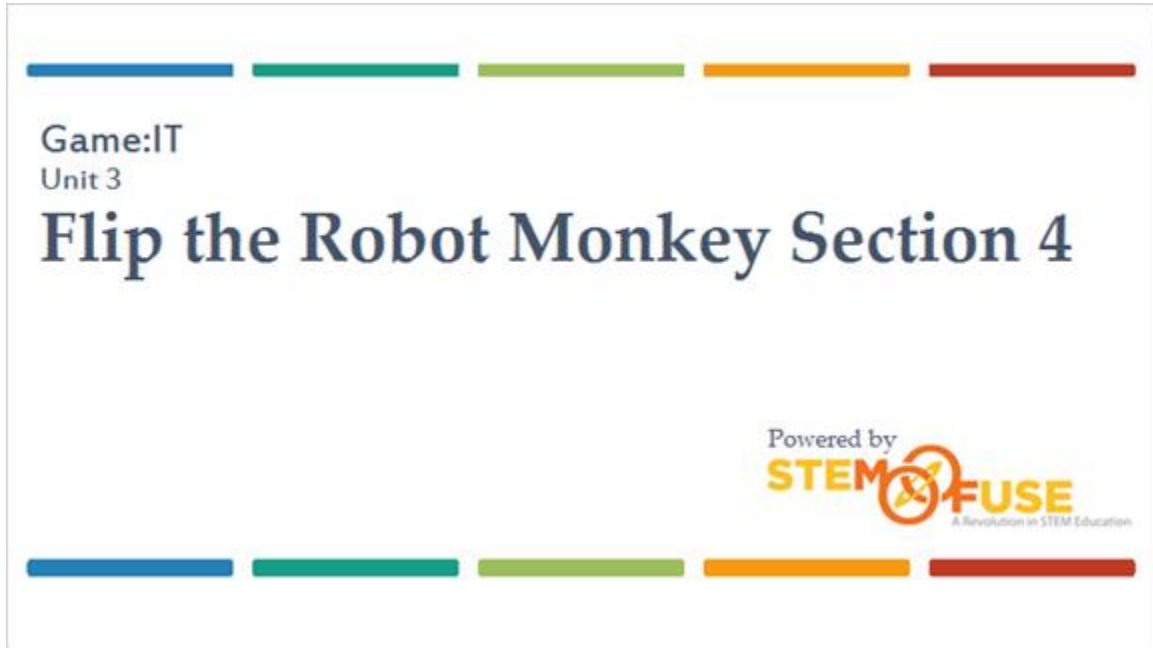


Flip the Robot Monkey Section 4

1 Title Slide: Flip the Robot Monkey Section 4



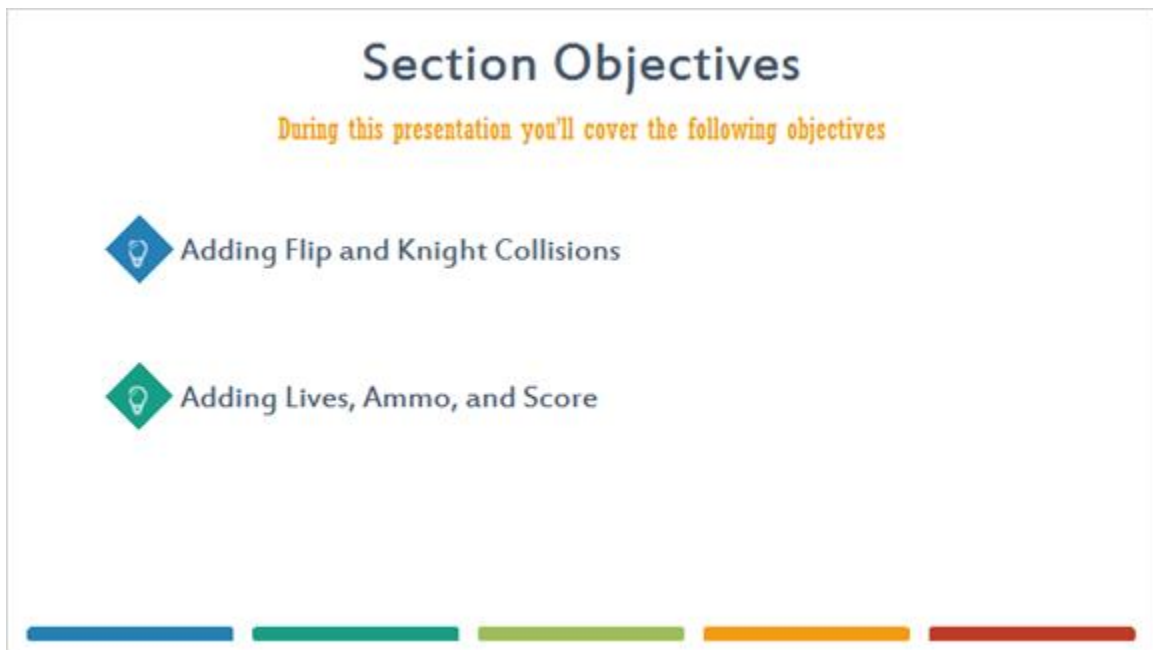
Game:IT
Unit 3

Flip the Robot Monkey Section 4

Powered by
STEM & FUSE
A Revolution in STEM Education

The slide features a decorative border at the top and bottom consisting of five horizontal bars in blue, green, light green, orange, and red. The text is centered and includes the game title, unit number, and the STEM & FUSE logo.

