

# Process

## Your code should

- Create 2 players – one controlled by W and S, the other controlled by the Up and Down arrow keys.
- Code the ball to bounce around the screen, bounce and interact with players
- Allow the users to score goals and earn points

# How does SCRATCH work?

The screenshot shows the Scratch web interface. On the left is the 'Code' editor with a 'Motion' category selected, showing various script blocks like 'move 10 steps', 'turn 15 degrees', 'go to random position', and 'change x by 10'. The center is a large text area with a grid background. On the right is the 'Stage' area showing a Scratch cat sprite. Below the stage is the 'Sprite' window, which is highlighted with a dashed purple border and contains a 'Sprite1' window with a cat icon and a 'Backdrops' window with '1' backdrop. A purple arrow points from the text 'Sprites can be found in the sprite window here!' to the 'Sprite1' window.

In Scratch, the way you interact with your program is through **sprites**

Sprites are objects within your program that you can interact with - you can change the way they act using **scripts**

In the template you have been given, many of the sprites have already been given basic scripts, such as the buttons

**Sprites can be found in the sprite window here!**

# How does SCRATCH work?

The image shows the Scratch IDE interface with several annotations:

- Code Section:** A black arrow points to the left sidebar containing various code blocks categorized by function (Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, My Blocks).
- Display Window:** A purple arrow points to the large white area on the right where the sprite (Scratch Cat) is visible.
- Sprite Window:** A purple arrow points to the bottom right area where the sprite's properties (name, x, y, size, direction) are displayed.

Text annotations within the image:

- "This is our display window, where our sprites will appear and interact with the user" (purple text, top right)
- "This is the code section! Here is where we can drag and drop blocks of code." (black text, middle left)
- "Blocks are colour coded depending on their function" (black text, middle left)
- "Sprites can be found in the sprite window here!" (purple text, bottom right)

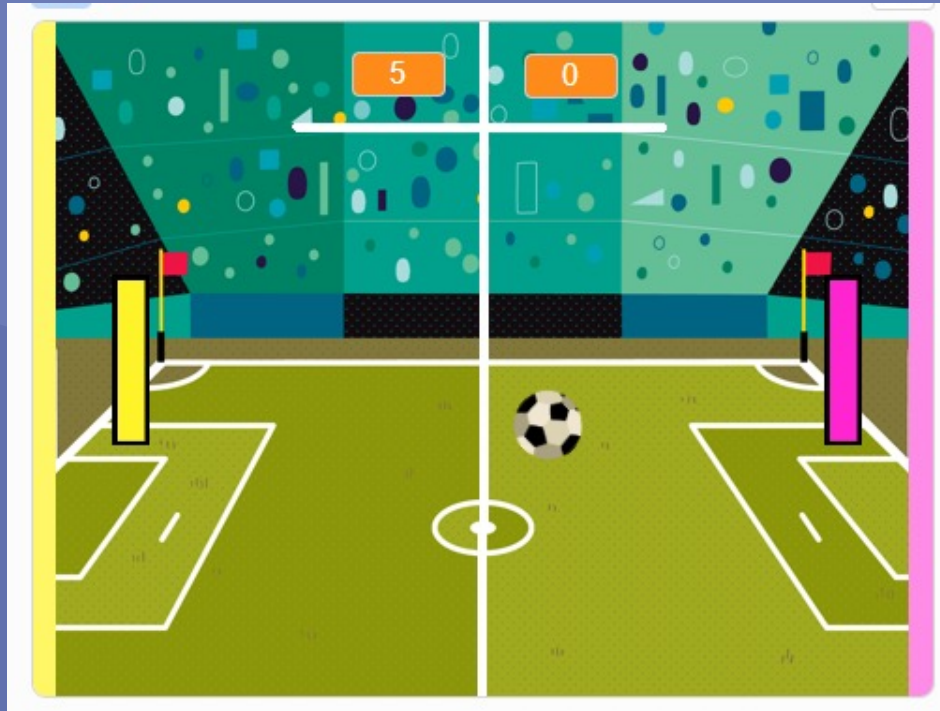
# How does SCRATCH work?

