FLIP THE ROBOT MONKEY

SECTION 1
For the first full Construct 2 game you’re going to create a platform game called Flip the Robot Monkey.

Platform games involve a character moving across obstacles to reach an end goal. A couple of examples of this type of game would be Super Mario Bros. and, more recently, Ratchet & Clank. For your game you’ll have a main character named Flip going through a castle, avoiding knights and archers, and freeing his monkey friends.
STOP! READ THIS BEFORE CONTINUING!

Make sure you have completed the Flip Intro. Section 1 is dependent on the fact that you have completed those instructions.

You'll now start making your game. Make sure to open your Flip Intro game.
SECTION 1 OBJECTIVES

Before you start, read the section objectives below. These objectives will be completed at the end of this section.

- Setting Collision Points
- Create a Level
- Adding a Super Jump
OBJECTIVE 1 - SET COLLISION POINTS

Collision points and polygons allow you to indicate where on a sprite collisions will occur. Setting these will be helpful to prevent object from "catching" on platforms and will make animations look more smooth.
The first object you'll set the collision polygon for is the Flip object. In the Objects Bar double click on the Flip object to open the image editor.
When the image editor appears click the **Collision polygon** button. This button is the bottom button on the left toolbar.
A blue box and red points will now appear on the image. The edge of the blue box is where collisions with solid objects will occur. Anything outside of the box, like the tail, will ignore collisions.

By default Construct will guess the shape of the polygon. But for the Flip image you'll change it to be a rectangular box.
For the collision box you create you'll only need four points. To easily do this right click on the image and select **Set to bounding box**. This will set the collision polygon to the entire image.
Click on the top left point, this will select the point as indicated by it turning yellow. You can click and drag these points to their desired position or input the X and Y value right below the top toolbar. For these points you will input the X and Y values.
For the top left point set the X value to 17 and the Y keep at 0.
Set the X and Y value for the other 3 points starting at the top right and moving clockwise.

45, 0; 45, 100; 17, 100
To keep things consistent you'll want to set the same collision box for all the animations of the Flip object. Instead of doing this individually you can do it off of the one you already set.

Right click in the image editor and then select **Apply to all animations**. When the warning popup appears click **Yes** and your collision polygon will now be applied to all the animations.
Next you’ll set the collision points for the ground and platform. Open up the image editor for the Ground object first. You will notice that the Ground object has a front face and a top portion that will the area that you want to make Flip appear to be walking on.

Make sure you have the collision polygon button selected.
You should have 4 points on your image already. If not right click and select **Set to bounding box**.

You'll only need to change the location of the top points. Set the top left point to **0, 9** and the top right point to **572, 9**. Close the image editor when this is done.
Run the layout now and see that Flip appears to be on the top of the Ground object.

Open up the image editor for the platform object and you'll set the collision polygon for that next.

**STOP! READ THIS BEFORE CONTINUING!**

Don't forget to save your game, you wouldn't want to lose all your sweet stuff.
Use **Set to bounding box** to get four points. Now starting at the top left and moving clockwise set the collision point locations to the following:

- Top Left - 0, 6
- Top Right - 55, 6
- Bottom Right - 55, 40
- Bottom Left - 0, 40
If you run the layout now you'll see that flip moves smoothly on the platform now and appears to be walking on the top portion of it.
OBJECTIVE 2 - CREATE A LEVEL

Before you start to build out your level you'll want to rename your layout and event sheet. Renaming layout and event sheets is helpful to keep your project organized.
To rename the layout go to the Project Bar and in the Layouts folder right click on **Game** and click **Rename**.

The name will now appear as a text box and you can change the name. Type **Level1** to change its name.
Next you’ll rename the event sheet by right clicking on Event sheet 1 in the Projects Bar and selecting Rename. Change its name to esLevel.
Now that you have those renamed you’re going change the size of the layout. Click on **Level1** in the Projects Bar and then go to the Properties Bar on the left. In the Layout properties change the Layout Size to **2400, 720**.
Increasing the Layout Size will allow you to create a bigger level with more obstacles and enemies. But, if you remember back to when you first started the project you set the window size to 1280, 720. This means that only the first 1280 pixels of your level will be displayed. In order to see the rest of the level you’ll need to add a behavior to the Flip object so the window will follow him as he moves across the level.
Select your **Flip** object in the Objects Bar and in the Properties Bar click the **Behaviors** link. Click the **Plus** button and select the **Scroll To** behavior.
Next, you'll want to make sure that there's a surface across the entire layout. To do this you'll add copies of your Ground object. To help place these objects you're going to enable a grid on the layout. Grids are helpful for placing objects precisely on a layout.
At the top of the window click **View** to bring up the view tab.
The View options will appear and you'll see section for Grid options and Grid size. Here is where you'll set the width and height of the grid as well as its visibility and an option for object to be only set on grid intersection points.