2 Section Objectives



Section Objectives

During this presentation you'll cover the following objectives

- Adding Flip and Knight Collisions
- Adding Lives, Ammo, and Score

The slide features a decorative border at the bottom consisting of five horizontal bars in blue, green, light green, orange, and red. The text is centered and includes the section title, a subtitle, and two bullet points with lightbulb icons.

3 Objective 1: Adding Flip and Knight Collisions

Objective 1

Adding Flip and Knight Collisions

The final thing you'll do is set up the events for when Flip and the Knight collide.

Add a new sub-event to the Flip Controls group. Select **Flip** as the object, and for the condition, select **On collision with another object**. For the Object parameter, select **Knight**.

4 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

◇ You'll first set up the event for if the Knight is stunned. Add a sub-event to your new event that checks if the Knight's Action variable is equal to "stun".

Show Knight Stun Action



Objective 1 – Adding Flip and Knight Collisions



Knight Stun Action Image (Slide Layer)

On Your Own

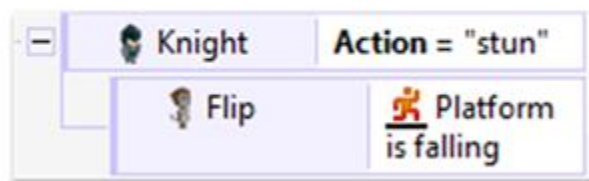
Complete all the tasks before continuing to the next slide



5 Flip Is Falling

Next, you'll want to test if Flip is falling when the collision happens. You'll do this as a sub-event of the last sub-event you added.

Add a sub-event to the last sub-event that uses the Flip object and the Is falling condition under the Platform section.



Objective 1 – Adding Flip and Knight Collisions

6 Flip Compare Y

Now you'll add a condition to this event to test if Flip is above the Knight when the collision occurs. Remember, the Y-values in Construct start with 0 and increase as you move down.

Add a condition to the sub-event that uses the **Flip** object and the **Compare Y** condition under the Size & Position section. For the Comparison, select **Less than**.

Objective 1 – Adding Flip and Knight Collisions



7 Flip Compare Y

For the Y coordinate, type in **Knight.Y-40**. Since the Y coordinate of the Knight is based on its origin point, you need to remove 40 to get the co-ordinate near the top of the image.

Click **Done** to add the condition.



Objective 1 – Adding Flip and Knight Collisions



8 Knight Set Vector Y

Next, you'll add actions to the event where your condition was just added. The first action will destroy the Knight. Add this action by using the **Knight** object and the **Destroy** action.

You'll want to add an action that will simulate Flip bouncing up off the Knight. To do this, add an action that uses the **Flip** object and the **Set vector Y** action in the Platform section.

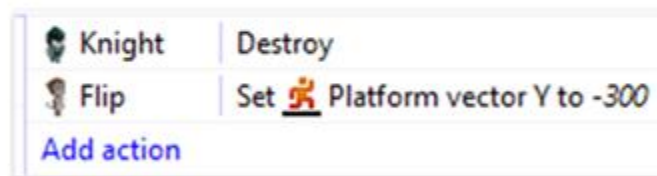


Objective 1 – Adding Flip and Knight Collisions

9 Flip Set Platform Vector Y

The Y vector will set the vertical motion of Flip, with positive values setting the motion down and negative values setting the motion up.

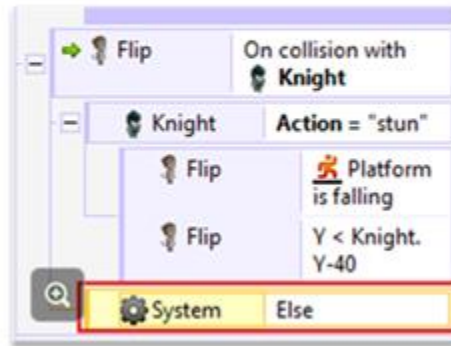
For your Vector Y field, set the value to **-300**. Press **Done** to add the action.



Objective 1 – Adding Flip and Knight Collisions

10 System Else

Next, you'll set up the events for when Flip collides with the Knight when it's not stunned. Right-click on the event that tests if Action is equal to stun, go to add, and select **Add 'Else'**.



Objective 1 – Adding Flip and Knight Collisions

11 Flip Set Ignoring Input

The first action you'll want to set on this event is one that will disable Flip from moving. So, add a new action to your Else event that uses the **Flip** object and under the **Platform** section select the **Set ignoring input** action.

Change the Input parameter to **Start ignoring** and click **Done**.



Objective 1 – Adding Flip and Knight Collisions

12 Knight Set Max Speed

The next thing you'll want to do is stop the Knight from moving. Add an action that uses the **Knight** object and the **Set max speed** action under the Platform section. Set the Max Speed parameter to **0** and click **Done**.



