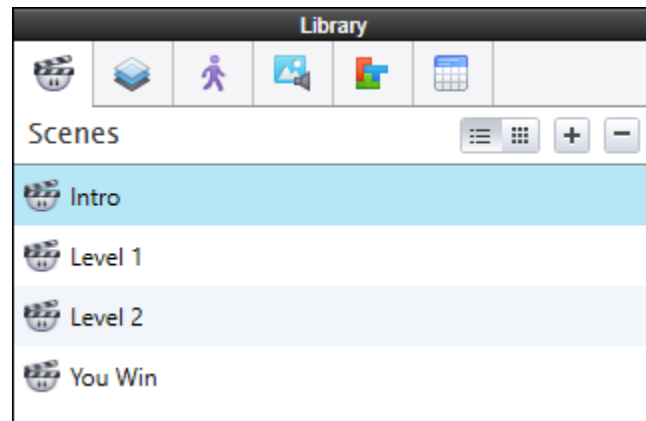
You can create your own game or actor attributes through the "+" dropdown button in the upper right of the Attributes Inspector.



## Scenes

The last major concept to understand is scenes. Simply put, scenes are what the player sees when playing the game. Scenes are where you add your actors so that the player can interact with them. You can think of a scene as an individual level for your game.

You can view your existing scenes and create new ones in the Scenes tab of the Library (the 1<sup>st</sup> tab). This is also where you'll select which scene you want to edit.



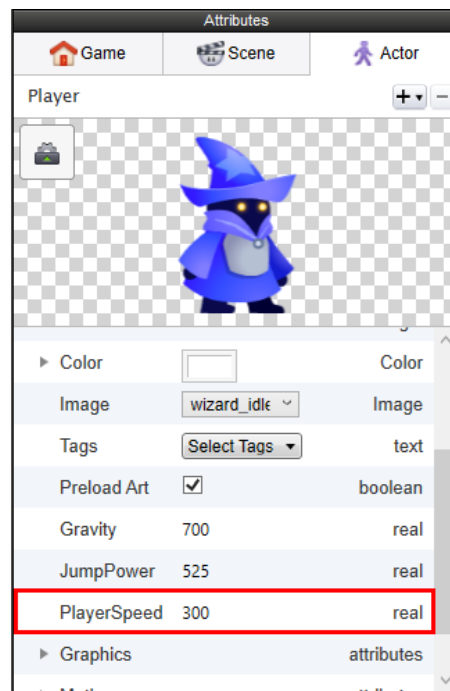




## Provided Actors

In the Hour of Code project provided, there's some premade actors already available that you can check out, edit, and use! The actors were created with custom attributes to make editing different aspects of them easy and intuitive.

For example, if you want to make the character you control move faster or slower, you can select the actor named "Player" in the Actors tab of the Library, and edit the "PlayerSpeed" attribute in the Inspector (you can edit attributes by double clicking their values).



1. **StudentName** – This actor displays the text "Created By: Your Name Here". that you see on the Intro scene of the game.

If you select this actor and look in the Inspector, you'll see an attribute with the text "Your Name Here" inside it. You can replace that text with your name to have it display on the Intro scene!

2. **Background** – This actor doesn't have any behaviors inside it, but it serves as the background image on each scene.
3. **Player** – This is the most complex actor in terms of behaviors. It moves, animates, plays sounds, and collides with other actors.



It moves through changing its built-in linear velocity x and y attributes, and changes the built-in graphics attribute “flipHorizontally” to make sure that the animation is facing the correct direction that the actor is moving.

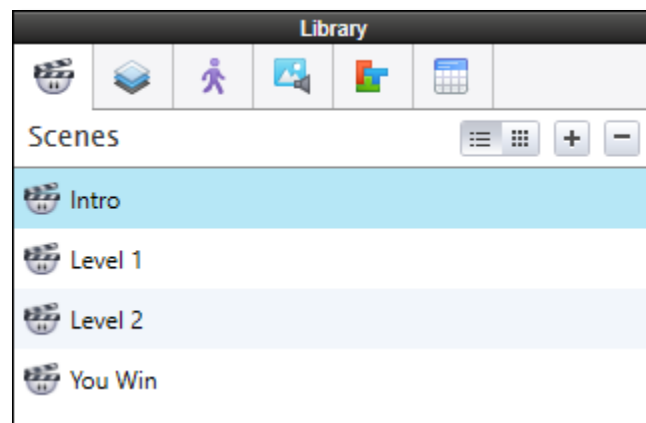
You can change the Player actor’s gravity, speed, and jump power via attributes in the Inspector.

4. **Dragon** – The dragon moves using the same behaviors as the Player actor, though it moves on its own instead of when a player presses specific keys.

