KidzCoding Hour of Code - Platformer Game

Activity:

In this hour of code, students will learn how to make a **complex** and fun **platformer** game in **Scratch**. This game will use a large variety of concepts including: **loops, conditionals, variables, custom blocks, variables, broadcasting, and logic.** This game will consist of a ball that the player maneuvers around the screen to get to the next level. The player is affected by **gravity**, and they will have to jump their way across the platforms to get to the end of each level.

Learning Objectives:

* Learn how to use velocity and acceleration to control players
* Utilize variables to keep track of information about the current game state
* Develop an understanding of custom blocks and their importance
* Understand how to use broadcasting effectively
* Become comfortable with using logic statements with conditionals

Materials:

* Starter Project - <https://scratch.mit.edu/projects/876899568/>
* Completed Project - <https://scratch.mit.edu/projects/877127400/>

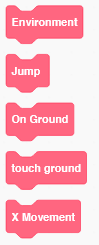
Coding Time!

Variables



This game will use 3 variables. The first variable, Is Jumping, keeps track of whether the player is currently jumping and in the air. If Is Jumping is 0, then this means that the player is not jumping and is on the ground. If Is Jumping is 1, then this means that the player is jumping and in the air. This will be useful because it will prevent the player from being able to jump when they are already in the air. The second variable, x-velocity, will be used to control how fast the player is moving side to side. The last variable, y-velocity, will be used to control how fast the player is moving up and down.

**Custom Blocks**

****

This game will use custom blocks to control everything.

Environment: This block will control the basics of the game like how gravity works and how the player moves.

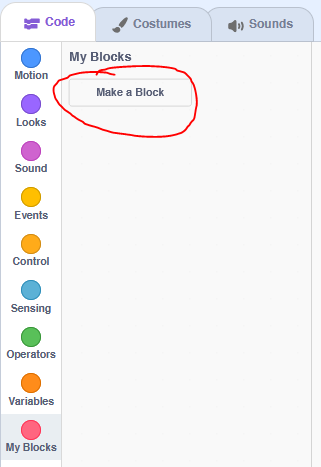
Jump: This block will control how the player jumps

On Ground: This block will control what the player does when it is on the ground.

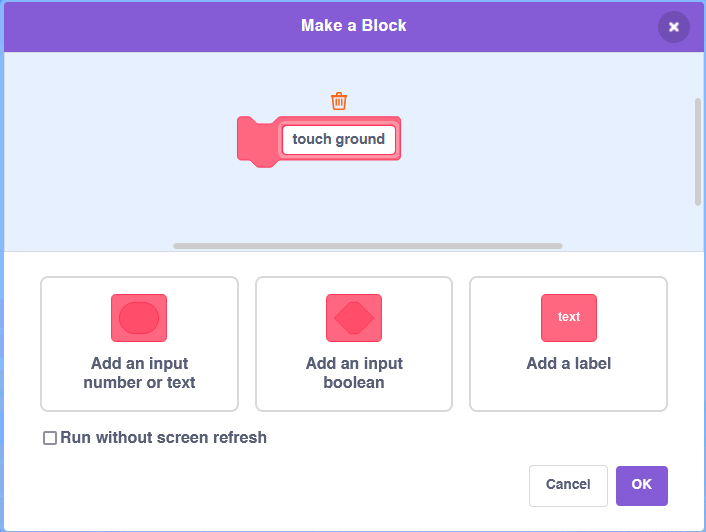
Touch Ground: This block will make the player touch the ground correctly. It makes it so that the player doesn’t fall through the ground and the platforms.

X-movement: This block allows the player to control the ball side to side.

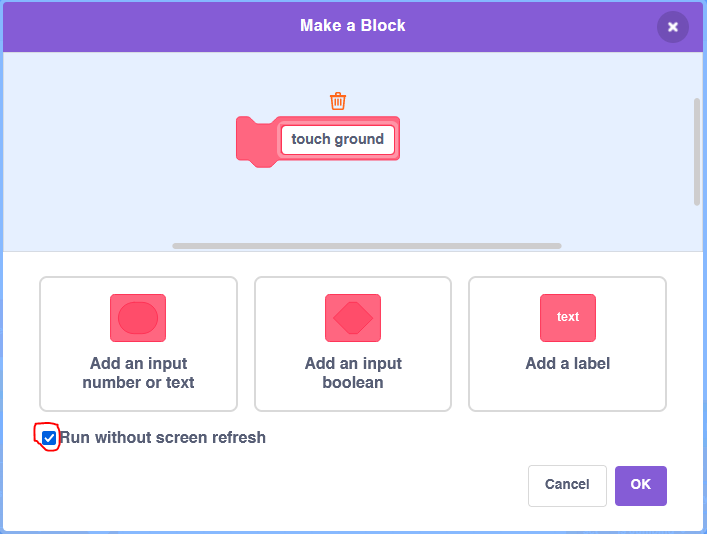
**1. Creating the “touch ground” block**

****

First, scroll down to the my blocks area. Then, click on the “Make a Block” button at the top.

