

Lab #7

IICT BS DS Fall 2022

Scratch Programming:

What is Scratch

Scratch is high-level visual programming language that makes computer programming exciting. With Scratch it is easy to create interactive stories, animations, and games. You can use Scratch to learn computer programming concepts while playing with the fun tool.

Scratch has an extensive collection of graphics, animations, sounds, and music included that you can use to create your projects. You can also use the built-in graphic design tool to create unique content for their projects.

Why Learn Scratch:

Computers are powerful tools that can be used for solving problems, completing tasks, watching movies, or playing games. However, the programs that we use on the computers must be written. Scratch will allow you to write your own programs and games that you can use or share with your friends.

While learning Scratch, you can also use skills that you learn in other classes such as math, music, or science. For example, you could write a calculator program, you could create a virtual piano, or you could write a program to simulate a scientific experiment.

Scratch Interface:

Scratch provides you with a powerful interface to create programs such as games or animated stories. The scratch team at MIT designed the interface to be easy to use.

Use the links on the left to read brief explanations of each section of the scratch interface.

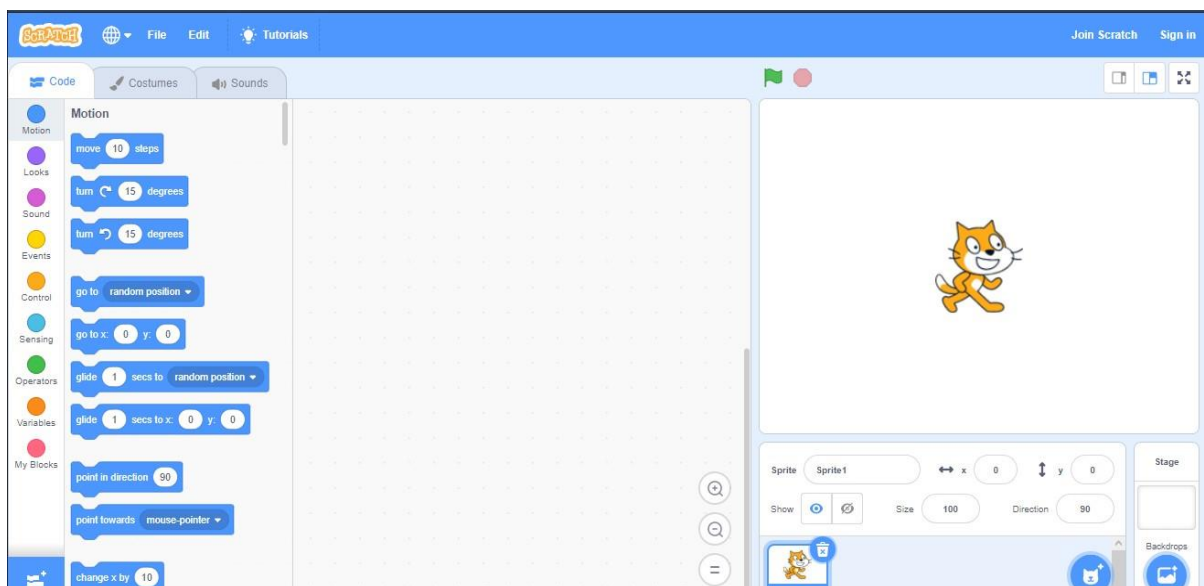


Fig. 1 (Scratch Interface)

Scratch Interface - Block Palette

The block palette is where the different script blocks are located. The different type of script blocks include: motion, control, looks, sensing, sound, operators, pen, variables. This website will go into greater detail for each function as you progress. The purpose of this section is to just give you a rough idea of each script type.

- **Motion** - controls the up, down, left, right, and rotation movement of the sprites.
- **Control** - allows you to specify things such as what starts and stops your sprites, how you can move the object, conditional operators such as if/then and repetition.
- **Looks** - affects the appearance of your sprite such as color and costume.
- **Sensing** - controls to sense if your sprite is touching the edge, another color, another sprite, at a specific X or Y coordinate, the sound volume, etc.
- **Sound** - control sounds, pitches, and volume.
- **Operators** - Logic operators including tools to perform math functions, select a random number, greater than, less than, equal to, etc.
- **Pen** - pen functions allow you to draw lines and objects on the stage. □ **Variables** - create variables to hold numbers or text.

Scratch Interface – Stage:

It is important to not confuse the stage and the stage window.

The stage is basically the background of your project. Like sprites, the stage can have different costumes that change as the story plays out.

The stage window is the main area where the action of your program takes place. If you were to program a game or animated story, the stage window is where the action would take place when you start the program. You should also pay attention to the green flag and red stop sign in the upper right corner of the stage window. These buttons can be configured to control the beginning and end of your program.

You will also find the stage toolbar above the stage window.

The stage toolbar functions, from left to right, are:

- **stamp** - create a copy of an existing sprite.
- **scissors** - delete a sprite
- **grow sprite** - increase the size of a sprite
- **shrink sprite** - decrease the size of a sprite
- **small stage** - decrease the stage windows and increase the script area.
- **full stage** - increase the stage windows and decrease the script area. □ **presentation mode** - switch the stage window to full-screen.



Fig. 2 (Sprite in Scratch)

Start a new project

To code in Scratch, first open the Scratch. Next, click on the “**Create**” button to make a new project. You should have a screen that looks like this:

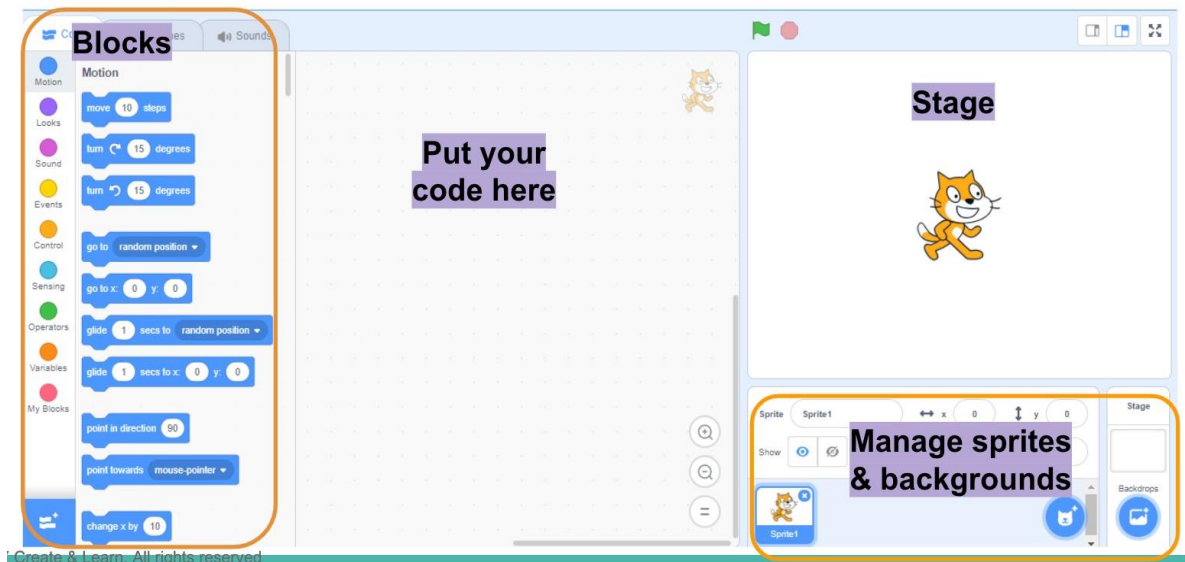


Fig. 3 (Blocks in Scratch)

Drag the code blocks:

The code blocks are on the left-hand side of the screen. To code, click and drag the blocks to the large center space. On scratch, the characters and objects are called “**Sprites**” You can add or delete as many sprites as you want. Each time you add a sprite, it will appear on the stage.

Click on sprites to code for them:

Click on each sprite to code for that particular sprite. There are hundreds of fun sprites to choose from.

Whether it’s a soccer player, a butterfly, or a ballerina, our students in our Scratch Ninja course never cease to amaze by their creative choices and story lines.