STOP! READ THIS BEFORE CONTINUING!

Make sure that the Layout tab is active. If you're on the Event Sheet, you will not be able to change the grid options section.
Check both boxes in the Grid options.

Since the Ground object is 572 x 60 you'll use these values for the width and height. Set the Grid width to 572 and the Grid height to 60.
If you click on the layout you’ll see the grid now appears and you can easily add more ground sections in their proper places.

To make more Ground objects you’ll create new Instances of the object. Instances will share the same events, images, and behaviors. In order to make a new instance of the Ground object hold the Ctrl key and then click and drag the Ground object in the layout. If you drag to the right a new instance of the Ground object will appear in the next grid space. If you let go of the mouse the instance will be created.
Create 3 more instances (4 total) of the Ground object, placing each one in the grid space to the right of the last.
Once you have ground objects across the entire layout run the layout. You can see that the window will now follow Flip as he moves across the level.

You’ll also see that the background stops. To fix this you’ll have to increase the size of the Background object.
Select the **Background** object from the Object Bar. In the Properties Bar set its Size to **2400, 720**. The Background will now extend the entire layout.
Next you'll want to add more instances of the Platform object to build out your level. To help with this you can set the grid size to a different size. In the top toolbar set the Grid width to 55 and the Grid height to 40.
• Using instances of the Platform object to build out the level. Create some groups uses 2 or 3 instances of the Platform object. Arrow down for a reference of a completed level.
Arrow down for the 2nd half of the level
The last thing you'll want to do for the level is prevent Flip from going off the left and right of the layout. To do this you're going to add a solid object to left and right edges of the level.
On your layout add a new object. For the object type select **Sprite** and name the object **Edge**. When the crosshairs appear click on the left edge of the layout to bring up the image editor.
For this object you're just going to use a box with a filler color to make it easy to see in the editor. First you'll want to resize the object smaller. In the top toolbar click the **resize** button. It is a diagonal line with arrows on each end.
In the Resize image canvas, set the Width and Height both to 20. Click the OK button to resize the canvas.
Next you're going to fill tool to give the object a color and make it easily visible in the editor. On the left toolbar click the **Fill** button. It's the button that looks like a paint can.
This will bring up a Color Palette, click in the **upper left corner** of the palette to select a red color. Then click inside the canvas to make your object a red box.
Close the image editor to get to the layout. If still active, uncheck the Snap to grid box in the View toolbar.

You'll now want to set position of the Edge object so it's just outside of the layout. With the Edge object selected go to the Properties Bar and set its Position to (-10, 0).
Next, make sure the Edge object is still selected and resize the image so it covers the entire left edge of the layout. To do this click the bottom middle box surrounding the object and drag to the bottom of the layout.
You’ll want to create a new instance of the Edge object to cover the right edge of the layout. Hold the Ctrl key and click and drag the edge object to the right edge of the layout to create a new instance. Make sure it vertically covers the right edge and in its Properties Bar set its X value to 2410.
Next, you'll need to add a solid behavior to the object. Click on the **Edge** object in the Objects Bar and add the Solid behavior to the object.
Play your game now and notice that you no longer can walk off the left and right edge of the layout.

You have now added all that you need for your level at this time. Next, you’re going to add a super jump for Flip.

STOP! READ THIS BEFORE CONTINUING!

Yes, the Super Jump. Before you get too excited, save your game.
OBJECTIVE 3 - ADDING A SUPER JUMP

You'll now add a super jump to Flip. You'll make it so when you hold the down key the super jump will charge and when the up key is pressed it will enable the super jump.

To do this you'll have to use an instance variable and some events. To start, click on Flip in the Objects Bar.
Before you add the instance variable you'll want to change one of the values in the Platform Behavior. In the Properties Bar go down to the Behaviors section and in the Platform behavior change the Jump strength to 750.

<table>
<thead>
<tr>
<th>Behaviors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Platform</strong></td>
<td></td>
</tr>
<tr>
<td>Max speed</td>
<td>330</td>
</tr>
<tr>
<td>Acceleration</td>
<td>1500</td>
</tr>
<tr>
<td>Deceleration</td>
<td>1500</td>
</tr>
<tr>
<td><strong>Jump strength</strong></td>
<td>750</td>
</tr>
<tr>
<td>Gravity</td>
<td>1500</td>
</tr>
<tr>
<td>Max fall speed</td>
<td>1000</td>
</tr>
</tbody>
</table>
Next you're going to add an instance variable to Flip. An instance variable is a variable that is used in all the instances of that object, but each instance can have a different value for its instance variable.