You can easily change the distance the Dragon travels before turning around and the speed it moves at through actor attributes.

5. **Key** – The key plays a sound and changes the scene from whatever scene it’s currently on to the next scene based on the order in the Scenes tab of the Library.

For example, based on the following order, if you had a key on the Intro scene, and it collided with the Player actor, it would change the scene currently being displayed to the Level 1 scene.



Using the same example, if you moved the You Win scene above the Level 1 scene, when the collision occurs the scene would change to the You Win scene instead.

6. **Bricks** – This actor doesn’t have any logic inside of it, but some of its built-in attributes have been edited to allow it to tile the same image over and over when resized.
7. **BricksWithGrass** – This actor is very similar to the Bricks actor, though it only tiles when resized left to right.
8. **DeathAnimation** – When the Player actor collides with the Dragon actor it destroys itself and spawns this actor. This actor moves up off the scene while rotating, and restarts the current scene.



9. **StartButton** – This actor changes the current scene to the Level 1 scene when it's clicked.
10. **LayoutLevelText** – This actor has no logic in it, and just has an image to show the text on the “Level 2” scene.
11. **Title** – This actor has no logic in it, and just has an image to show the text on the “Intro” scene.
12. **YouWinText** – This actor has no logic in it, and just has an image to show the text on the “You Win” scene.

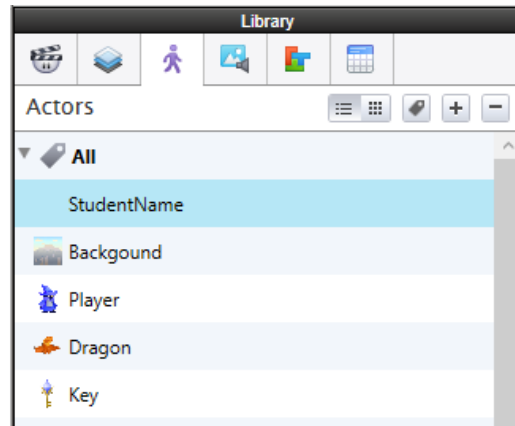


## Hour of Code Steps

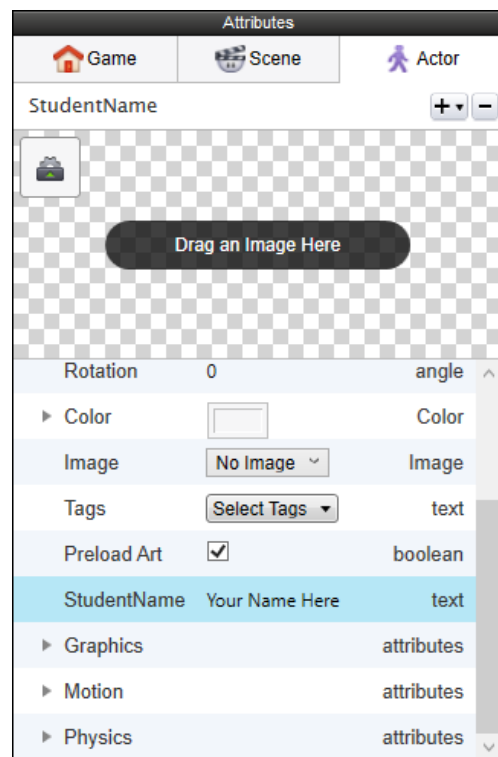
### Adding Your Name to the Project

First things first you'll need to open the supplied Hour of Code project.

1. Navigate to the Actors tab in the Library (it's the 3<sup>rd</sup> tab in the upper left with a little purple person) and select the StudentName actor.

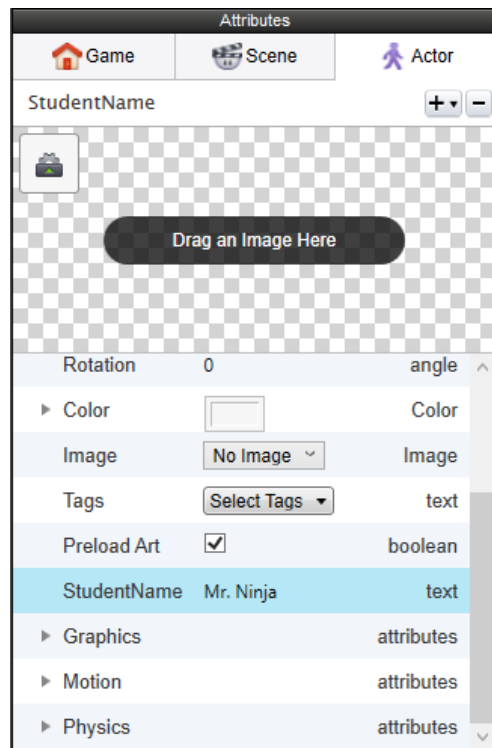


2. Make sure you're looking at the actor attributes (the Actor tab in the Attributes Inspector), then locate the StudentName attribute. It will have the text "Your Name Here" in it by default.