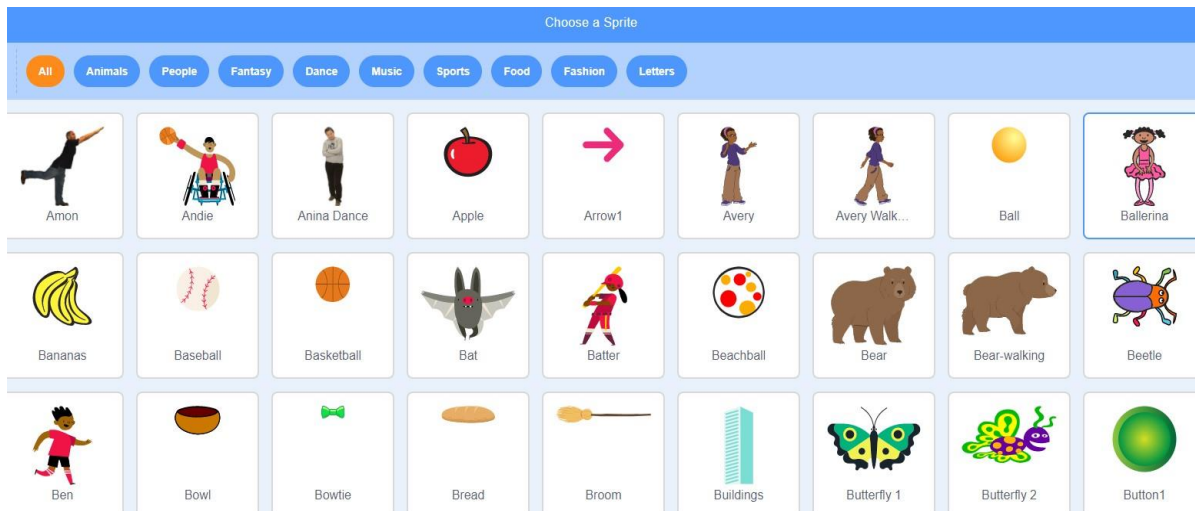


Fig. 4 (Objects in Scratch)

To code, you can drag blocks of code from the left-hand side and connect them together. Each sprite, as well as the background, will have its own code.

These blocks can make sprites move, make sounds, and change color. And when connected together form a series of actions to build your games, animations, and other projects.

Watch your code run:

After you've coded your project, you can click on the Green Flag to see your code run on the Stage.

If you want to save or share your project, make sure it's saved under your account. If you already have an account, click **"Log In"** If you need to make an account, click **"Join Scratch"** and follow the instructions (make sure you have a parent with you!).

Explore basic coding blocks to learn to code with Scratch:

There are many different types of blocks on Scratch and we're always excited to learn which blocks are our students' favorites.

Notice how most of the blocks are shaped with a special notch at the top and the bottom; this is so that they can connect together! Here are some of the most important blocks:

Events blocks in Scratch:

These yellow-colored blocks have a special shape, with the bump at the top. These blocks are **"Starting blocks"** meaning they must go at the top of any chunk of code we create. They tell us when the code will be run.

Event Block	When: How it Works



Run the code when the green flag is clicked (when the program begins). Most of the time, we use this block



Run the code when a key is pressed. Use the dropdown menu to choose which key you want!



Run the code when the sprite is clicked.

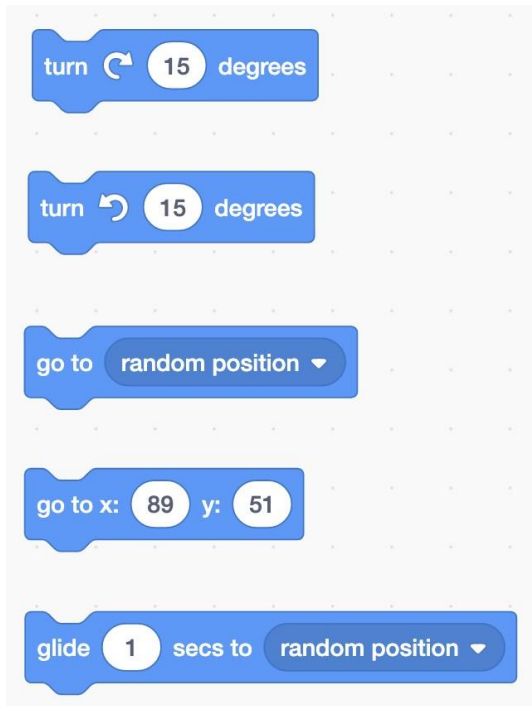
These blue colored blocks allow your sprite to move, rotate and glide.

Motion Block



Types of Movement

This block allows you to move your sprite. It will move in the direction your sprite is facing



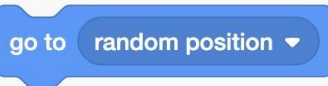
These blocks allow you to rotate your sprite to the right or the left.



turn 15 degrees



turn 15 degrees



go to random position ▼



go to x: 89 y: 51



glide 1 secs to random position ▼

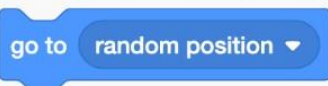
These blocks allow you to rotate your sprite to the right or the left.



turn 15 degrees



turn 15 degrees



go to random position ▼

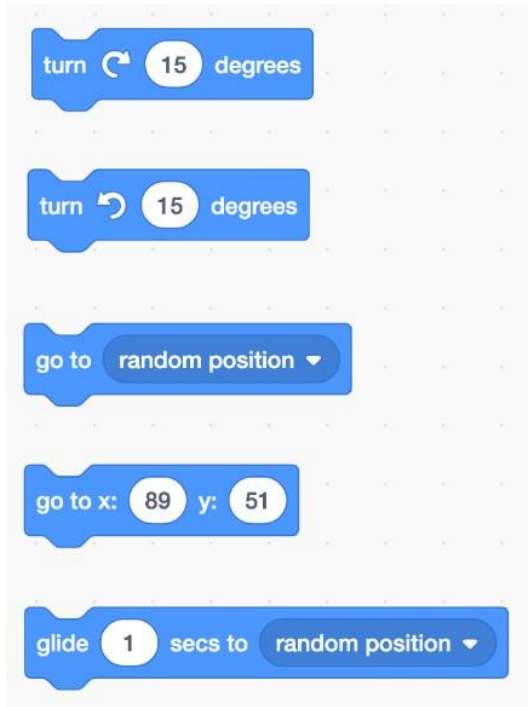



go to x: 89 y: 51



glide 1 secs to random position ▼


This block lets your sprite “**jump**” to a position. You can click on the dropdown menu to see the different options!

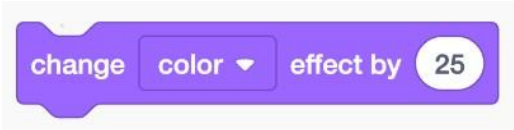
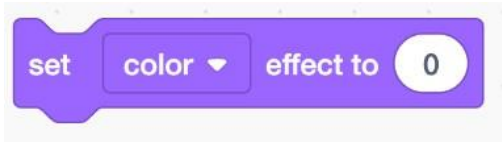
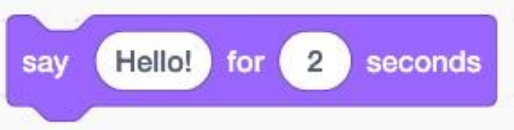
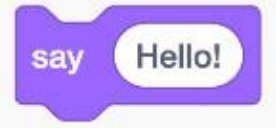
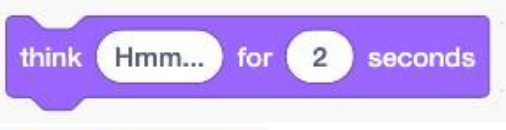
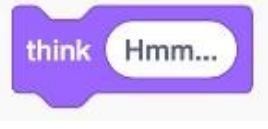
	<p>This block allows your sprite to smoothly glide across your screen to a position of choice.</p>
	<p>These blocks allow you to adjust the direction that your sprite is facing</p>

Try combining a motion block with an event block to see what happens!

Looks blocks

These purple-colored blocks change the appearance of your sprite.



Looks Block	Appearance: Types of Looks
	<p>These blocks change the size of your sprite.</p>

	<p>These blocks will change the color of your sprite.</p> <p>Use the dropdown menu to see other fun effects!</p>
    	<p>These blocks will create a speech or thought bubble for your sprite, with the text in the code block</p>

Control the flow of your Scratch code:

These blocks are found in the “**control**” section, colored in orange. Like the events blocks, they also have a special shape. Loops enable the continually run and repeat.



<p>Loops Block</p>	<p>Repetition: How Long to Run the Code</p>
---------------------------	--

	<p>The forever loop will keep running the code (as long as your program is running)</p> <p>Notice that the forever loop doesn't have a notch at the bottom. This is because the loop will keep running forever, so nothing added under it will run!</p>
	<p>The repeat loop allows you to specify how many times you want to run the code.</p>

Sounds - Blocks to enable “talking” sprites and more

This section allows you to add sound to your program. Each sprite has different sounds, but you can also add your own from the “**sounds**” tab. For example, get your dog sprite to bark or record some sounds to have it “**talk**”.

Sounds Block	Start or Play: Types of Sound
--------------	-------------------------------

	<p>Will start playing the sound. Any code blocks under this one will run as your sound it playing</p>
	<p>Will play the sound until finished. Any code blocks under this one will run once the sound is over</p>

Catch Game by using Scratch:

Let's create a Game by using scratch. A game where objects are falling from the sky. And you catch them to make scores. It could be any object that's falling. It can be character or object that's catching. Let's break this into five steps.