The image shows the Scratch web interface. At the top, there is a blue header with the Scratch logo, a globe icon, and menu items: File, Edit, Tutorials, and a text field containing 'Untitled'. To the right of the text field are 'Share' and 'See Project Page' buttons. On the far right of the header, there is a user profile icon and the name 'itsgracebennett'. Below the header, there are three tabs: 'Code', 'Costumes', and 'Sounds'. The 'Code' tab is selected. On the left side, there is a vertical sidebar with various categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The 'Motion' category is expanded, showing several blue code blocks such as 'move 10 steps', 'turn 15 degrees', 'go to random position', 'glide 1 secs to random position', 'point in direction 90', 'change x by 10', 'set x to 0', and 'change y by 10'. In the center of the workspace, there is a semi-transparent text box with the text: 'Use these tabs to switch between code, costumes and backdrops, and sound.' A black arrow points from this text box to the 'Code', 'Costumes', and 'Sounds' tabs. On the right side, there is a stage area with a white background and a small Scratch cat sprite. Below the stage, there is a control panel for the selected sprite, 'Sprite1', showing its x and y coordinates (0, 0), size (100), and direction (90). At the bottom right, there is a 'Stage' panel with a 'Backdrops' section showing 'Backdrops 1'. A purple dashed box highlights the 'Stage' and 'Backdrops' panels. A purple arrow points from the text 'Here we can interact with and customize the backdrops of our game' to the 'Backdrops' section. At the bottom of the interface, there is a 'Backpack' button.

Use these tabs to switch between code, costumes and backdrops, and sound.

Here we can interact with and customize the backdrops of our game

# What our game will look like...





The screenshot shows a Scratch project page for '2-Player Football Pong'. At the top left is the Scratch logo. The title '2-Player Football Pong' is in the center, with a 'Remix' button (green) and a 'See inside' button (blue) to its right. Below the title is a small thumbnail of the project. The main preview area shows a soccer field with a green flag in the center, a soccer ball, and two goalposts. To the right of the preview is an 'Instructions' section with the following text: 'Player 1: W and S', 'Player 2: Up Key and Down Key', '7 points to win!', and 'Enjoy <3'. Below the instructions is a 'Notes and Credits' section with a placeholder text: 'How did you make this project? Did you use ideas, scripts or artwork from other people? Thank them here.' At the bottom of the page, there are icons for likes (0), favorites (0), remixes (0), and views (1). The copyright notice '© Jan 16, 2022' is on the left, and 'Add to Studio' and 'Copy Link' buttons are on the right.

Once you have logged into Scratch...

Click [here](#) to access the template to our project!

Click **Remix** to get started with your project

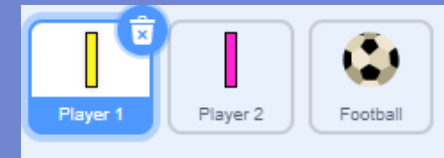
By the end of this resource, your project should look like [this!](#) (Use this if you get stuck)

# Step 1

## Coding player movement...

Inside of the Player 1 sprite, we want to set the initial position of our player and show the player's sprite when the green flag is clicked.

Then, using a forever block, we want to constantly check if the 'W' and 'S' keys are being pressed, and change the player's position accordingly.

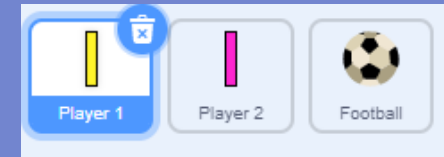


```
when green flag clicked
  show
  go to x: -190 y: 0
  forever loop
    if key w pressed? then
      change y by 10
    if key s pressed? then
      change y by -10
```

# Step 2

Coding player movement...

Inside of the Player 1 sprite, we want to set the initial position of our player and show the player's sprite when the green flag is clicked.



```
when green flag clicked
  show
  go to x: -190 y: 0
  if y position > 170 then
    set y to -170
  if y position < -170 then
    set y to 170
  forever
    if key w pressed? then
```



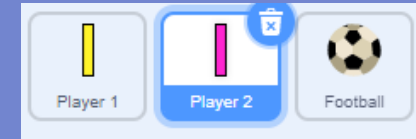
# Step 3

## Coding player movement for player 2...

We want to repeat this process for our Player 2 sprite, but we need to change the IF blocks slightly.

We need to change our co-ordinate values

We now need to check IF the 'up arrow' and 'down arrows' are pressed



```
when green flag clicked
  show
  go to x: 190 y: 0
  forever loop
    if y position > 170 then
      set y to -170
    if y position < -170 then
      set y to 170
    if key up arrow pressed? then
      change y by 10
    if key down arrow pressed? then
      change y by -10
```

# Step 4

Coding movement of the ball...