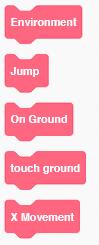


Second, type in “touch ground” in the box.



Last, make sure that the “Run without screen refresh” box is checked. Just click on the box if it is not checked. After this, just press OK and then your custom block will be created!

**2. Creating the Other Blocks**

****

Repeat the process that we just did with the “touch ground” block to make the Environment block, Jump block, On Ground block, and X movement block. However, **MAKE SURE THAT “RUN WITHOUT SCREEN REFRESH IS NOT CHECKED THIS TIME.**

**Coding the Player**

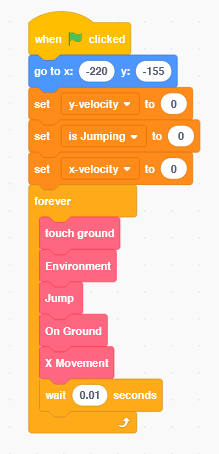
****

**1. Setting Up the Game**

****

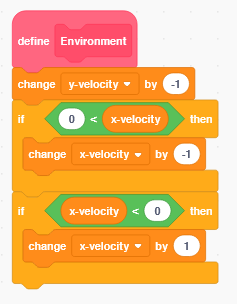
At the start of the game, we need to put the player at the starting position with the “go to” block. This will put the player in the bottom left corner. We also need to set all the variables to 0.

**2. Creating the Main Loop**

****

For this step, simply add a forever loop and insert all of the custom blocks that we made.

**3. Creating the Environment**

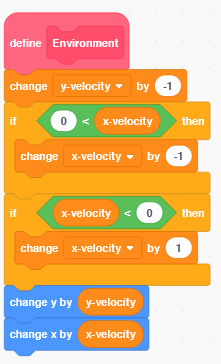
****

To imitate gravity, we’re going to change the y-velocity of the player by -1 every time that the game runs the “Environment” block. This is similar to how gravity works in real life, as gravity constantly makes the downward velocity of objects increase. This means that objects fall faster and faster towards the ground.

Remember: y-velocity is how fast the player moves up and down. So as we are making the y-velocity more and more negative, this will make the player fall down faster and faster.

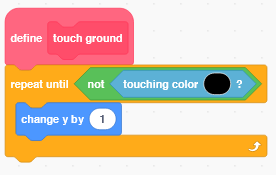
Next, using the two if statements, we’re going to change the x-velocity of the ball towards 0. This happens because if the x-velocity is bigger than zero, then we subtract 1. If its less than 0, then we add 1.

This is so that the player will stop moving if the user is not inputting movement with the arrow keys.



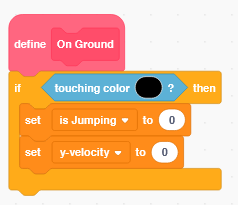
Now, we’re going to add the blue movement blocks. Since y-velocity and x-velocity control how fast the player moves in those directions, we will change the x and y coordinates of the player with these variables.

**4. Touch Ground**

****

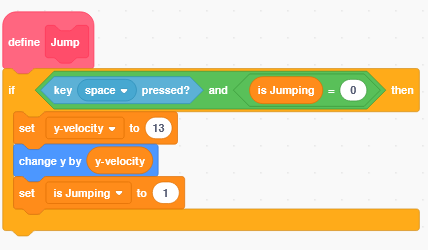
Touch ground will make the player stand on the ground, instead of just going through the ground. If the player is ever inside of the ground, this block will push it up until it is no longer in the ground, allowing the player to stand on it.

**5. On Ground**

****

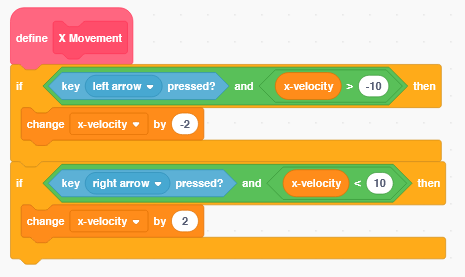
Next, we have to detect if the player is on the ground. We need to know this because the player should only be able to jump if they are on the ground. We also want to set the y-velocity to 0 when it is touching the ground so that the ball doesn’t try to go through the ground.