Objective 1 – Adding Flip and Knight Collisions

13 Knight Actions

On Your Own

Complete all the tasks before continuing to the next slide

- ◇ Add an action to the Else event to set the Knight's Action to "swing"
- ◇ Add an action that sets the Knight's animations to "Swing"

Show Knight Actions

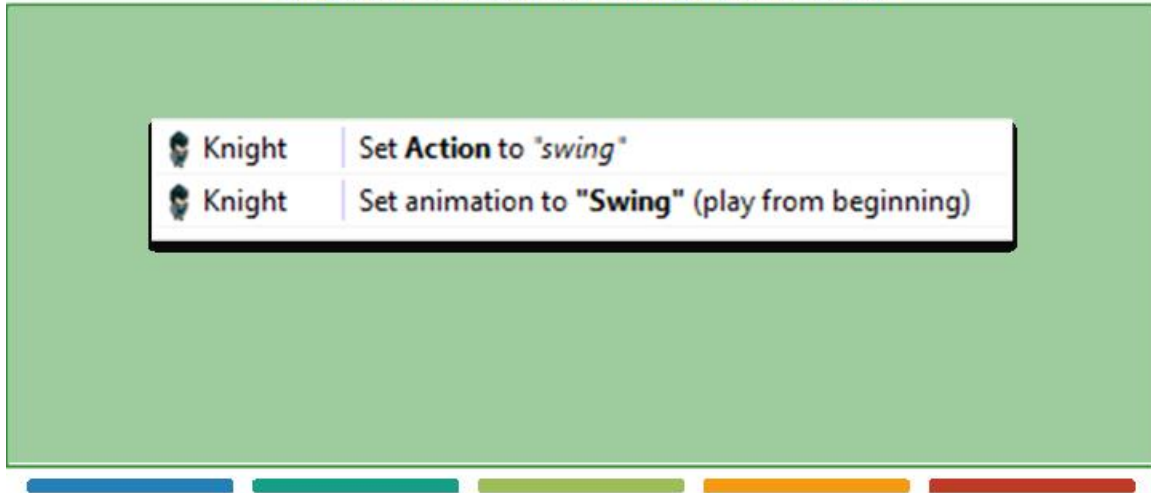


Objective 1 – Adding Flip and Knight Collisions

Knight Action Image (Slide Layer)

On Your Own

Complete all the tasks before continuing to the next slide

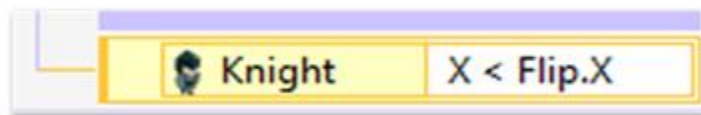


The screenshot shows a green workspace with a white task list box. The first task is 'Knight' with the instruction 'Set Action to "swing"'. The second task is 'Knight' with the instruction 'Set animation to "Swing" (play from beginning)'. Below the workspace is a progress bar with five colored segments: blue, green, light green, orange, and red.

14 Knight Compare X

The last thing you'll want to do is set up a couple sub-events to make sure the Knight is facing the direction of Flip. Add a sub-event to the Else event that uses the **Knight** object and the **Compare X** condition.

In the Parameters window, set Comparison to **Less than** and the X coordinate to **Flip.X**.



Objective 1 – Adding Flip and Knight Collisions



15 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

- ◇ Add an action to the last created sub-event that sets the Knight to Not mirrored
- ◇ Add a sub-event off of the Else event that tests if the Knight's X value is greater or equal to Flip's X
- ◇ Add an action to this sub-event that sets the Knight to Mirrored



Objective 1 – Adding Flip and Knight Collisions





Show Knight Actions




Knight Actions Image (Slide Layer)

On Your Own

Complete all the tasks before continuing to the next slide

	Add action
 Knight $X < \text{Flip.X}$	 Knight Set Not mirrored
	Add action
 Knight $X \geq \text{Flip.X}$	 Knight Set Mirrored
	Add action



16 Example Level

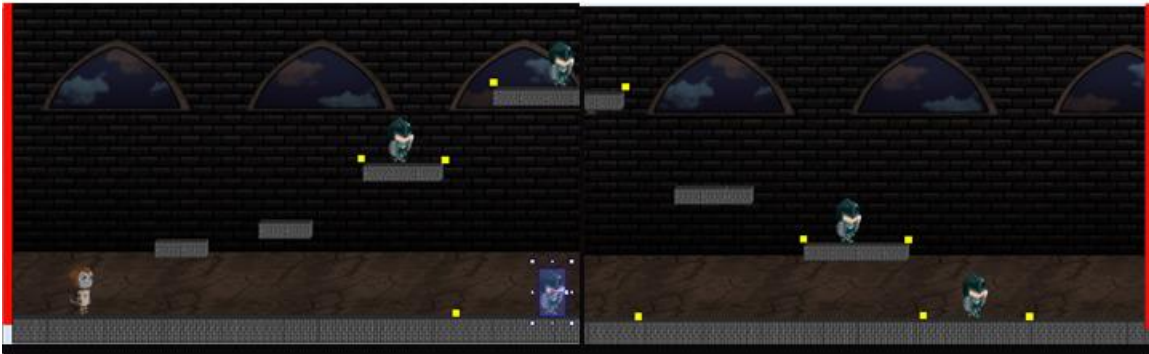
Go to your layout and add more instances of the Knight. It is recommended to have 4 or 5 instances. Make sure to include EdgeMarkers to contain your Knights.