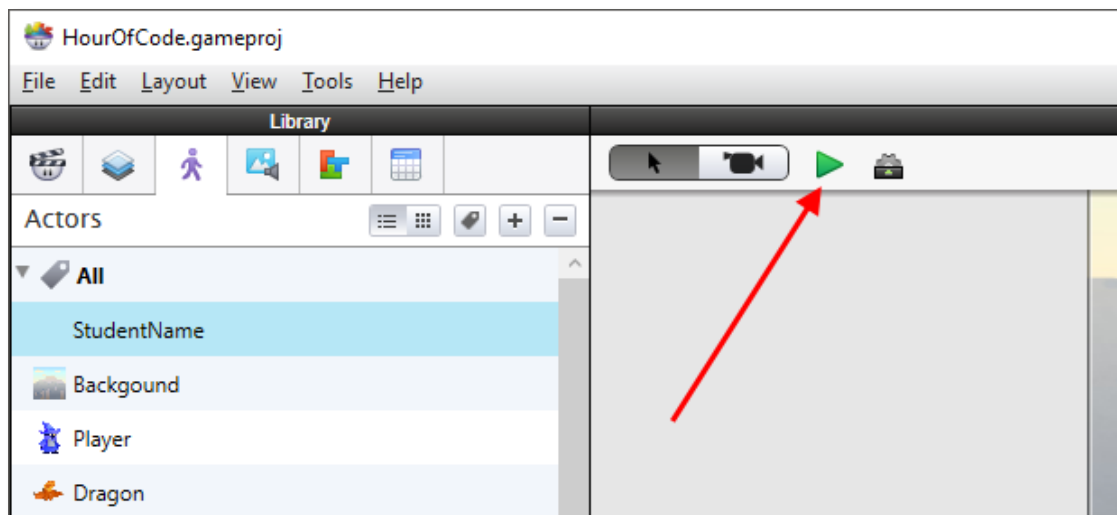


3. Double click the text inside the attribute (to edit it) and replace it with your name. I went with the name “Mr. Ninja”.



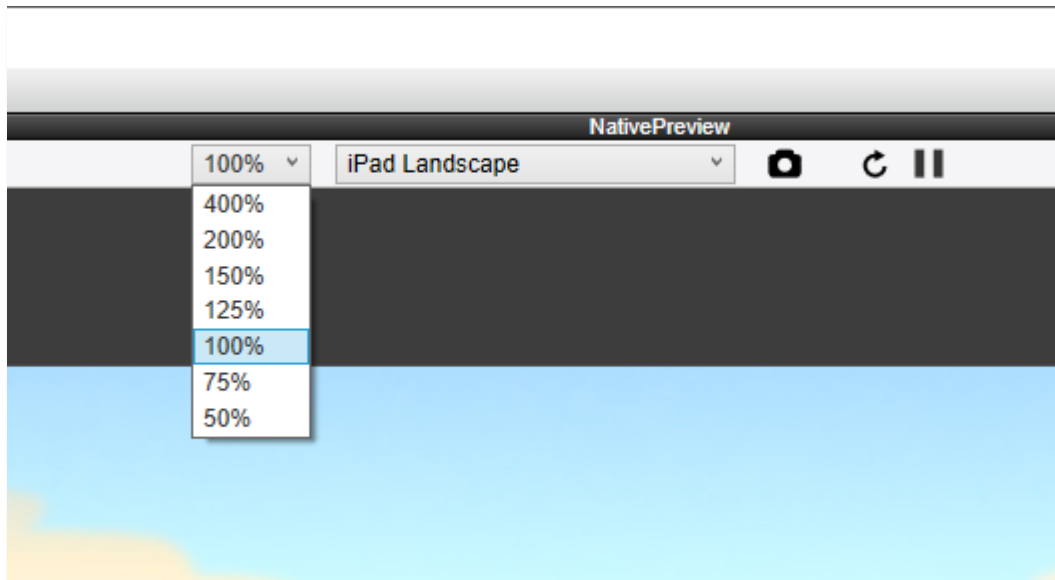
## Previewing the Game

Now that your name is filled in, go ahead and preview the project by clicking the green play button at the top of the editor!





