**2.05 Step-By-Step Project**

**Step 1**

**​**Start off your project the same way you did with 1.04 and 2.02: head to the [Scratch website](https://scratch.mit.edu/) or your [desktop Scratch app](https://scratch.mit.edu/download) and login. Then, click "Create" to start a new program.



​​Navigate to the Tutorials menu in the top left corner of the screen and scroll down until you see the "Pong Game" tutorial. This will take you through all the steps to create your very own game!



​You can start the tutorial by selecting the Play option for the video. You can also see each step by clicking the green arrows in the tutorial.



**Step 2**

Set the stage for your program's performance! Begin by choosing a backdrop to be the background on the screen. Then, choose a sprite to be the "ball" and another to be the "paddle". Be sure to choose different backdrops and sprites from the ones in the Scratch tutorial!

**Step 3**

Let's get things in motion! First, add the code to make the ball bounce around by having the sprite begin pointing in one direction, then as part of a "forever loop", move a certain # of steps. Be sure to have the sprite respond to the edge of the screen and bounce back, otherwise you'll lose your ball!



​Then, add your code to have the paddle move when clicked. As part of yet another "forever loop", you can have the mouse be set to where the mouse clicks.



Finally, add the code that will tell your ball to bounce off of the paddle when it makes contact.



**Step 4**

Wouldn't be much of a game if no one was keeping score though, right? In a similar fashion to your counter from 2.02, create a variable named "Score" and have it change when a player scores. Insert the block into your main code.



Don't forget to have the score reset to "0" when a new game is started!



**Step 5**

As with all things in life, it's important that we set boundaries! Create a new sprite and add it to the bottom of your screen to be the "ground". Add some code to have the game come to an end when the ball touches the ground.



**Step 6**

**Save and submit!** Use the **File menu to save the work to your computer, then upload your .sb3 file through Educator and submit your assignment!**

****

**Grading Rubric**

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | **Excellent** | **Good** | **Needs Improvement** |
| **Program Design** | 21–30 pointsWork contains **all** required elements:* backdrop different from the tutorial
* a sprite to be the paddle that is different from the tutorial
* code to make the ball bounce
* code to move the paddle
* code to bounce the ball off the paddle
 | 11–20 pointsWork contains **most** of the required elements:* backdrop different from the tutorial
* a sprite to be the paddle that is different from the tutorial
* code to make the ball bounce
* code to move the paddle
* code to bounce the ball off the paddle
 | 0–10 pointsWork contains a **few** of the required elements:* backdrop different from the tutorial
* a sprite to be the paddle that is different from the tutorial
* code to make the ball bounce
* code to move the paddle
* code to bounce the ball off the paddle
 |
| **Program Content** | 26–40 pointsWork contains **all** required elements:* a variable named score
* one change score block
* one set score block
* a sprite to the bottom for the ground
* code to end the game if the ball touches the ground
 | 11–25 pointsWork contains **most** of the required elements:* a variable named score
* one change score block
* one set score block
* a sprite to the bottom for the ground
* code to end the game if the ball touches the ground
 | 0–10 pointsWork contains a **few** of the required elements:* a variable named score
* one change score block
* one set score block
* a sprite to the bottom for the ground
* code to end the game if the ball touches the ground
 |
| **Program Performance** | 21–30 pointsWork contains **all** required elements:* the program runs correctly
* the program contains no syntax or runtime errors
* the program produces the expected results
 | 11–20 pointsWork contains **most** of the required elements:* the program runs correctly
* the program contains no syntax or runtime errors
* the program produces the expected results
 | 0–10 pointsWork contains a **few** of the required elements:* the program runs correctly
* the program contains no syntax or runtime errors
* the program produces the expected results
 |
| **Total points: 100** |   |   |   |