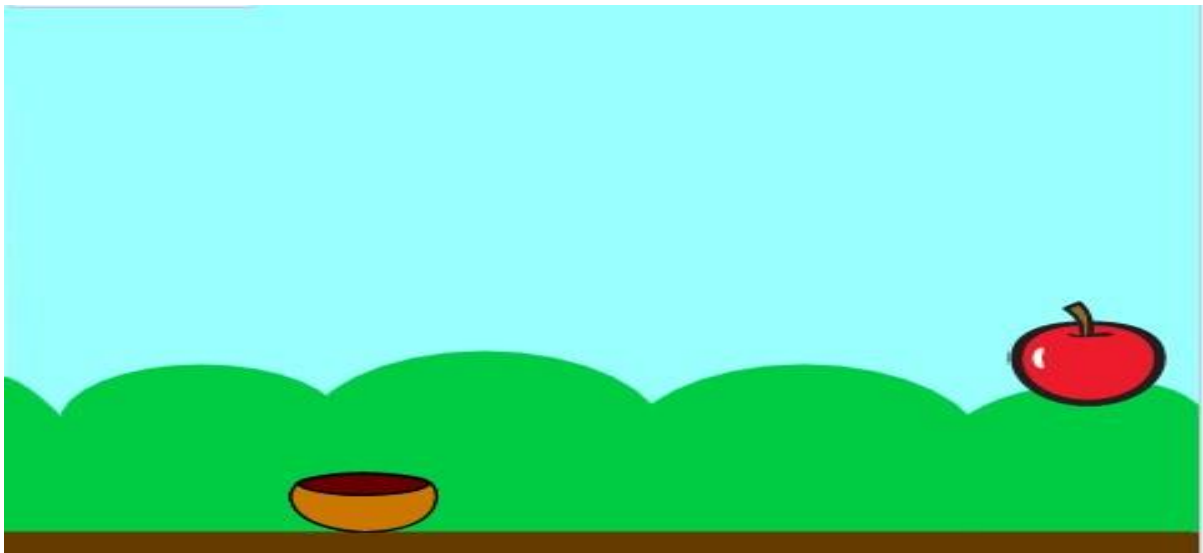


Fig. 5 (A Drop Catch Game)

- Move a catcher
- Go to the Top
- Fall down
- Catch it
- Keep scores **Move a catcher:**

We want to catch the things falling from the sky. And we're going to make it move with arrow keys. To make a catcher click the button **"Sprite"** to choose character or object.

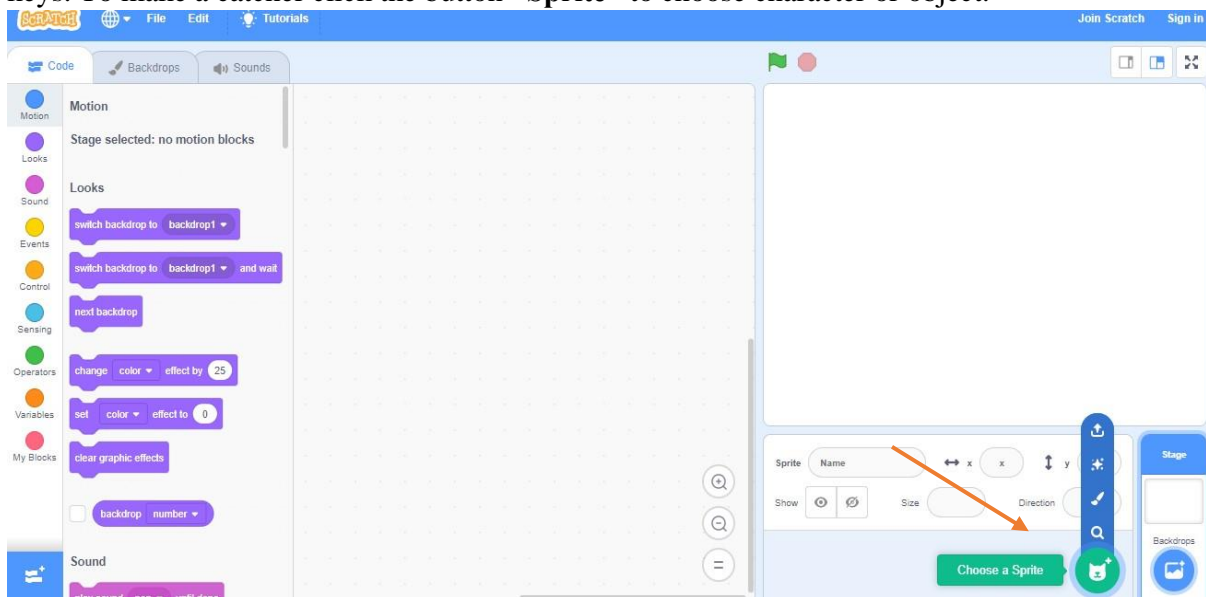


Fig. 6 (Add an Object in Scratch)

By clicking Sprite, you will get a list of characters and objects. You can choose whatever to want.

Let's use bowl for this purpose.

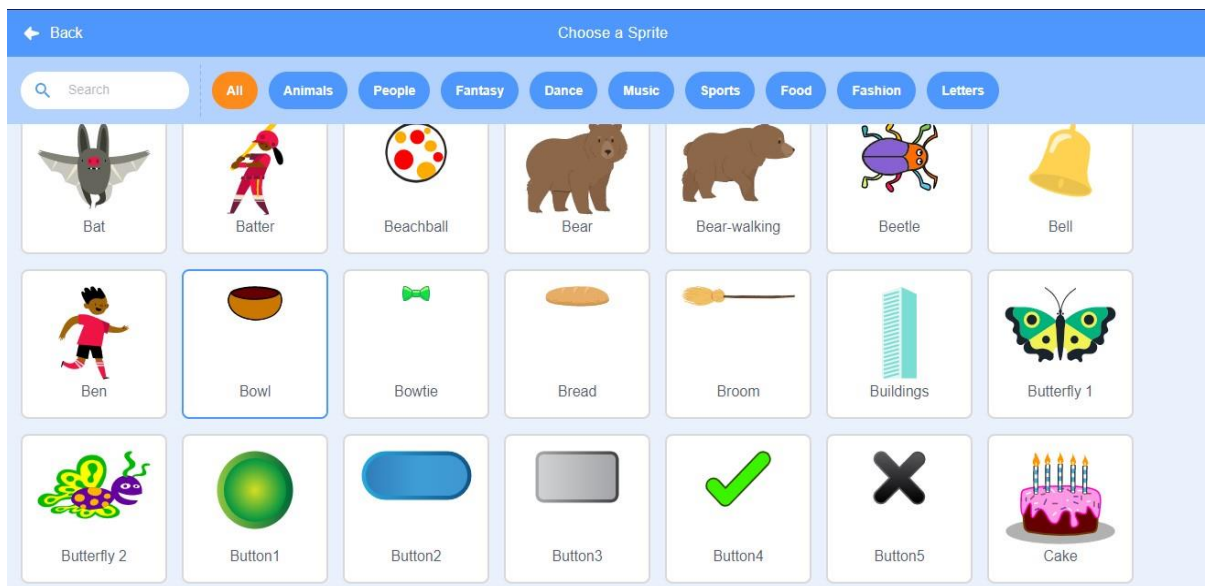


Fig. 7 (Select Sprite)

Let's add a backdrop too. You can pick any background.

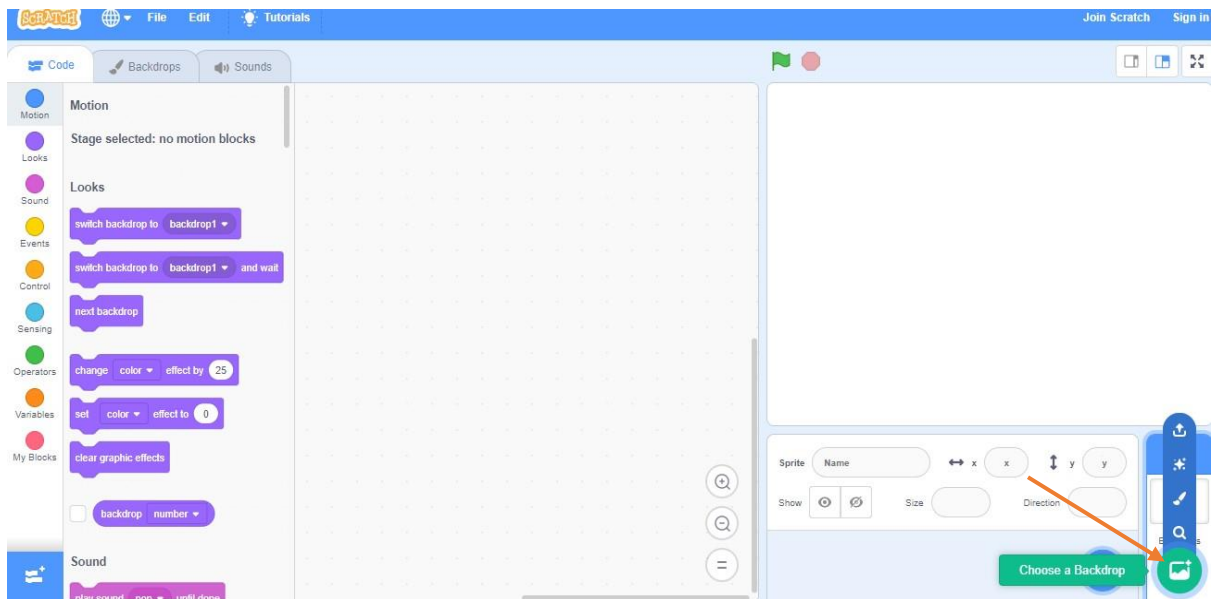


Fig. 8 (Add Background)

After choosing you board will look like as follows.