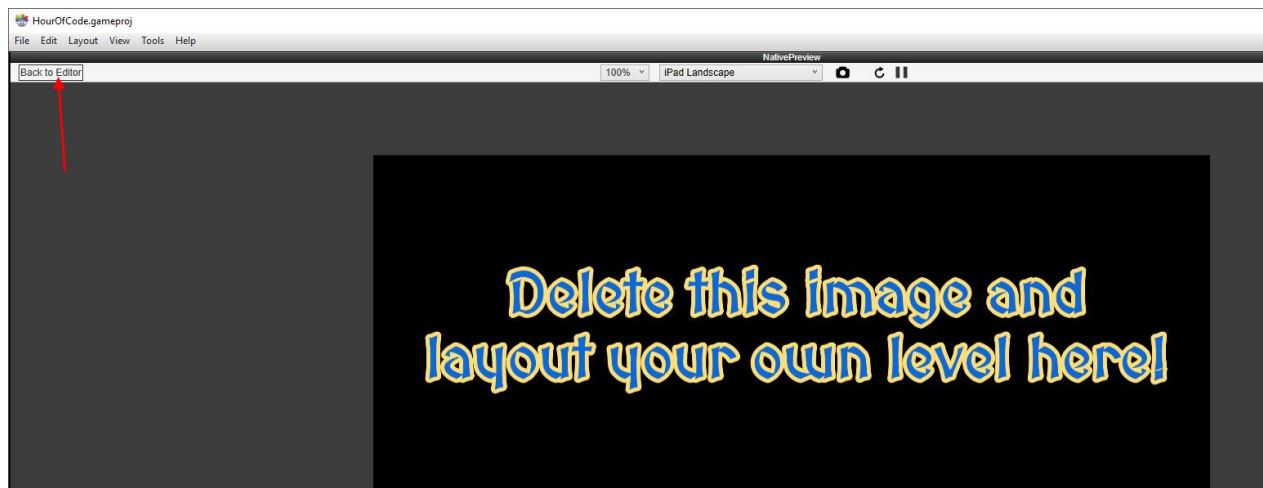
If the game is showing up a little bit too big for your screen, you can use the dropdown box at the top of the window to resize it and make it a little smaller.



You should see your name below the text “Hour of Code with GameSalad”. Go ahead and click the start button and play through the first level.

The game is set up to use the left and right arrow keys for movement, and the spacebar for jumping. Once you grab the key you should be taken to a new scene that’s blank except for the text “Delete this image and layout your own level here!”.

Click the “Back to Editor” button in the upper left to return to the editor.