Inside of the football sprite, we need to set the ball's starting position and direction

Then, using a forever block, constantly move the ball 10 steps and make it bounce when it hits an edge

A vertical stack of Scratch code blocks on a dotted grid background. From top to bottom: a yellow 'when green flag clicked' block; a purple 'show' block; a blue 'go to x: 0 y: 0' block; a blue 'point in direction 45' block; an orange 'forever' loop block containing a blue 'move 10 steps' block and a blue 'if on edge, bounce' block. A small circular arrow icon is at the bottom right of the 'forever' block.

# Step 5

Coding movement of the ball...

Then we add two IF blocks that check, IF the ball touches the Player, it should turn 180 degrees.

At this point, the ball should bounce around the screen!



```
point in direction 45
forever
  move 10 steps
  if on edge, bounce
  if touching Player 1 ? then
    turn 180 degrees
  if touching Player 2 ? then
    turn 180 degrees
```

# Step 6

## Adding to the player's score...

In the Football sprite, we want to add some code that will check when the ball hits either goal.

If the ball hits Player 1's side, Player 2's score goes up by 1 – and vice versa.

At this point, the score system won't reset itself. This is what we'll fix next!



```

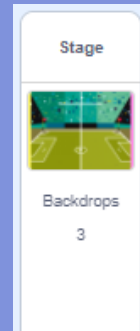
when green flag clicked
  forever loop
    if touching color yellow then
      change player2score by 1
      go to x: 0 y: 0
    if touching color pink then
      change player1score by 1
      go to x: 0 y: 0
  
```

# Step 7

## Deciding a winner...

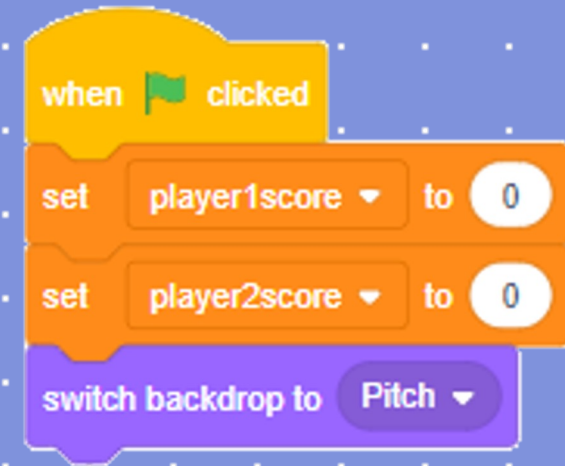
In order to decide a winner, we need to add some code to the backdrop section of our game.

To do this, select the Backdrop icon on the right hand side of the screen.



Add some code that will reset the screen when the green flag is clicked.

It will set the scores back to zero and make sure the correct backdrop is showing!




# Step 7

## Deciding a winner...

Then, using a forever block, we check to see if either player has a score of 7.

If they do, broadcast a 'game over' signal and set the backdrop to a celebration message!



```
switch backdrop to Pitch
forever
  if player1score = 7 then
    broadcast game over
    switch backdrop to Player 1 Win
  if player2score = 7 then
    broadcast game over
    switch backdrop to Player 2 Win
```

The image shows a Scratch script on a grid background. It starts with a purple 'switch backdrop to Pitch' block. Below it is an orange 'forever' loop block. Inside the loop, there are two conditional blocks. The first is an 'if' block with a green arrow-shaped condition 'player1score = 7'. Inside this 'if' block are a yellow 'broadcast game over' block and a purple 'switch backdrop to Player 1 Win' block. The second is another 'if' block with a green arrow-shaped condition 'player2score = 7'. Inside this 'if' block are a yellow 'broadcast game over' block and a purple 'switch backdrop to Player 2 Win' block. The script ends with a small white arrow at the bottom of the 'forever' loop.

# Step 8

## Clearing the screen...

When the player wins, we need to clear each sprite off of the screen. To do this, add this code into:

- The player 1 sprite
- The player 2 sprite
- The football sprite



# Extension

Making the game your own...

In the backdrops and sprite editors, have a go at experimenting with different backgrounds and images!

Could you make a space or battleship themed pong game?

Get creative!

To take it even further, could you find a way of adding sound affects?