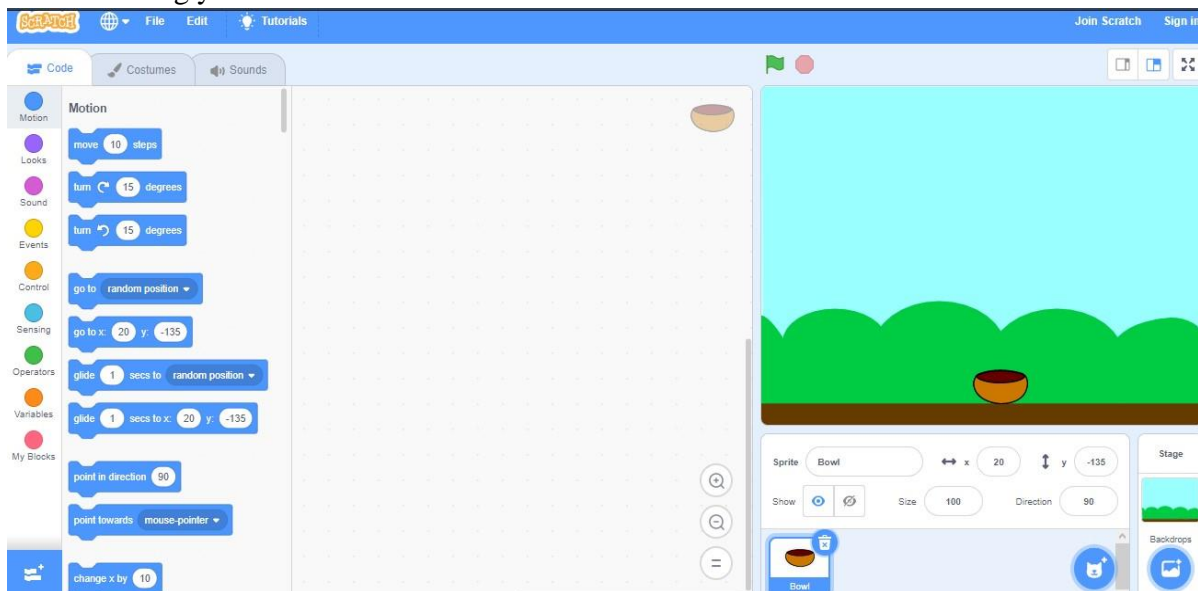


Fig. 9 (Make Moving a Bowl)

We want to make bowl able to move side by side by pressing arrow keys. So, it can catch objects whatever going to fall from top.

We will use a block (change x by 10) from the motion nav bar.

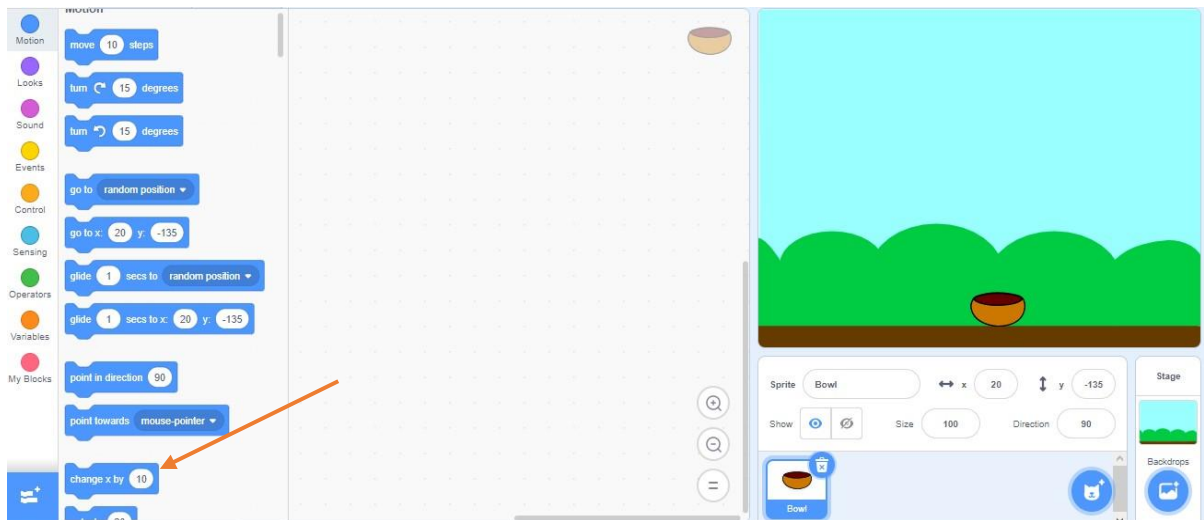


Fig. 10 (Change Position of Bowl)

By clicking this bowl will change its position. Let's make the bowl able to move by pressing keys.

There is a helpful block for this purpose. Let's take if then block from control bar.

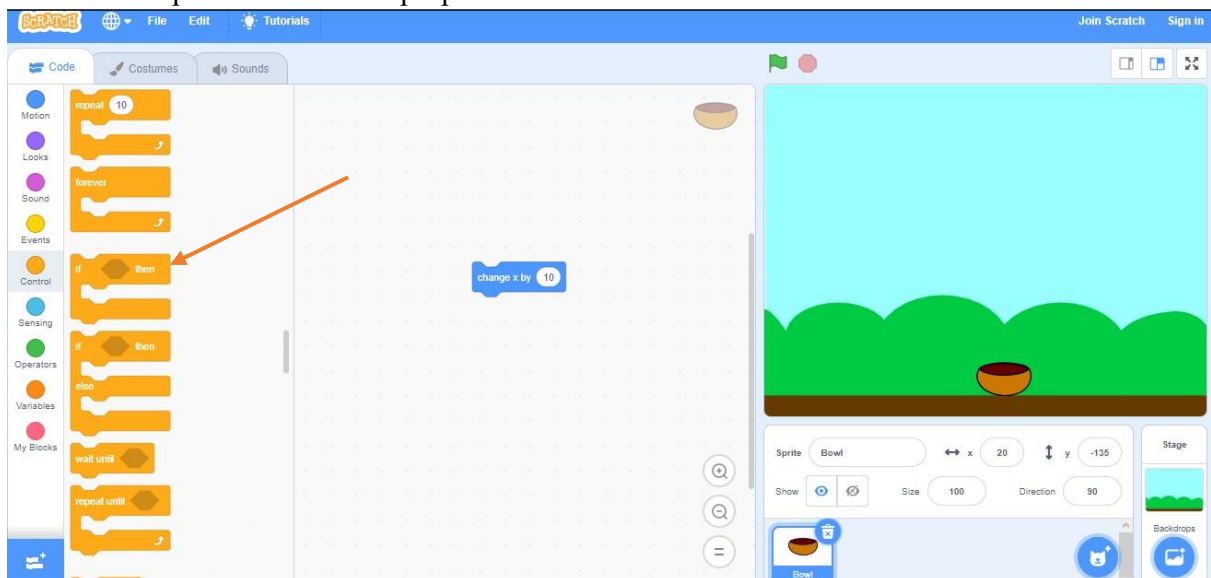


Fig. 1111 (Add if-then Block)

By using if then block bowl will move only when a specific event in if condition will occur. So, in our game if right key is pressed. We can make the bowl change x by 10. To choose event we will go to **"Sensing"** bar.