The button below will show an example of an image.

Click to Show Level

Objective 1 – Adding Flip and Knight Collisions

Level (Slide Layer)



17 Testing Your Layout

Run your layout a few times to see how you can now defeat Knights and how they will defeat Flip.

The next thing you'll want to do is add a few key features to the game. These would be a score, lives, and ammo for the coconut gun. As we did previously, we will add the necessary objects to the game and then move on to the events.

Objective 1 – Adding Flip and Knight Collisions

A horizontal progress bar with five segments of different colors: blue, green, light green, yellow, and red. The first two segments (blue and green) are filled, indicating progress.

18 Objective 2: Adding Lives, Ammo, and Score

Objective 2

Adding Lives, Ammo, and Score

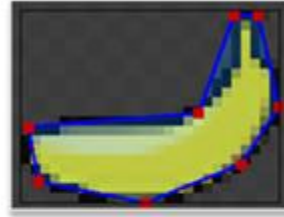
In many platform games, the main character will collect coins as a way to earn points. But since Flip is a Robot Monkey that would have no use for coins, you're going to use bananas. As Flip moves through the level, he will collect bananas and earn points.

On your layout, add a new Object that is a **Sprite** and name it **Banana**.



19 Banana Collision Polygon

When the cross-hairs appear, click above the first platform in your level to bring up the image editor. In the image editor, click the **Open** button and select the **Banana** file to import your banana image.



The only other thing you'll need to do in the image editor is fix the collision polygon. Click on the **Collision Polygon** button in the left toolbar. Click and drag the points until you have the polygon roughly around the banana. Close the image editor when this is done.

Objective 2 – Adding Lives, Ammo, and Score



20 Ammo Sprite

The Banana object won't need any other behavior or property changes. So, in your layout, add more instances of your Banana throughout the level.

Next, you'll add the object for the coconut gun ammo. Add a new object that is a **Sprite** and name it **Ammo**. Click above the first platform in your level with the cross-hairs to bring up the image editor.

Another example of the layout with Ammo and Banana objects placed will be provided after more elements are added to the layout.

Objective 2 – Adding Lives, Ammo, and Score



21 Adding Ammo

In the image editor, click **Open** and select the **Ammo** file. Close the image editor when the ammo image is imported.

In the layout, add one or two more instances of the Ammo object to your level.

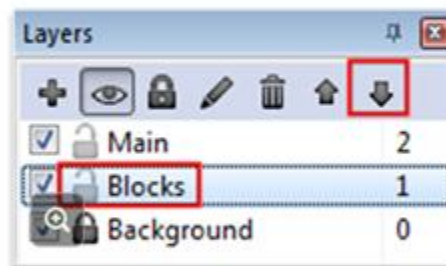
When you add the score, lives, and ammo to your game, you're going to want to display them on the screen. To do this, you're going to add a new layer to your game. This layer will be for an HUD, or heads up display. This is found in many games and is used to display relevant information for the player.

To begin, click on the **Layers** tab on the right side of the screen.

Objective 2 – Adding Lives, Ammo, and Score

22 New Layer Blocks

Click the **plus** button to add a new layer. **Rename** the layer to **Blocks** and click it to make it active. Next, click the **Down arrow** button in the Layers Bar to move this layer below the Main layer.



Objective 2 – Adding Lives, Ammo, and Score

23 Parallax Settings

Now we're going to add a heads-up display (aka HUD). It's the interface that shows the player's health, score and other information in-game. First, add another new layer and **Rename** it **HUD**. This layer can stay on the top. Click it to make it active and go to the Properties Bar. Change the value of Parallax to **0, 0**.

Parallax allows different layers to scroll at different rates for a sort of semi-3D effect. If we set the parallax to zero, though, the layer won't scroll at all, which is ideal for an HUD.