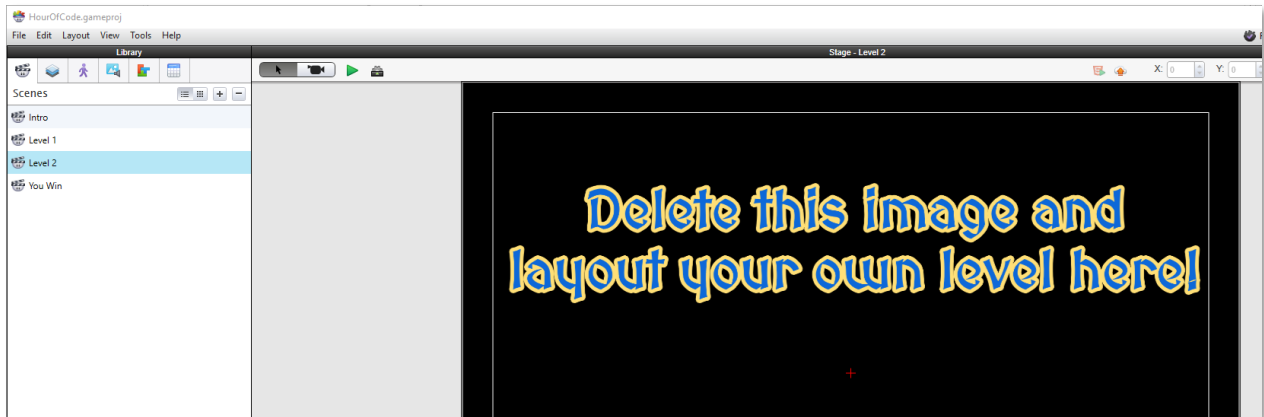




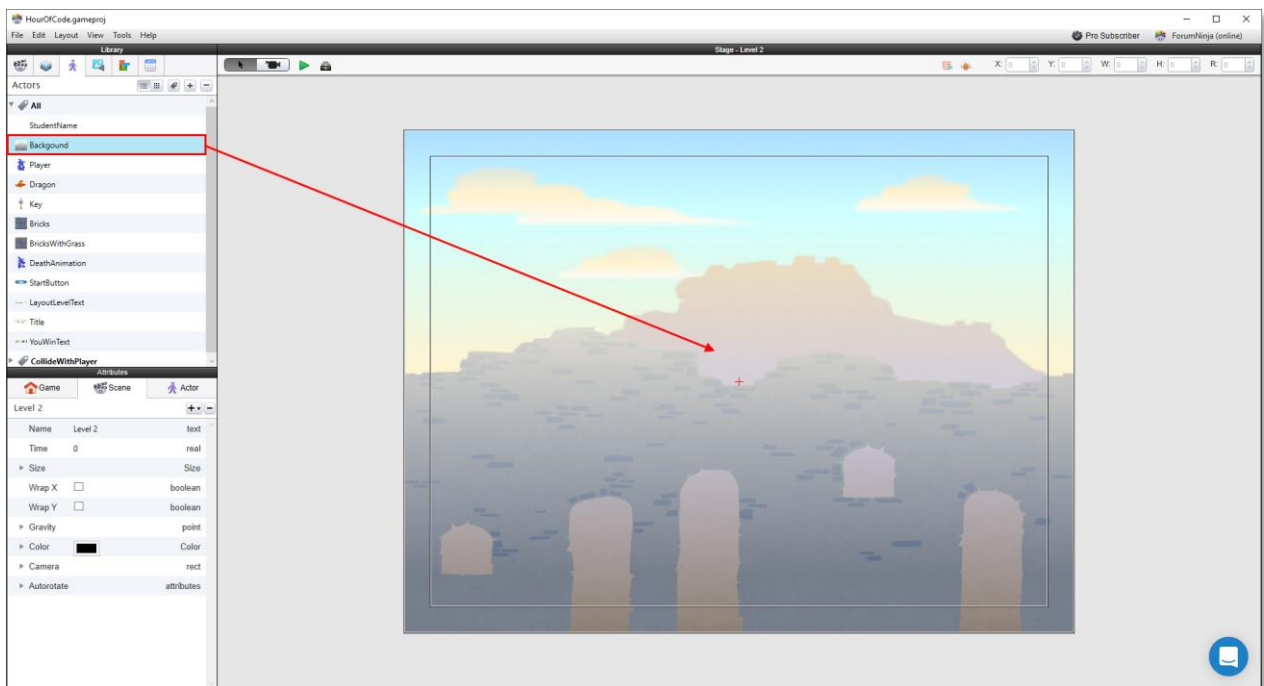
## Creating Your Own Level!

The next thing we'll need to do is get rid of that text we saw and put together our own level.

1. Navigate to the Scenes tab in the Library (it's the first tab) and select the Level 2 scene. You should see the scene that we were on when previewing appear in the Stage area.



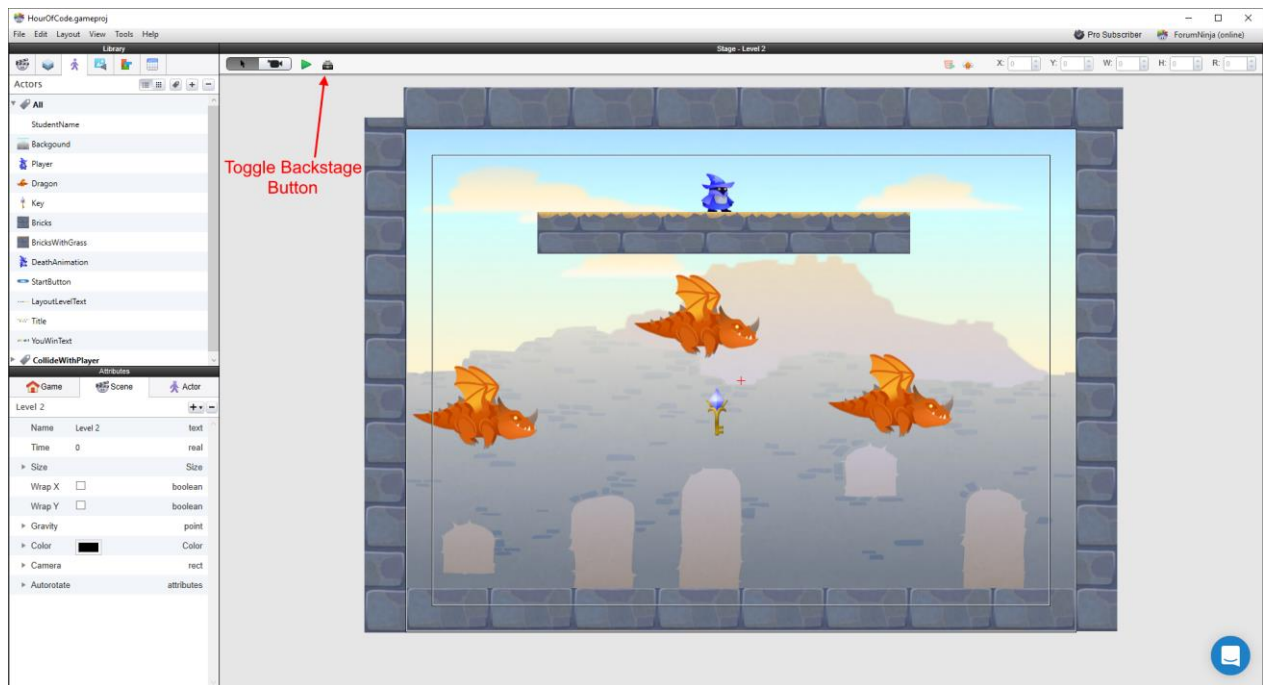
2. Select the text on the scene and delete it using your backspace/delete key so that you're left with a completely blank scene.
3. Select the Actors tab in the Library (the 3<sup>rd</sup> tab again) and locate the "Background" actor. Drag that actor onto the scene and position it towards the center.