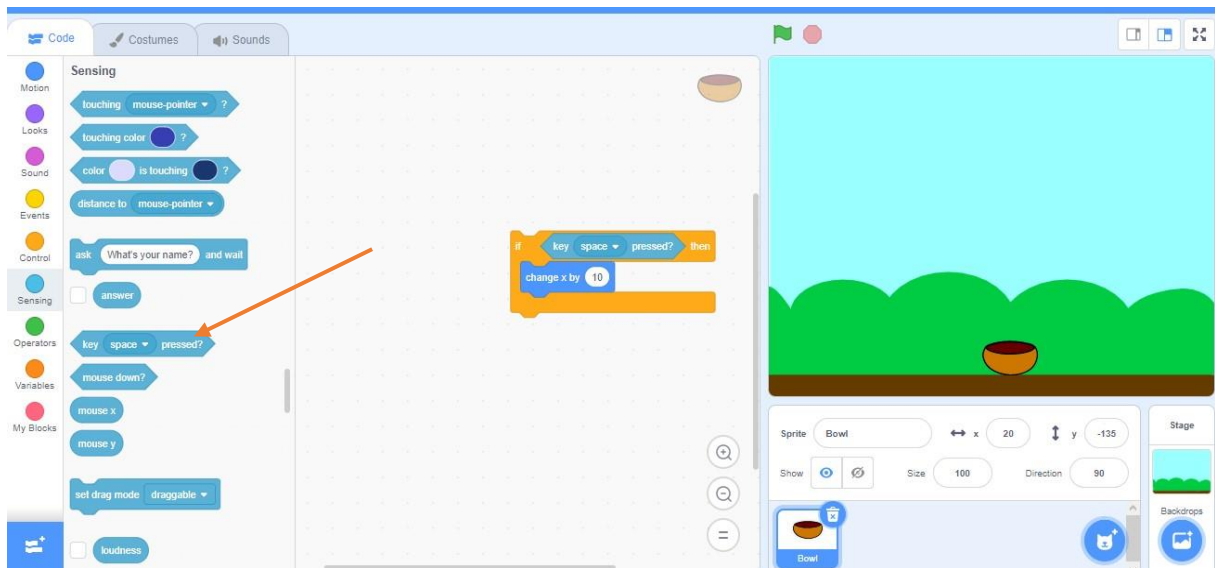


Fig. 12 (Add Condition Sensor)

Let's try it out. And let's make it start by clicking flag. Here by pressing space key nothing will happen. It is because if then block only being checked once. When clicked green flag. Its asking is the right key being pressed right now? But then it just stops doing the code. It should check if then statement always.

For this purpose, let's add a forever block around if then block from control bar.

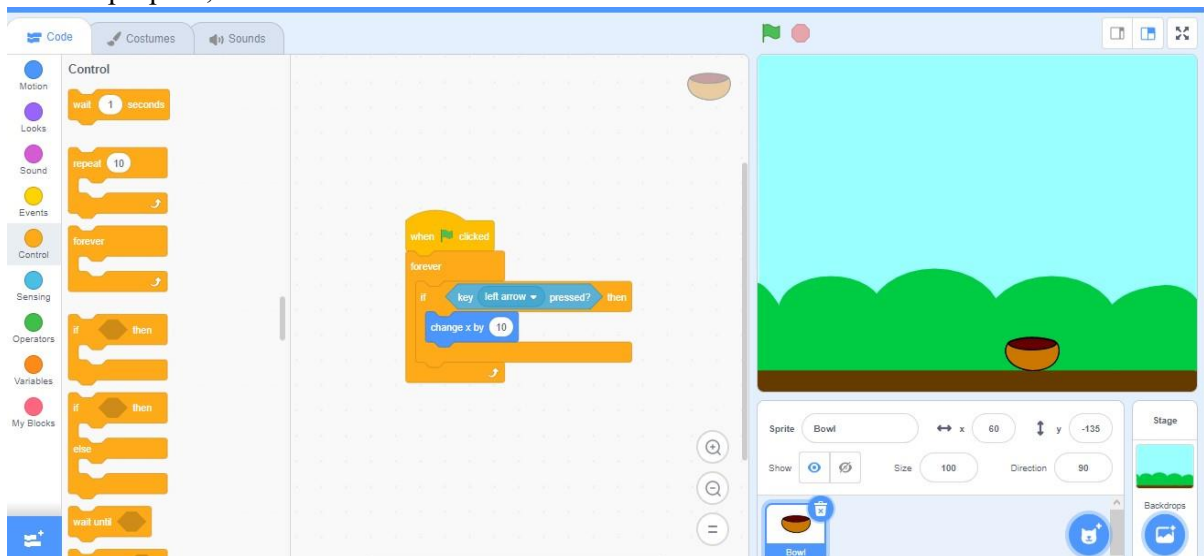


Fig. 13 (Add Click Event)

Now it is always checking, is the left arrow key being pressed? Now for right key pressed instead of changing x to 10, we will make it to change x by -10.

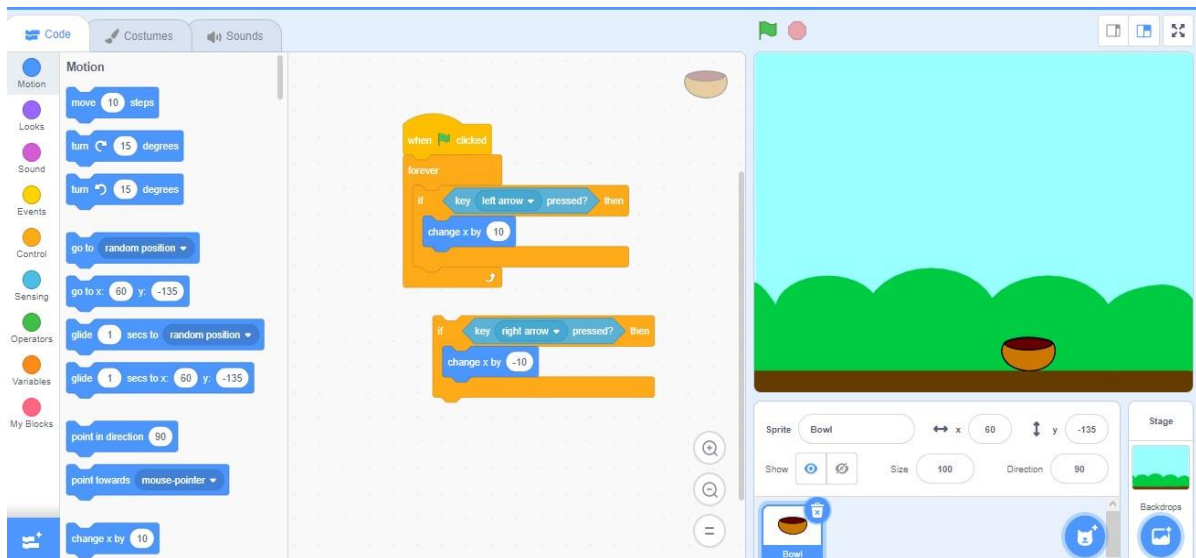


Fig. 15 (Making Bowl Moving left & Right)

Now let's put this block inside forever loop.

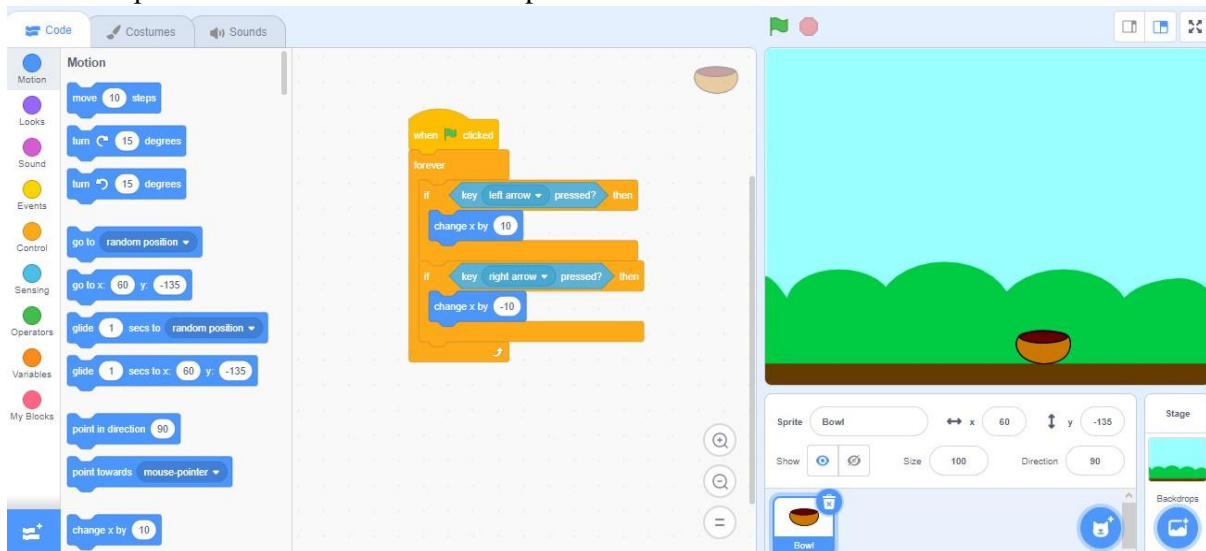


Fig. 16 (Add Right Arrow Key Listener)

From now you will be able to move bowl by pressing right and left arrow keys.

Go to the Top:

Next, let's pick an object that we want to be falling. Make it go to the top of the screen. For this purpose, you can use any object, Let's pick apple to be falling.