**6. Jumping**

****

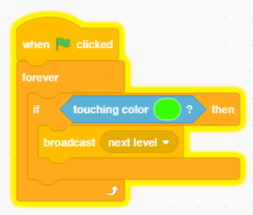
Using the if statement, we will only allow the player to jump if they are on the ground. This happens because is Jumping will only be equal to 0 if it is touching the ground. The “and” block means that the if statement will only happen **if** the space key is pressed **and** the variable is Jumping is 0. Once we know that the player is on the ground, we will make the ball jump by setting the y-velocity to 13. This means that the ball will start moving upwards since the y-velocity is more than 0. We also need to set is Jumping to 1 so that our game won’t let the ball jump again until it hits the ground.

**7. X Movement**

****

By making x-velocity negative, this will make the ball move left, and by making x-velocity positive it will make the ball move right. We also need to check if x-velocity is greater than -10 or 10 because we don’t want the ball to move faster than that.

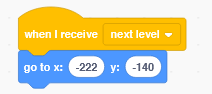
**8. Detecting Level Finish**

****

We’re going to use a separate forever loop to check if the ball is ever touching the color green because we are using green to show the end of the levels. If we touch the end, then we are going to use broadcasting to tell the game to go to the next level.

Note: Broadcasting is a way to tell all sprites and the backdrop about a specific event. There is a block that lets you tell sprites what to do when they receive a backdrop, and we’ll use it in the next step.

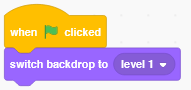
**9. Next Level**



When the next level broadcast happens, we want to put the ball back in the bottom left of the screen to start the next level.

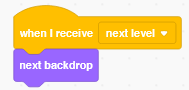
Coding the Backdrops

**1. Starting the Game**

****

The game needs to start on level 1.

**2. Next Level**

****

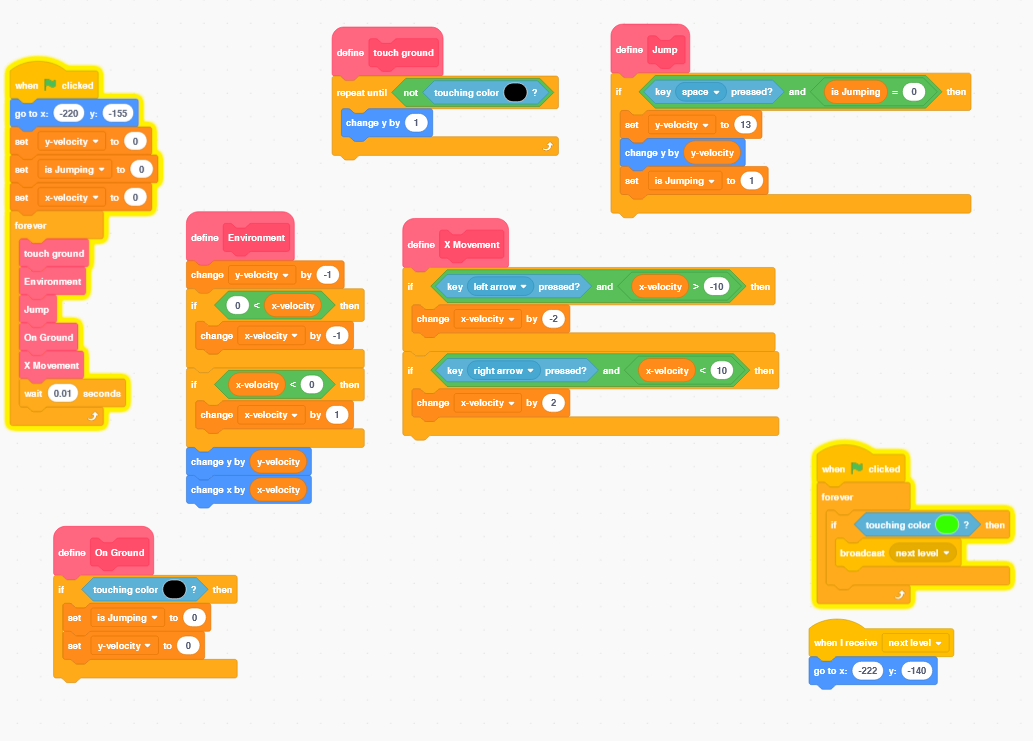
When the next level broadcast happens, we need to change the backdrop to the next level.

Note:

We have only created one level. You get to create the next levels of your game!

Congratulations, you have just completed an Hour of Code!

Completed Ball Code:



Completed Backdrop Code:

