KidzCoding Hour of Code - Scratch Balloon Game

Activity:

In this activity, students will learn how to make a simple and fun balloon popping game in Scratch. This activity makes use of conditional statements, variables, and loops. In the game, the ball will move up and down while the user controls the motion of the ball. Variables will be used to keep track of the score and the amount of time that the balloon waits for in between movements.

Learning Objectives:

* Develop an understanding of movement on the x/y plane
* Understand how to use conditional statements for in-game events
* Use variables to keep track of score and other factors

Materials:

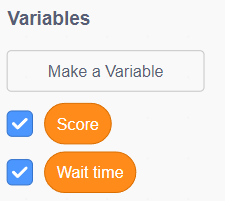
* Scratch starter project: <https://scratch.mit.edu/projects/872824384>
* Completed Scratch project: <https://scratch.mit.edu/projects/872576509>

Coding Time!

Starting the project

Open the Scratch starter project, and you will have all of the included sprites and costumes.

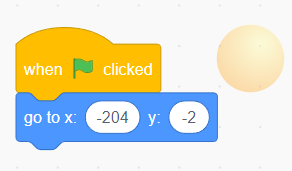
Variables



You will need two variables, wait time and score (these are included in the starter project). The score variable keeps track of the number of times you hit the balloon, and the wait time variable keeps track of the amount of time the balloon waits in between movements.

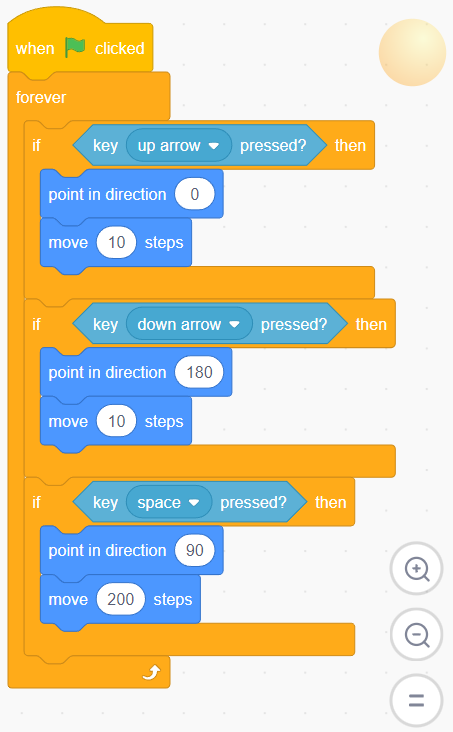
Coding the Ball

1.  **Placing the ball**



You will need to place the ball in a starting position, on the left side horizontally and in the middle vertically.

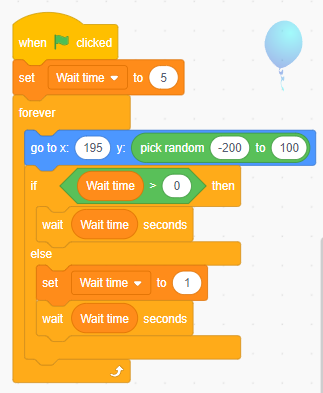
2. **Ball movement**



For this part, the ball will react to the up and down arrow keys, allowing the user to move it up and down the screen, the space key will cause it to move forward. This requires the use of sensing blocks for key inputs and control blocks for the forever loop and if statements (if you do not use the forever loop the actions will only happen once).

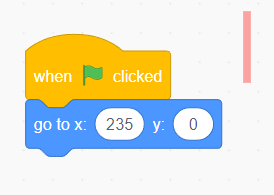
**Instructor note:** using if statements can be difficult for some students--if necessary, demonstrate how to code the blocks for one of the arrows and have them repeat it for the other.

**Balloon Movement**



First, we need to set the amount of time the balloon waits to a default value. In this case we chose 5 seconds. Next, we want the balloon to start on the far right-hand side of the screen and move up and down without leaving the screen. To do this, we will permanently set its x coordinate to x = 195 and have it randomly move between y = -200 and y = 100 using an operators block. Now we need to have the balloon stop moving for an amount of time determined by the variable wait time before moving again. To do this, we use an if else statement to make sure that we never have a time less than 0 seconds by setting the variable wait time back to 1 second if wait time is not greater than 0.

**Coding the wall**

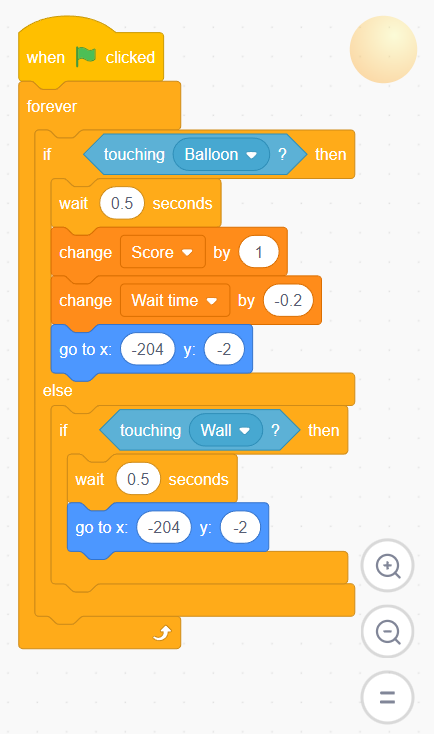


We simply use the red wall to prevent the ball from leaving the screen. We will set its coordinates to the far right side of the screen.

**Reacting to collisions**

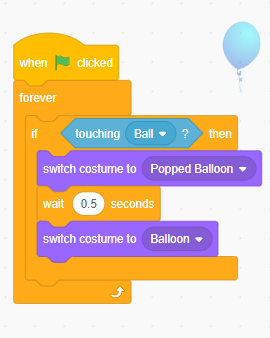
In this part, we will code the ball and balloon to react to collisions with each other and the wall.

**Ball collisions**



For this part, we want the ball to react if it collides with the wall or the balloon. If it hits the balloon we want it first to wait a short period of time the player can react, then we will increase the score to reward the player and decrease the wait time to make the difficulty higher as the player scores more points. After adjusting the score and wait time, we return the ball to its starting position. If the ball hits the wall instead, we will simply wait and then return the ball to its starring position (for a higher difficulty, the score could go down by 1).

**Balloon collisions**



This is a fun, and complete optional part that makes the balloon look like it has been popped after being hit by the ball. We simply switch to the popped balloon included in the starter package, wait half a second a switch back to the normal balloon custom.

Congratulations, you have just completed an Hour of Code!