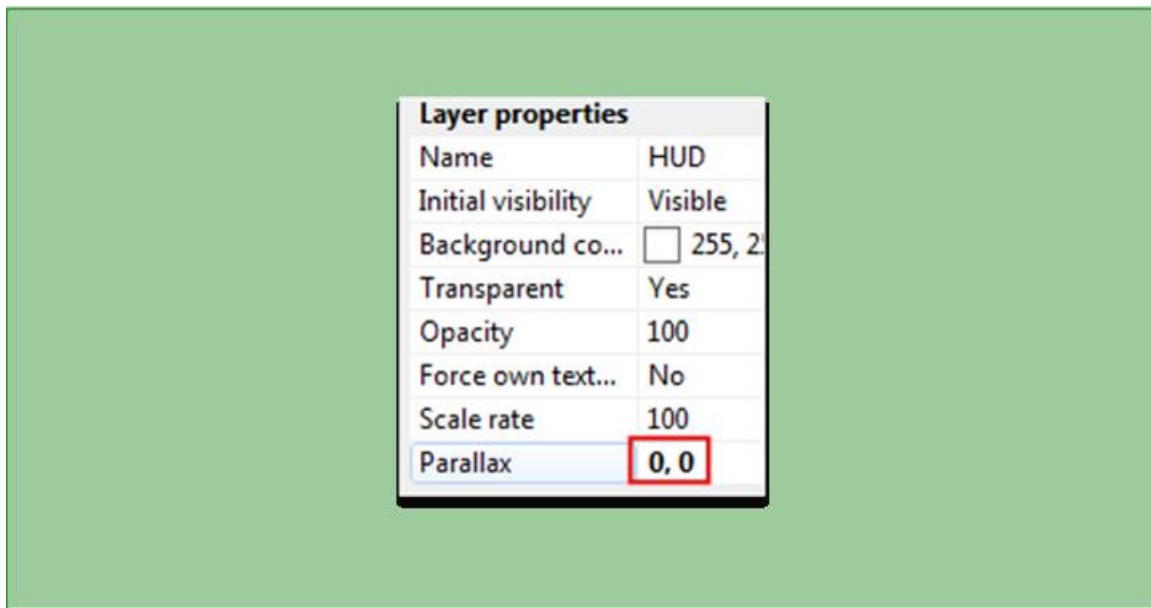
To learn more about Parallax Scrolling,

[Click Here](#)

[Show Parallax Settings](#)

Objective 2 – Adding Lives, Ammo, and Score

Parallax Settings Image (Slide Layer)



24 Background Parallax

In the Layers Bar, click on the **Background** layer, and then in the Properties Bar change its Parallax value to 75, 75.

The higher the value you enter, the faster that layer appears to scroll.

Opacity	100
Force own text...	No
Scale rate	100
Parallax	75, 75
Editor properties	

Objective 2 – Adding Lives, Ammo, and Score



25 Moving Objects to Different Layers

Before you add the objects for your HUD, you're going to change the layer of a couple of objects. With a layout tab selected (not an event sheet tab), click on the **Ground** object in the Objects Bar or the layout itself. In the Properties Bar, change its Layer to **Blocks**. This will make sure the Flip and Knight objects appear on top of the ground. Do the same for the **Platform** object as well.

Common	
Layer	Blocks
Angle	0
Opacity	100

Objective 2 – Adding Lives, Ammo, and Score



26 Text Objects

Make sure your active layer is HUD and add a new object to your layout. Select Text as the type and name the object TxtLives.



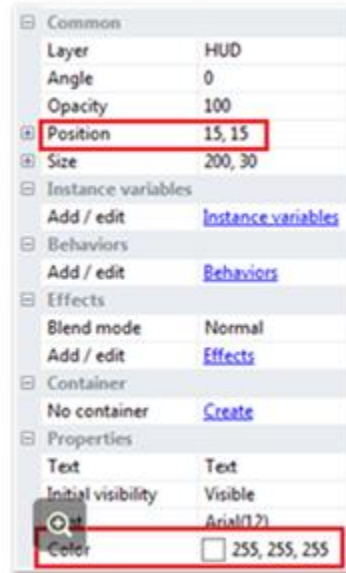
Click in the upper left of your layout when the cross-hairs appear to insert the object.

To learn more about the Text Object, [Click Here](#)

Objective 2 – Adding Lives, Ammo, and Score

27 Text Object Settings

With the object selected, go to the Properties Bar and change its Position to (15, 15). Next, go to the Properties section and change the Color setting to **White**.



Objective 2 – Adding Lives, Ammo, and Score

28 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

- ◇ Add two new Text objects; one named TxtAmmo, the other TxtScore
- ◇ Set the position of TxtAmmo to (1100, 15) and TxtScore to (1100, 50)
- ◇ Change both of their Color properties to White

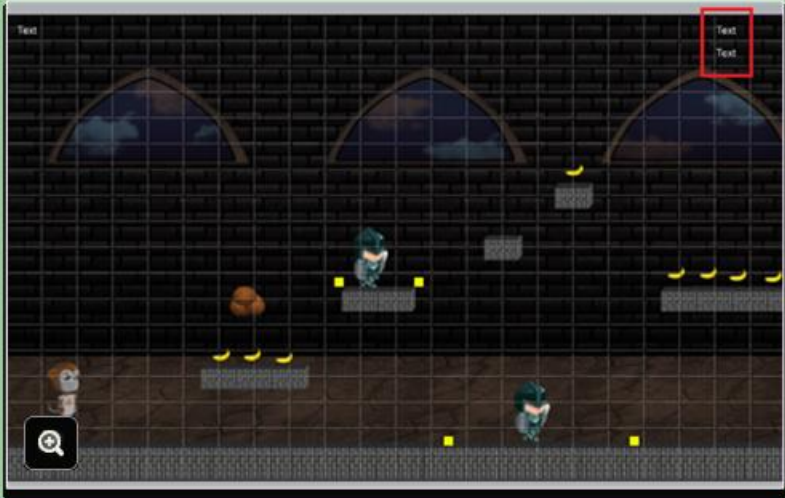


Objective 2 – Adding Lives, Ammo, and Score

Show Level Example

Level Example (Slide Layer)

On Your Own



The screenshot shows a game level with a grid overlay. A red box highlights a 'Text' object in the top right corner. The game scene includes a character, a tree, a moon, and various platforms.

29 Level Example

Since your newly created TxtAmmo and TxtScore object will be displayed on the right side of the screen, you'll want to change their text alignment to the right.