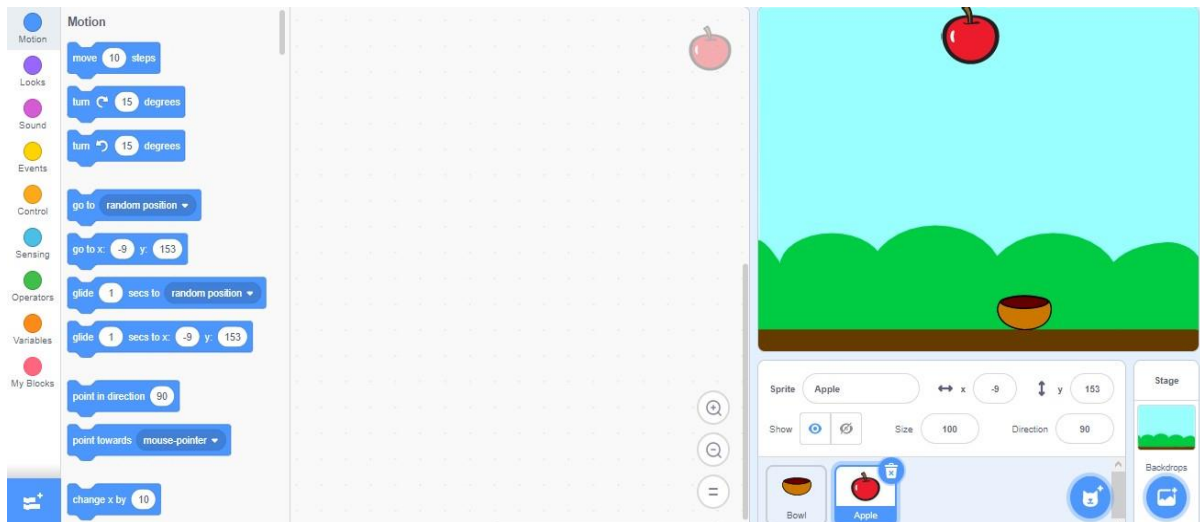


Fig. 17 (Add Apple to Game)

We want apple to start from the random position from the top. For random position we will use this block from motion bar.

To set its height of apple, let's set "Y" to 170 to make it high on the top. By this it would go to random position on the top, because, "Y" is set to 170 which is top position.

Fall down:

Now let's make our apple to fall.

If you change "Y" by 10 it would go up and if change "Y" by -10. It would go down.

If put this inside forever loop, it will be falling down and it will be keep falling while hit the surface.

So right now, we have an apple that can fall one time. What would we like to do, is if it's at the bottom it should go to the top again? This is another good time to use if then block.

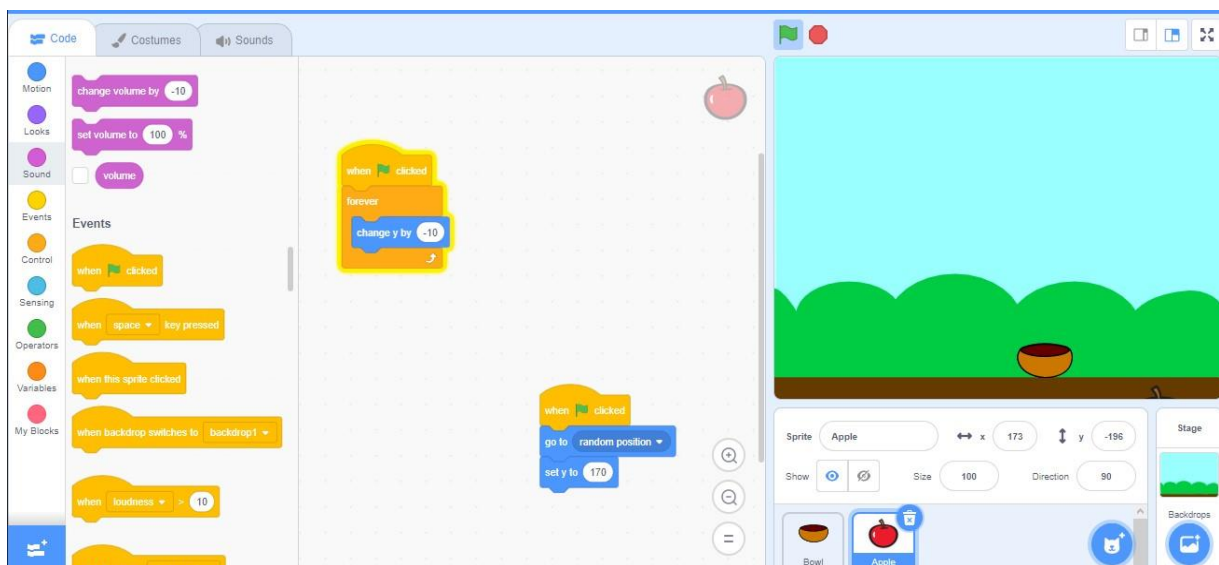


Fig. 18 (Make Apple Moving Down)

Let's drag on if then block out. In here we need to put some code, that check if the apple is at the bottom. How can it check that apple is at the bottom?

As we know “Y” check how much apple is on height. “Y” is -170 when apple is at the bottom. Cause if the apple’s “Y” position is that low. To check position, we will use “Y” position blue block from motion. And also need to use green block to check value from operation bar.

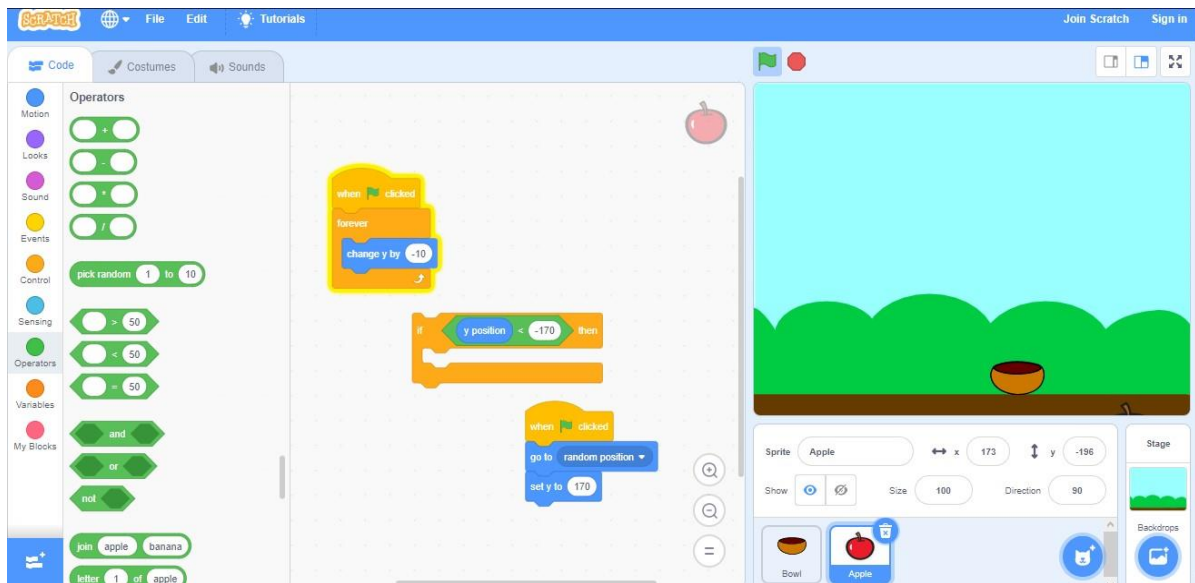


Fig. 19 (Position Checker for Apple)

We are checking if “Y” position is less than 170. On this check we want apple should go on top again. Let’s add a logic to take it on top.

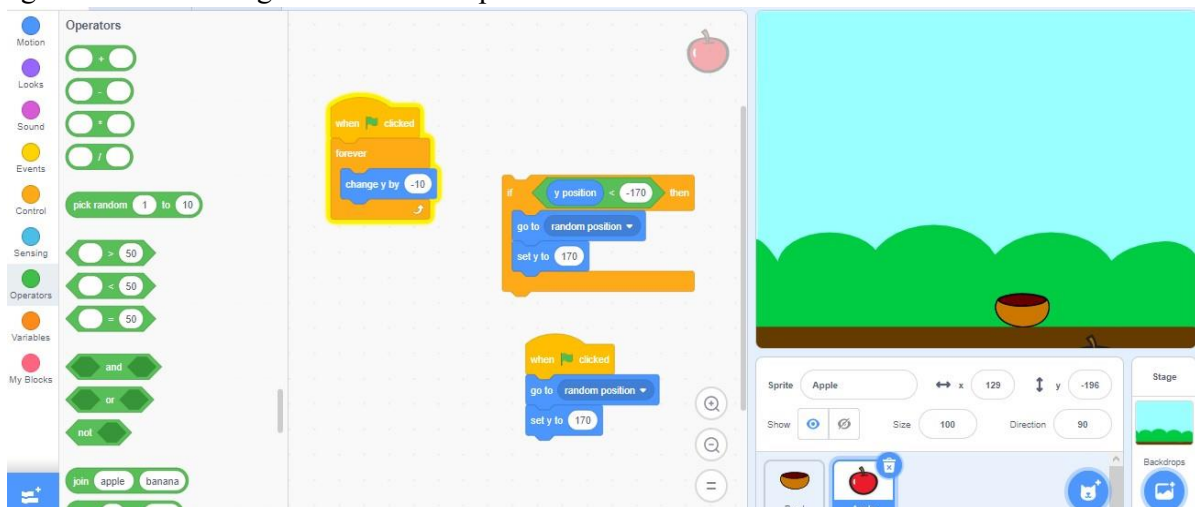


Fig. 20 (Make Apple Move to Top After Touching Bottom)

By logic inside if block it will go on the top. And let’s put this block inside forever loop. Now you will see apple will go to the top when it touches bottom.