4. Build out your own level by dragging, resizing, and positioning the various provided actors.



Here's what my level looks like. (Tip: You can toggle the Backstage on and off with the Toggle Backstage button that's located right next to the preview button)



Once you've laid out a level, preview the game again and make sure you can beat it! If you make it through your level and collect the key, you should see the You Win scene.

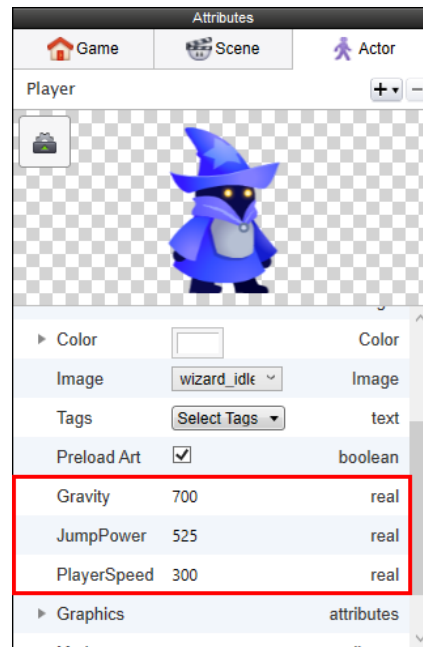






After you've created a level that's beatable and can make it all the way through the game, try editing the Player actor's attributes, adding more logic to actors, or creating your own actors to make the game unique!

There's 3 provided attributes that you can easily change that will greatly affect how the game plays, and those are Gravity, PlayerSpeed, and JumpPower.



When you're done playing around with the attributes, try taking on one of the challenges below!



## Challenges

### Challenge 1: Personalize Your Game

Find and modify assets (art and sounds) for your game.

- What theme will your game have?
- What will the main character be?
- What will your obstacles or collectables be?
- What sounds/music will enhance the theme of your game?

### Challenge 2: Make Your Game More Difficult

Make each level of your game more challenging than the previous level. Here are some ideas to get you started:

- Speed: How does increasing or decreasing the speed of your character change the level of difficulty? How fast is too fast to play? How slow is too slow to stay interested?
- Obstacles: What are the best spots for placing obstacles to increase the difficulty of your level? What size should your obstacles be in each level?
- Difficult Jumps: Can you create multiple platforms for the character to jump on? Can you place platforms in a way that will require more skill from the user?
- Longer Levels: You can increase the size of your levels by increasing the size of your level scenes. If you do this and give the Player actor a control camera behavior, you can design levels that are much wider or taller!

### Challenge 3: Keep Score

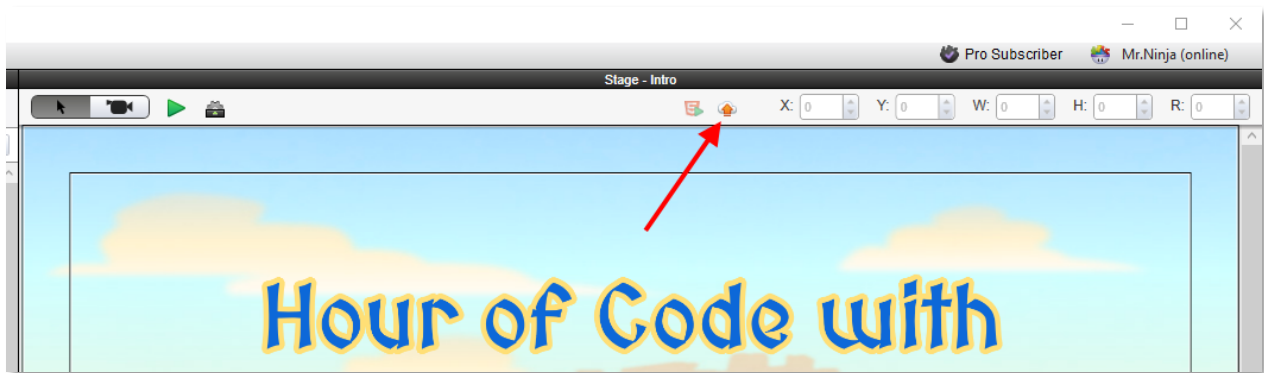
Can you add something for the player to collect along the way that increases a score value? Or perhaps you could give the player a score based off how fast they complete the levels.