In the Properties Bar, under the Properties section, set the Horizontal alignment of both to **Right**.

Font	Arial(12)
Color	<input type="checkbox"/> 255, 255...
Horizontal alignment	Right
Vertical alignment	Top

Show Level Example

Objective 2 – Adding Lives, Ammo, and Score

Level Example Image (Slide Layer)



30 TxtAmmo Object

Now you're ready to set up your events. The first thing you'll want to do is set it so the TxtAmmo and TxtScore objects are on the right side of the screen. Since many screens are not the same size, you'll want to set it so these object will be positioned based off the screen size.

Go to the event sheet and add a new action to the Start of layout event. Select the **TxAmmo** object, and for the action select **Set X** under Size & Position.

Objective 2 – Adding Lives, Ammo, and Score



31 Viewport Right

For the X value, you're going to use the expression ViewportRight. This will return the X value of the right side of the viewport boundaries of a layer. In this case, you'll use ViewportRight to set the X position to 215 pixels less than the right edge. The three within the parentheses refers to the layer number. In this case, the 3 refers to the HUD layer.

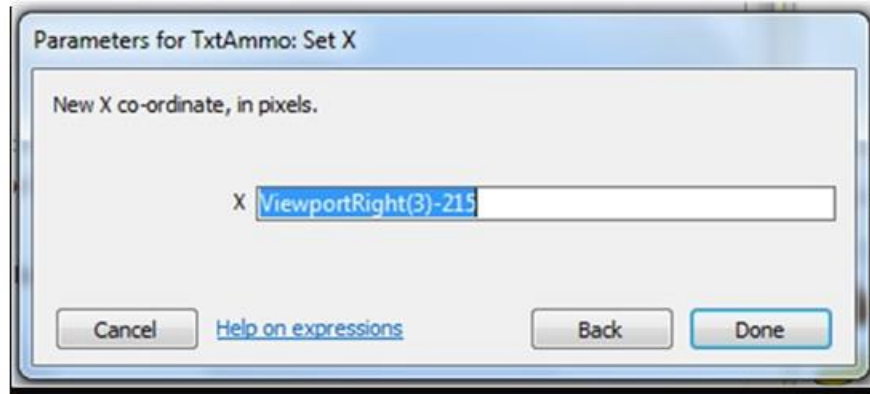
In the X field input `ViewportRight(3)-215`.

Show Parameters

Objective 2 – Adding Lives, Ammo, and Score

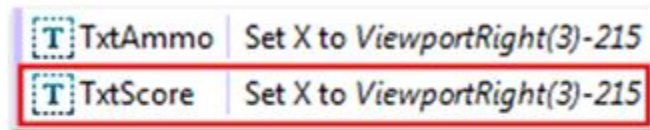


Parameters Image (Slide Layer)



32 Adding Global Variables

On the Start of layout event, add another action that sets the X of TxtScore to `ViewportRight(3)-215`.



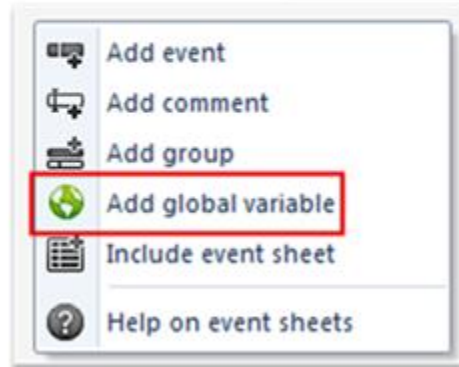
You're now going to add the score, ammo, and lives to the game as Global Variables. Global variables differ from instance variables in that they aren't tied to any certain object and can be used across your game. Unlike instance variables, they can only have one unique value, whereas instance variables can be different for different instances of an object.

Objective 2 – Adding Lives, Ammo, and Score

33 New Global Variable

To add a global variable, right-click on the blank space of the bottom of your event sheet and select **Add global variable**.

To learn more about Global Variables, [Click Here](#)

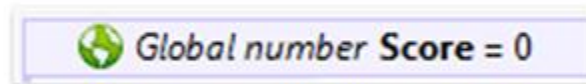


Objective 2 – Adding Lives, Ammo, and Score

34 Score Global Variable

Name your variable **Score**. Keep the Type as a **Number** and the Initial value as **0**, then click **OK** to add your variable.

You'll see that the variable has been added to the top of the event sheet and is ready to be used.



Objective 2 – Adding Lives, Ammo, and Score

35 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

- ◇ Add a new global variable named AmmoAmt. It will be a Number with an Initial value of 3.
- ◇ Add another global variable named Lives. It will also be a number with an Initial value of 3.

Show Global Variables



Objective 2 – Adding Lives, Ammo, and Score

Global Variables Image (Slide Layer)

On Your Own

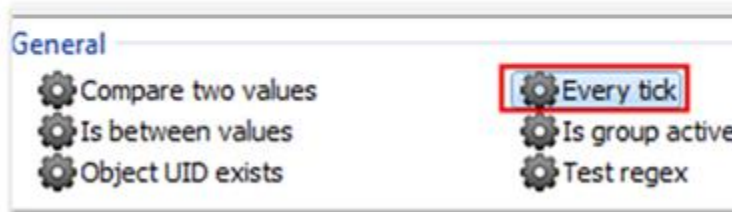
Complete all the tasks before continuing to the next slide

 Global number Lives = 3
 Global number AmmoAmt = 3
 Global number Score = 0

36 System Every Tick

With the variables added, you can set the events that will display the values in the text objects. You'll do this using an event that will run every tick. A tick will occur about 60 times per second, so essentially, this will run all the time.