In this case you're going to add a variable to store jump boost value. This value will be added to the Jump strength property to increase Flip's jumping ability.
With Flip still selected go to the Properties Bar and click the **Instance variables** link.
In the Instance variables window, click the **Plus** button to add a new variable.

This will bring up the New instance variable window. For the name type in **JumpBoost**. Keep the Type as **Number** and the Initial value as **0**. Click **OK** to insert the variable.
You can close the Instance variable window and go to your event sheet. Before you start to add events for the super jump you're going to start to organize your events using groups.

Groups are helpful to keep your event sheet clean and organized. Groups are also able to be activated or deactivated which is helpful to have events trigger at only specific times.
To add a group go to the bottom of the event sheet and right click on the **empty white space**. In the list that appears click on **Add group**.
In the Edit event group window that appears set the Name field to **Flip Controls/Collisions** and the Description to **Contains keyboard controls and collision events for Flip**.

Keep the box check for Active on start and click **Ok** to add your group.
Using the same method add another group. This one named **Flip Animations** with a description that says **Sets Flip's animations**. This one will also be active on start.
Events in the event sheet can be moved and rearranged by dragging and dropping. To do this, click on the left edge of your first event and drag it down to the Flip Animations group. When you move the event below the group you'll see an indented black line appear. This indicates that the event you're dragging become a sub-event of the event above that line. Release the mouse to move your event to the Flip Animations group.
Your Flip Animation group will now have an event nested below.
Move your other events, except the start of layout event and Flip Controls group, into the Flip Animations group. Make sure as you move the event that they become nested under the Flip Animations group and not as a sub-event of another event.
**IT SHOULD LOOK LIKE THIS WHEN YOU'RE DONE**

<table>
<thead>
<tr>
<th>Event</th>
<th>Action</th>
<th>Animation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>On start of layout</td>
<td>Flip</td>
<td>Set animation to &quot;Default&quot; (play from beginning)</td>
</tr>
<tr>
<td>Flip</td>
<td></td>
<td>Move to top of layer</td>
<td></td>
</tr>
</tbody>
</table>

**Flip Controls/Collisions**
Contains keyboard controls and collisions event for Flip

**Flip Animations**
Sets Flip's animations

<table>
<thead>
<tr>
<th>Event</th>
<th>Action</th>
<th>Animation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right arrow is down</td>
<td></td>
<td>Flip</td>
<td>Set Not mirrored</td>
</tr>
<tr>
<td>Keyboard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left arrow is down</td>
<td></td>
<td>Flip</td>
<td>Set Mirrored</td>
</tr>
<tr>
<td>Flip</td>
<td>Platform On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>jump</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flip</td>
<td>Platform On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>landed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flip</td>
<td>Platform is</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moving</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flip</td>
<td>Platform On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fall</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flip</td>
<td>Platform is</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moving</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flip</td>
<td>Platform On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>moved</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
With your events organize you’re ready to add the events for the super jump. These event will be added to the Flip Controls group, to do this right-click on the group and go to add and then select **Add sub-event**.
This will bring up the normal Add event window. For your object select **Keyboard** and for the condition select **Key is down**.
In the Parameters window, click the **click to choose** button, press the **Down arrow** key, then the **Ok** button, and finally the **Done** button.

This will add your new event in the Flip Controls group that tests if the down arrow is pressed.
You'll next add two more conditions to this event. One will test if the JumpBoost variable is less than 400 to prevent Flip from being able to jump too high. The other will test if Flip is on the Ground or Platform object.
To add an additional condition to your event right click on the event, go to Add, and then press **Add another condition**.
You’ll use this condition to test the JumpBoost value. Select the **Flip** object and for the condition select **Compare instance variable** in the Instance variables section.
Here you'll test that the JumpBoost Value isn't higher than 400. To do this keep the Instance variable field as **JumpBoost** change the Comparison to **Less than** and set to value as **400**. Click **Done** to add the condition.
Add another new condition to the event and again use the **Flip** object. For the condition select **Is on floor** under the Platform section.
Next, you'll add two sub-events to this event. One will test to see if flip is not moving and if he isn't you'll increase the JumpBoost. The other will test if Flip is moving and set the JumpBoost to 0.

Right click on your event, go to Add, and then select Add sub-event.
For your object select **Flip** and for the condition select **Is moving** under the Platform section.
When your sub-event is inserted, right click on the condition and select **Invert**.