- Create an attribute to keep track of the players score.
- How/where will you display the score?
- If you add collectibles, how many points will each item be worth?
- Can you make different types of collectibles that will be worth different point values?

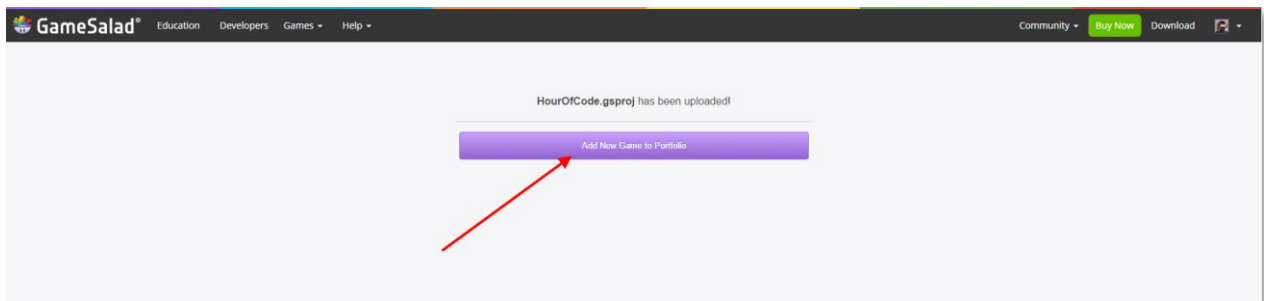


## Publishing to the GameSalad Arcade and Sharing Your Game

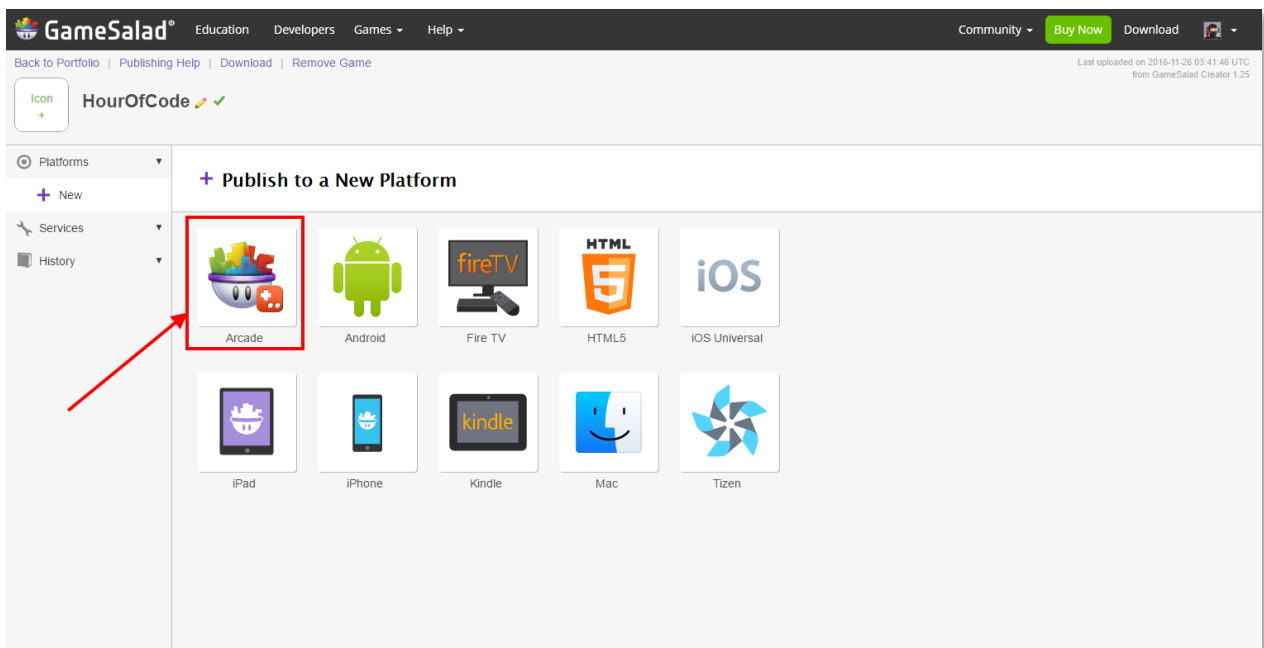
1. Click the publish button within GameSalad.