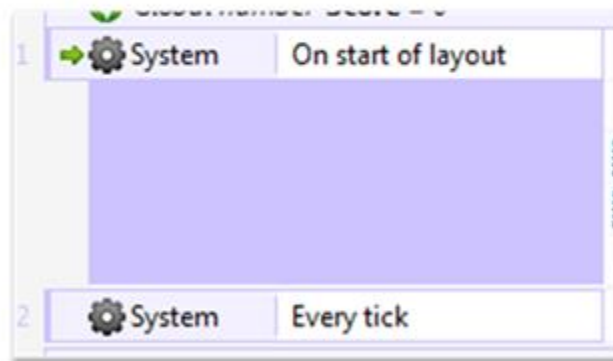
Add a new event that uses the **System** object and the **Every tick** condition.



Objective 2 – Adding Lives, Ammo, and Score

37 Moving Events

To keep your event sheet organized, move the Every tick right below the Start of Layout event.



Objective 2 – Adding Lives, Ammo, and Score

38 TxtLives Set Text

Now add an action to this event that uses the **TxtLives** object and the **Set Text** action.

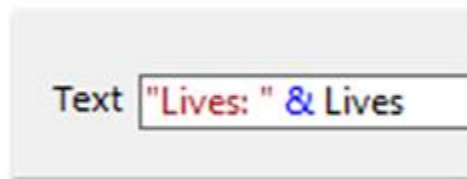
In the parameters, you're going to use the first instance of combining strings and a number in an expression. These can be combined by using an **&** symbol.



Objective 2 – Adding Lives, Ammo, and Score

39 Text Setting

In the Parameters window, set the Text field to `"Lives: " & Lives`. Note the space after the colon before the quotes closes and that the second Lives used is the Lives variable. Click **Done** to insert the action.



Objective 2 – Adding Lives, Ammo, and Score

40 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

◇ Add two more actions to the every tick event. One will set the text of TxtAmmo to "Ammo: " & AmmoAmt. The other will set the text of TxtScore to "Score: " & Score.



Show Every Tick Event

Objective 2 – Adding Lives, Ammo, and Score




Every Tick Image (Slide Layer)

On Your Own

Complete all the tasks before continuing to the next slide

System	Event	Action	Description
System	Every tick	T:TxtLives	Set text to "Lives: " & Lives
		T:TxtAmmo	Set text to "Ammo: " & AmmoAmt
		T:TxtScore	Set text to "Score: " & Score



41 Score Action

Now you'll just need to set up the event that will change the values of the global variables.

Go into the Flip Controls group, and in the Flip on collision with Knight event, find the action that Destroys the Knight. Add a new action that uses the **System** object and the **Add to** action under the Global & local variables section. Set the Variable to **Score** and the value to **100** and click **Done** to add the action.

Show Score Action

Objective 2 – Adding Lives, Ammo, and Score

Score Action Image (Slide Layer)

Flip	On collision with Knight	Add action
Knight	Action = "stun"	Add action
Flip	Platform is falling	Knight Destroy
Flip	Y < Knight. Y-40	Flip Set Platform vector Y to -300
		System Add 100 to Score



42 Ammo Event

Next, add a new sub-event to the Flip Controls group. Use the **Flip** object and the **On collision with another object condition**. Select **Ammo** as the Object in the parameters.

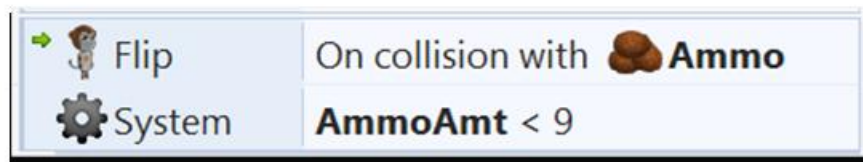
Add another condition to this event that uses the **System** object and **Compare variable** condition. Set the parameters to test if **AmmoAmt** is **Less than 9**.

This condition will only let Flip pick up ammo if he has less than 9.

Show AmmoAmt

Objective 2 – Adding Lives, Ammo, and Score

Ammo Event Image (Slide Layer)



43 System Add to AmmoAmt

Off of this event, add an action that will **Destroy** the **Ammo** object. Also add an action that uses the **System** object and the **Add to** action. In the parameters, set Variable to **AmmoAmt** and the Value to **3**.

Flip	On collision with Ammo	Ammo	Destroy
System	AmmoAmt < 9	System	Add 3 to AmmoAmt

Objective 2 – Adding Lives, Ammo, and Score

44 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

- Next, you'll need to add a sub-event that will limit the amount of ammo that Flip can have at one time.
- Make this new event for object **System** with the condition of **Compare variable**. Set the Variable to **AmmoAmt**, Comparison to **Greater or equal to**, and the Value to **9**.