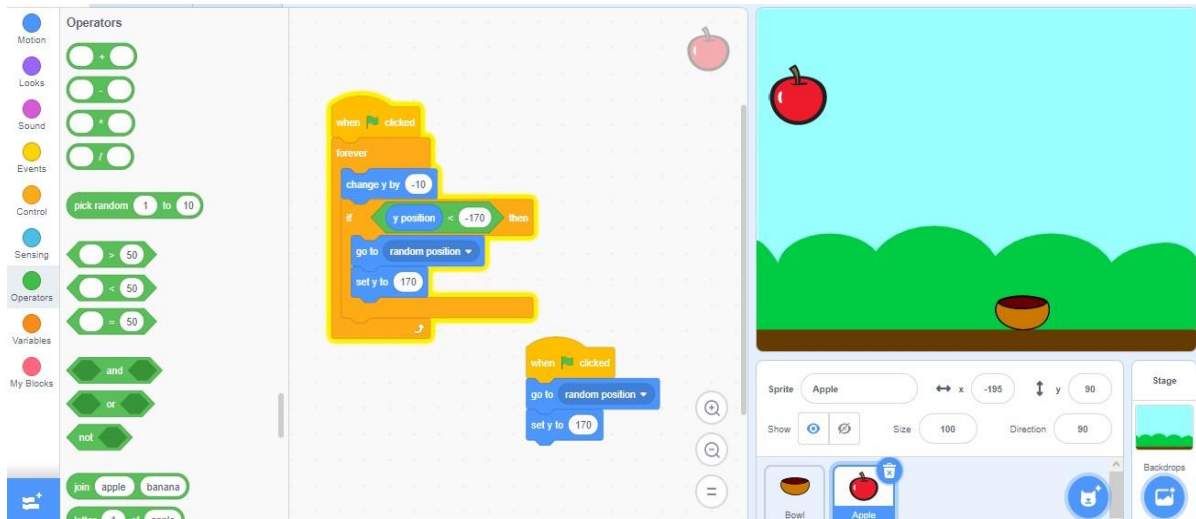


Fig. 21 (Apple Move)

Let's discuss logic of this block, so every time the apple moves down by -10, it asks me if on the bottom, if it is bottom position will be change to 170.

Catch it:

Now, let's make it so that we can catch the apple in the bowl. Instead of them just falling through the bowl. If the apple touches the bowl, then player catch it. So, it should stop falling. So, this is another good time to use if then block. And there is another helpful block in the sensor block, if object is touching to something else.

We will change mouse pointer to bowl to touch. Now block will test whether it touch bowl or not.

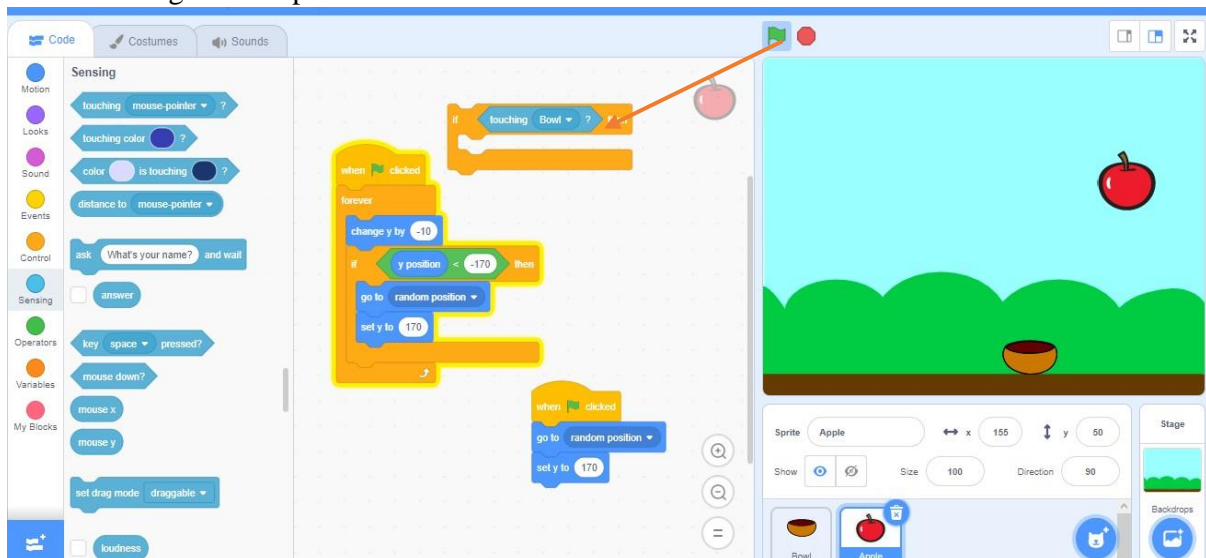


Fig. 22 (Catch the Apple)

Now after touching it should go on the top. When it touches bowl, it should produce a sound. It can be added from Sound bar.

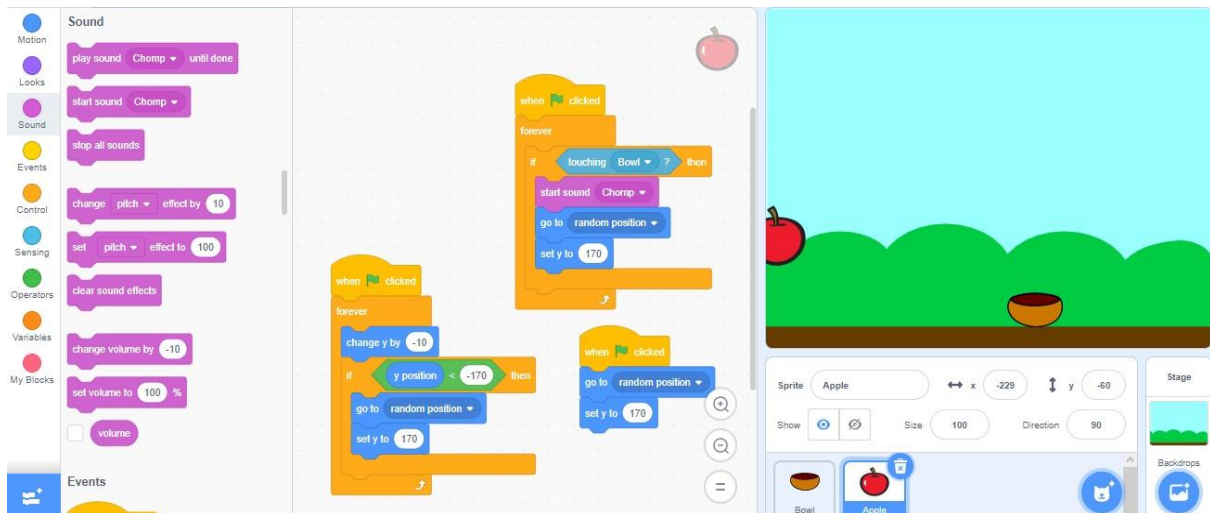


Fig. 23 (Catch the Apple)

Keep scores:

So now it should add score when bowl touches the apple. To make the score, we can add a variable. Let's add a variable to keep track of scores.

