KidzCoding Hour of Code - Number Guessing Game

Activity:

In this activity, students will make a simple number guessing **game** in **Scratch.** They will learn how to use **loops**, **conditionals**, **variables**, and **comparators**. In the game, the program will pick a number between 1 and 100 and the player will try to guess the number. The program will also provide feedback on each guess, saying whether it was too high or too low

Learning Objectives:

* Develop an understanding of comparators and their uses
* Learn how to use conditional statements with loops
* Utilize variables to keep track of score and in-game information

Materials:

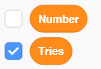
* Starter Project - <https://scratch.mit.edu/projects/875933116/>
* Completed Game - <https://scratch.mit.edu/projects/875928995/>

Coding Time!

Starting the project

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

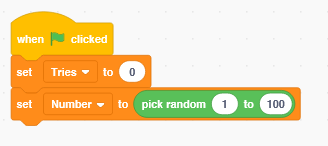
Variables



This guessing game will need two variables. The first variable, number, will keep track of the number that the player is trying to guess. Make sure to uncheck this variable so that the player isn’t able to see what the number is when playing the game! The second variable is tries, and it will keep track of how many guesses the player makes.

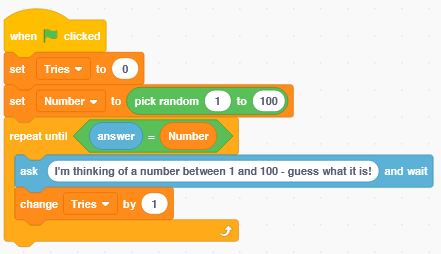
**Coding the Game**

**1. Setting up the variables**

****

At the start of the game, we need to set tries to zero because the player has not made any guesses yet. We will also use the “pick random” operator to choose a random number for the player to guess and assign it to the number variable.

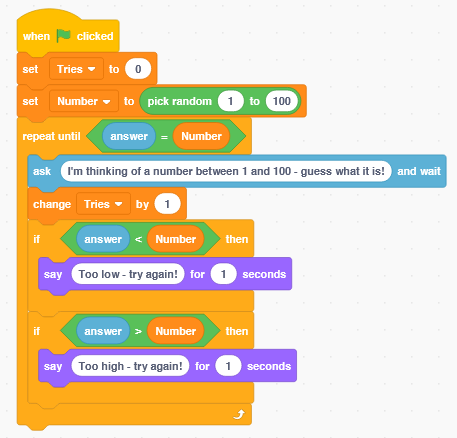
**2. Asking for the player’s guess**



For this game, the player will make guesses until the player guesses the number. To do this in Scratch, we will use the “repeat until” block. This block is a loop, and it keeps asking the player to guess until the player’s guess (answer) is the same as the random number. Also, after every guess we will change tries by one to keep track of the amount of guesses that the player takes.

Remember: if the player’s guess is ever equal to the random number, then the loop will end.

**3. Giving the Player Feedback**



Now, we will use if statements to tell the player if they guess too high or too low. If the player’s guess (answer) is less than (<) the random number, then the program will say that it is too low. If the player’s guess is greater than (>) the random number, then the program will say that the guess is too high. Remember that the loop will end if the guess is the same as the random number.

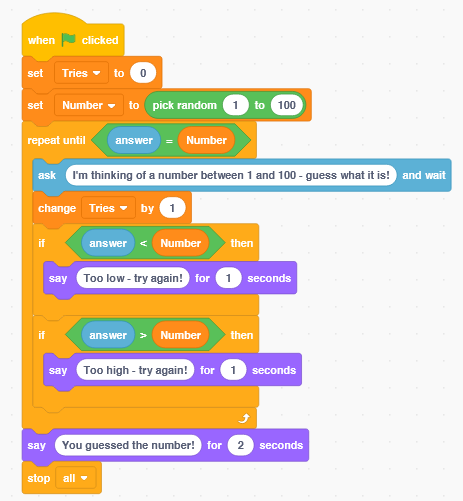
Note: Comparators are the green blocks with the symbols “<”, “>”, and “=”.

“<” means less than

“>” means greater than

“=” means equal to

**4. Ending the Game**

****

We know that when the loop ends when the guess is the same as the random number, so that is when the game should end. To end the game, we will say that the player guessed the number. Then, we will stop the program will the “stop all” block.

Congratulations, you have just completed an Hour of Code!