KidzCoding Hour of Code - Soccer Game

Activity:

In this activity, students will learn design how to make a **simple** and fun soccer **game** in **Scratch**.This game uses **loops, variables,** and **conditionals**. In the game, a computer controlled goalie will try to prevent a player from shooting a soccer ball into the goal.

Learning Objectives:

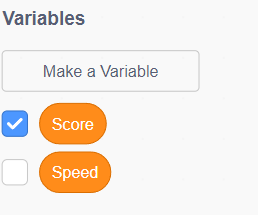
* Understand movement on the x/y plane
* Understand how to use conditionals for keyboard inputs and sprite interactions
* Understand how to use loops to continue actions
* Understand how to use variables to keep track of score and movement speed

Materials:

* Scratch starter project: <https://scratch.mit.edu/projects/873852038>
* Scratch completed game: <https://scratch.mit.edu/projects/873043554>

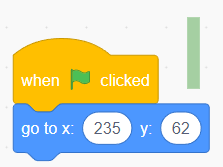
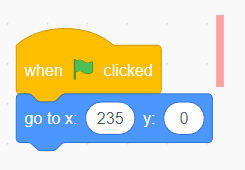
Coding Time!

Variables



We will need a score variable to keep track of the amount of goals the player has scored and a speed variable to keep track of the speed the goalie moves at.

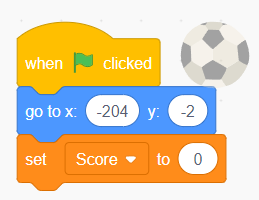
**Setting Wall and Goal Sprites to Starting Coordinates**

****

Set the red wall and the green wall (goal) to their respective coordinates.

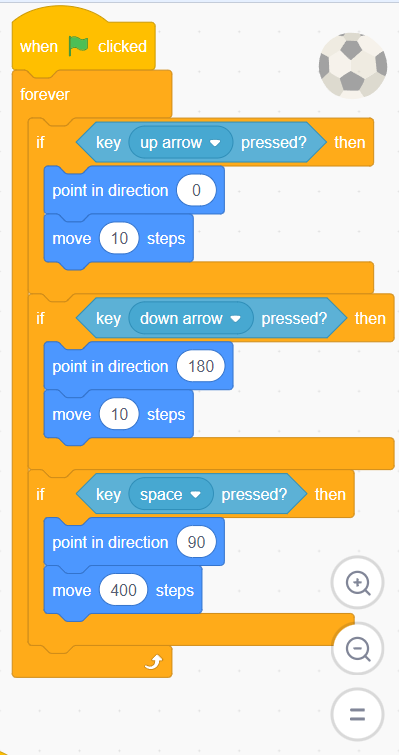
**Coding the Soccer Ball**

1. **Placing the soccer ball**



You will need to place the soccer ball in a starting position, here we have placed it roughly in the middle vertically of the left side of the screen.

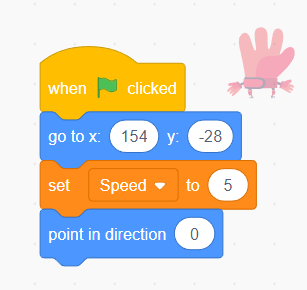
1. **Moving the soccer ball**



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).

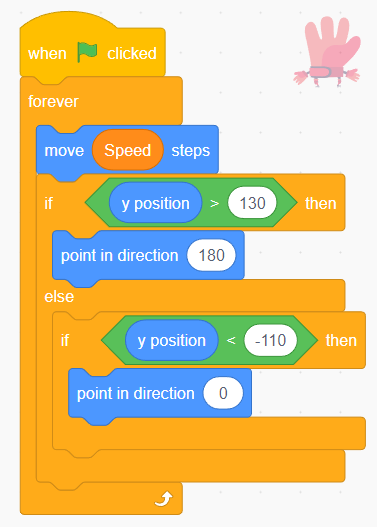
**Coding the Goalie**

1. **Placing the goalie**

****

First we must place the goalie, in this case we place it roughly in the middle of the screen vertically and to the far left side of the screen horizontally. We also set its initial direction to 0 degrees (pointing up) and its initial speed to five steps.

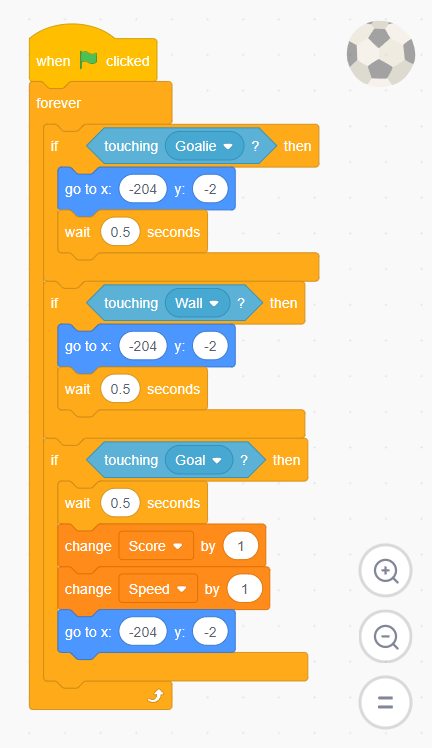
1. **Moving the goalie**



We want the computer to move the goalie up and down while the player attempt to shoot the soccer ball. To do this, we use a forever loops so that the goalie continuously moves throughout our game. Next we have the goalie move the amount of steps determined by our variable “steps” so that we can adjust how fast it moves. Finally, to keep the goalie from leaving the screen, we check to see if it has passed a y position above the goal and below the goal. If this is the case, then we change the direction, so it goes back across the goal.

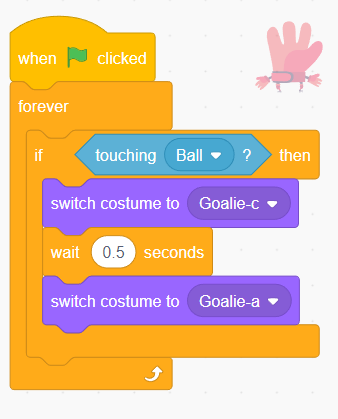
**Collisions**

1. **Soccer ball collisions**

****

First, we want to make sure that the ball is reset to its initial position if it touches anything but the goal. To do this, we use an if statement with a sensing block so that the ball reacts if it touches the goalie. Then we use the go-to movement block and use the wait block to delay it, so the user can react. We repeat the exact same process for the wall sprite. Finally we need to adjust score if the soccer ball hits the goal. To do this we use a sensing block combined with an if statement like the two previous steps, then we change the score by 1 and the speed by 1 using variable blocks and use the go-to block to reset the soccer ball to its initial position. Increasing the speed and the score will reward the player for scoring goals while making the game progressively more challenging as the player score more goals.

1. **Optional: changing the appearance of the goalie**

****

This completely optional fun addition changes the goalie sprite, so it looks as if the goalie dives to make a save after the soccer ball collides with the goalie. We use an if statement to check to see if the goalie is touching the ball, we then use the switch costume block in the looks section to switch the goalie custom (costumes are provided in start project), wait five seconds and reset back to the normal goalie costume.

Congratulations, you have just completed an Hour of Code!