Objective 2 – Adding Lives, Ammo, and Score

45 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

Next, Add an action for object **System** to **Set value** of **AmmoAmt** to 9.

Show New Sub-Event



Objective 2 – Adding Lives, Ammo, and Score



Sub-Event Image (Slide Layer)

On Your Own

Complete all the tasks before continuing to the next slide

19	Flip	On collision with Ammo	Ammo	Destroy
	System	AmmoAmt < 9	System	Add 3 to AmmoAmt
			Add action	
20	System	AmmoAmt ≥ 9	System	Set AmmoAmt to 9



46 System AmmoAmt Subtract From

With Flip now collecting Ammo, you'll want to set it up so when the gun is fired the AmmoAmt decreases. In Flip Controls group, find the event for when Space is pressed. On this event, add an action that will use the **System** object and the **Subtract from** action. Set the Variable to **AmmoAmt** and keep the value as 1.

Keyboard	On Space pressed	Flip	Set animation to "Shoot" (play from beginning)
		Flip	Spawn Coconut on layer 1 (image point 1)
		System	Subtract 1 from AmmoAmt

Objective 2 – Adding Lives, Ammo, and Score

47 AmmoAmmt Greater Than

You'll also want to add a condition to test if Flip has ammo to fire. On the same Space pressed event, add a condition that will use the **System** object to test if **AmmoAmt** is **Greater than 0**.

Keyboard	On Space pressed
System	AmmoAmt > 0

Objective 2 – Adding Lives, Ammo, and Score

48 On Your Own

On Your Own

Complete all the tasks before continuing to the next slide

◆ Add a sub-event to the Flip Controls group that tests if Flip collides with a Banana object. Add actions to this event that will destroy the Banana and add 5 to the Score variable.



Show Banana Event

Objective 2 – Adding Lives, Ammo, and Score




Banana Event Image (Slide Layer)

On Your Own

Complete all the tasks before continuing to the next slide

→ Flip	On collision with Banana	Banana	Destroy
		System	Add 5 to Score
		Add action	



49 Knight On Any Finished

Next, you'll set the event that will remove a life when a Knight collides with Flip. In the Knight Actions group, locate the event that tests if Action is equal to "swing." Off of this event, add a sub-event that uses the **Knight object** and the **On any finished** condition in the Animations section.

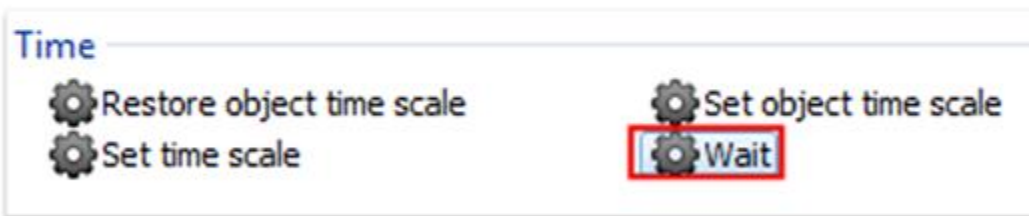


Objective 2 – Adding Lives, Ammo, and Score

50 System Wait

Add an action to this sub-event that uses the **System** object and the **Subtract from** action. Set the Variable to **Lives** and keep the Value as 1.







Next, add an action that uses the **System** object and an action called **Wait** which is under the Time section at the bottom.



Objective 2 – Adding Lives, Ammo, and Score

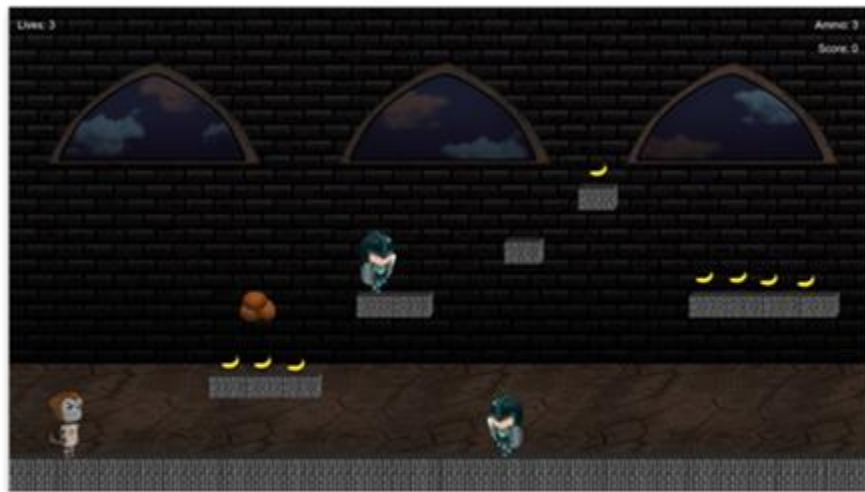
51 Restart Layout

Keep the Seconds parameter as 1.0 and click **Done**. Now add a final action that uses the **System** object and the **Restart layout** action.

 Knight	Action = "swing"	Add action
  Knight	On any animation finished	 System Subtract 1 from Lives
		 System Wait 1.0 seconds
		 System Restart layout

Objective 2 – Adding Lives, Ammo, and Score

52 Play the Game!



Play the Game!

53 Success