2. Click the add new game to portfolio button.



3. Select the Arcade platform.





#### 4. Select private game and add a description.

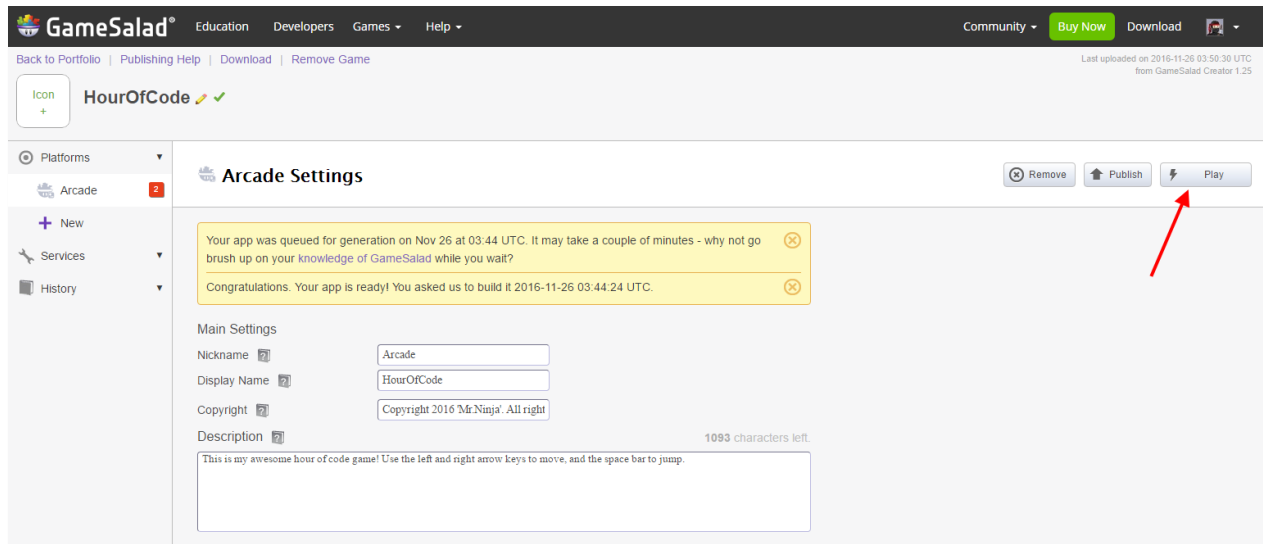
The screenshot shows the 'Arcade Settings' page for a game named 'HourOfCode'. The page has a sidebar on the left with 'Platforms' selected, showing 'Arcade' as the active platform. The main settings area includes fields for 'Nickname' (Arcade), 'Display Name' (HourOfCode), and 'Copyright' (Copyright 2016 'Mr.Ninja'. All right). The 'Description' field is highlighted with a red box and contains the text: 'This is my awesome hour of code game! Use the left and right arrow keys to move, and the space bar to jump.' Below the description field are three 'Screen Shots' placeholders, each with a '+ Image' button. The 'Visibility' section is also highlighted with a red box and shows the 'Private' option selected, with the text: 'Anyone with the link to this game can see it, but it will not be listed on arcade.' At the top right of the settings area are buttons for 'Remove', 'Publish', and 'Play'.

#### 5. Click publish button and wait. (you'll see a couple of notifications)

This screenshot shows the same 'Arcade Settings' page, but with the 'Publishing' button highlighted by a red arrow. A yellow notification box at the top states: 'Your app was queued for generation on Nov 26 at 03:44 UTC. It may take a couple of minutes - why not go brush up on your knowledge of GameSalad while you wait?'. In the bottom right corner, there is a 'GS Publishing Status Update' notification that says: 'Your app was queued for generation. It may take a couple of minutes - why not go brush up on your knowledge with GameSalad Cookbook while you wait?'. The 'Publishing' button is now disabled, and the 'Remove' button is visible.



- When your game is done publishing, the publish button will become clickable again, and the Play button will become clickable as well. Click play button to check out your game!



- After clicking the Play button, a new tab in your browser will open with your game playing on it. If you scroll down a little bit, you'll see a share link on the right side of the page. You can copy and paste that link to share your game with your friends!



- Make your game public, and add an app icon / screenshot to allow anyone to play your game.