Now you’re ready to add the action to increase the JumpBoost. On the new sub-event add an action that uses the **Flip** object and the action **Add to** which is in the Instance variables section.
When your Parameters window appears keep the instance variable as **JumpBoost** and set the value to **5**. Click **Done** to insert your action.
Add another new sub-event to the Keyboard Down arrow is down event. The condition for this one will again use the **Flip** object to test if it is moving.
With your sub-event added, add a new action to it. Use the **Flip** object and for the action select **Set value** in the Instance variables section.

Click **Done** when the Parameters window appears as it's already is setting JumpBoost to 0.
Next, you’ll set up the events that will execute your super jump and remove the super jump if Flip moves before he jumps.

Add a new sub-event off of the Flip Controls group. This event will use the **Keyboard** object and the condition **On any key pressed**.
With this event created you'll add another condition to it that will test if JumpBoost has a value. To do this, add a new condition to your new event that uses the **Flip** object and the **Compare instance variable** condition.
For your Parameters keep Instance variable as **JumpBoost**, change Comparison to **Greater than**, and keep the Value at **0**. Click **Done** to add your condition.
You'll trigger the actions for this event off of sub-events. Add a new sub-event to your newly created event that uses the **Keyboard** object and the **Key is down** condition. Set the Key value to **Up Arrow** and click **Done**.
This event will trigger your super jump so you'll add the actions to do that now. Add a new action that uses the **Flip** object and under the Platform section select the **Set jump strength** action.
In the parameters window you’re going to use your instance variable for the expression. Expressions are used in parameters to calculate sums or to retrieve information form objects. In this case you're going to use expressions to get the value of the JumpBoost variable and add it to the Jump strength value.

There are two ways to input object expressions into your parameter. Either through typing or through the Expressions panel. You'll use both for this parameter and from there you can decide which method you prefer.
First, you'll use the expression panel to get the JumpBoost variable. The expressions panel will be transparent above the parameters window. You want to access the JumpBoost instance variable on the Flip object, so in the window double click Flip and scroll down to find the JumpBoost instance variable and double-click it. This will cause the expression to input into the Jump Strength field in the parameters window.
Next, you'll add the Jump strength value to this. Click on the Jump strength field so your cursor is after the JumpBoost. Put a + after this and then start to type Flip. You'll see a window appear listing some expressions and as you type Flip will become highlighted. If you hit the Enter key it will complete Flip for you.
After Flip insert a . to be able to access expressions within the Flip object. Access the Jump strength by inputting `Platform.JumpStrength`. Your full completed expression will look like this.

```
```

Click **Done** to insert your action.
On this same sub-event you're going to add an event that sets the JumpBoost to 0. Instead of adding an entirely new action you'll see how to copy an existing action from another event.

In your event above locate the action that sets JumpBoost to 0.
Select the set JumpBoost to 0 event by clicking it. Now holding the Ctrl key, click and drag the action so a black line appears below the action that sets the jump strength. Release the mouse button when this appears to copy the action.
You can now run your layout and test your super jump. You can see that pressing and holding down then pressing up will cause your jump to be greater than normal, but a few issues are seen.

Currently the JumpBoost gets stacked, so each time you super jump it adds more to the jump strength. Also you can currently press down to get a jump boost, move, and then press up and still get a super jump. You want it set so the super jump goes away if Flip moves.
First, you're going to address the issue of the JumpBoost not resetting when you move. To do this you'll add two new sub-events to the On any key pressed event.

Add a new sub-event this event that uses the **Keyboard** to test if the Down Arrow key is down.
Now you'll add another sub-event for if the left or right arrows or spacebar is pressed. Instead of adding separate conditions for each key you can use an Else event. Else events will fire if conditions in previous events are not met. In this case if the JumpBoost is greater than 0 and a key is pressed that isn't the Up or Down arrow keys then this event will fire.
To add the Else event right click on the last sub-event you added (Down arrow is down) and go to Add and select **Add 'Else'**. You'll see that a new sub-event will be added that uses the System object and the else condition.

Add an action to this sub-event that sets the JumpBoost to 0.
You have all the events and action set in the Flip Controls group for now. To address the issue of the jump strength not resetting you'll add an action to an event in the Flip Animations group.

STOP! READ THIS BEFORE CONTINUING!

Dinosaurs are extinct, but that doesn't mean saving has to be.
Go to the Flip Animations group and find the event for Platform On landed (It will have two sub-events).

Add an action to this event that uses the Flip object and will Set jump strength to 750.
If you run the layout you'll now see that you have a super jump fully set up for Flip.

But jumping around can get boring, so next you'll add some bad guys for Flip to defeat!
SUCCESS!

YOU'VE JUST FINISHED THIS SECTION.