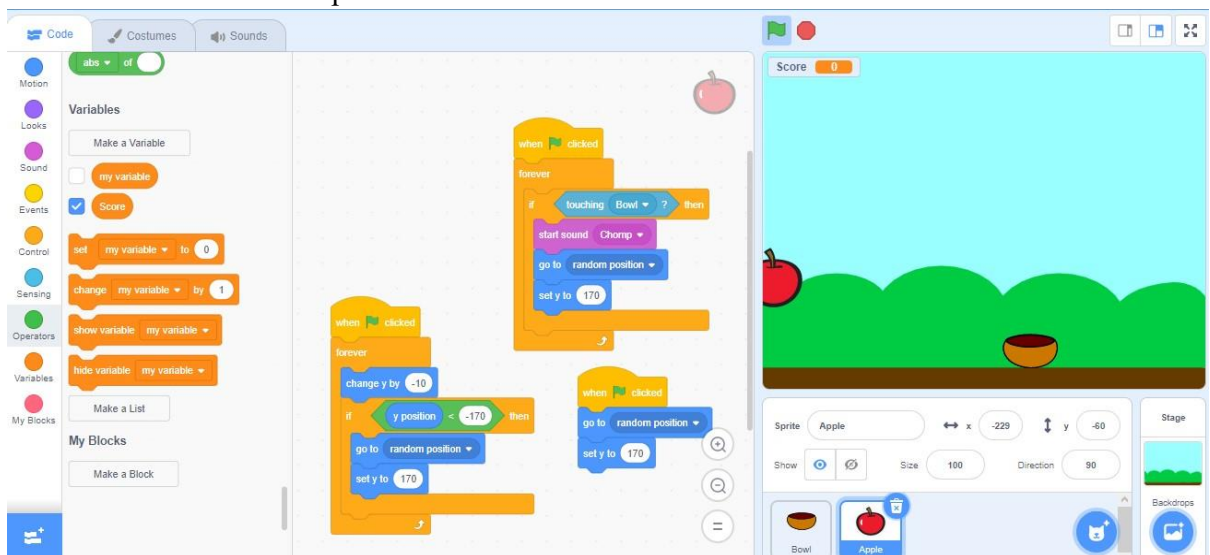


Fig. 24 (Count the Score)

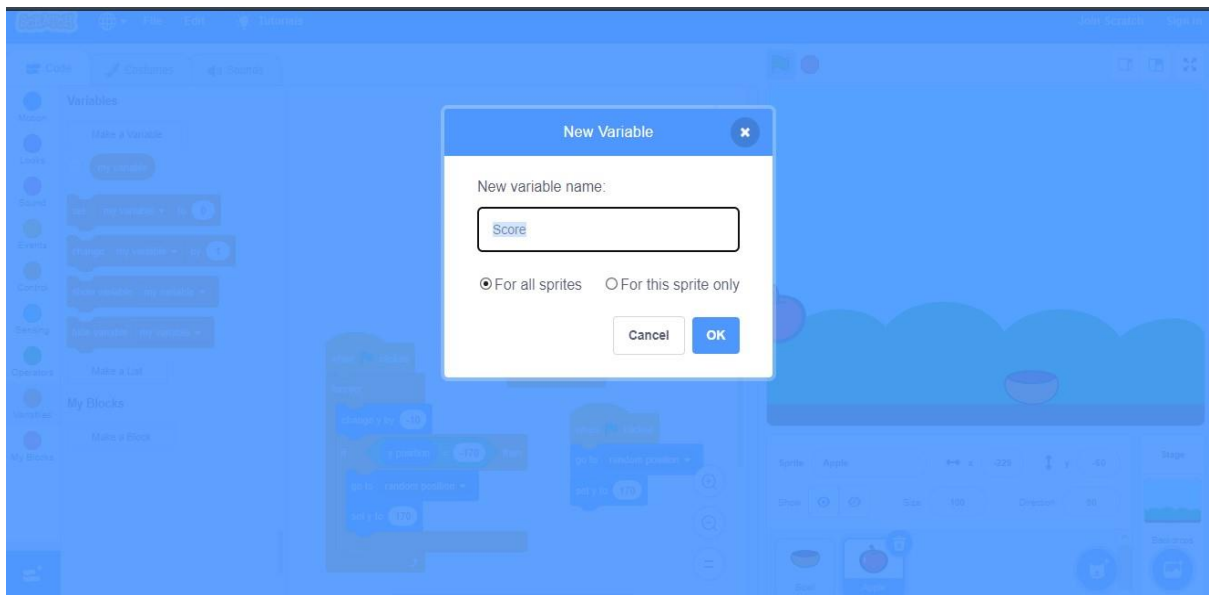


Fig. 25 (Make a Variable)

Change my variable to Score by 1 to make the score increase.

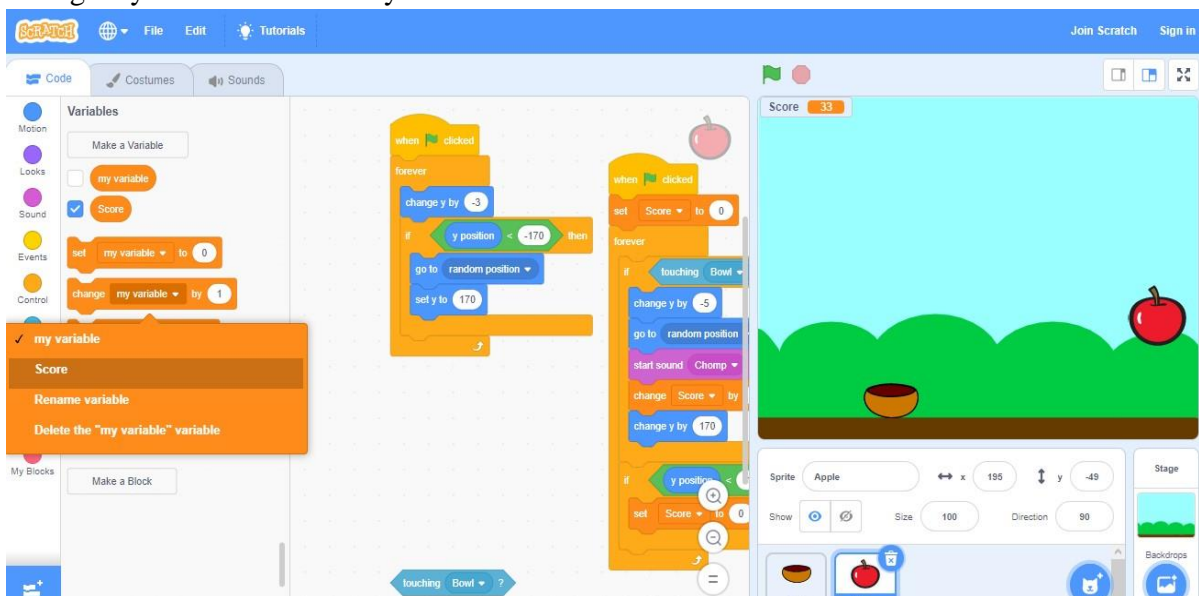


Fig. 26 (Increment Score by 1)

It should change score by 1 when it touches the bowl.

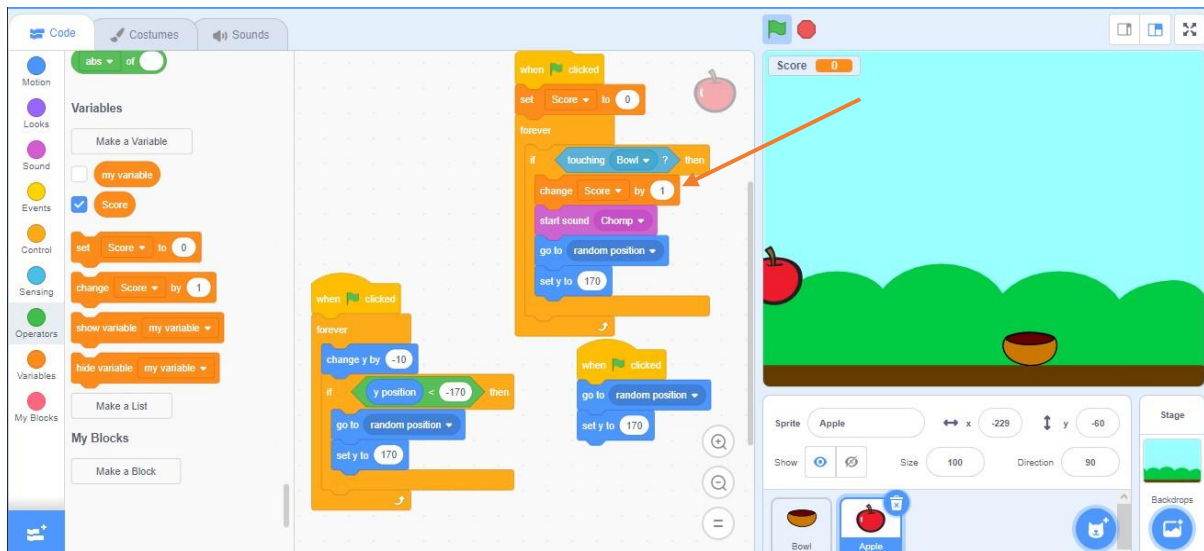


Fig. 27 (Catch Apple Game)

Also sets the score to zero at the start. By this when the apple will touch the bowl the score will be increase by 1.

So, this is the whole game.

Task 01: Catch Game in Scratch

[50 minutes]

1. Make apple catch game.
2. Save the file “rollno_game.sb3”.