

### Build a balloon pop game! Part 2

An MIT App Inventor tutorial

Feat. Tim the beaver





#### App overview: Build a balloon pop game! Part 2

In this second version of the Balloon Pop game, we will be adding functionality to respond to balloons reaching the bottom of the screen. We will be including:

- A "game over" message
- A restart button that appears after a player has lost the game

Because this builds on the first version of balloon pop, you must have gone through the "Build a balloon pop game! Part 1" tutorial first in order to complete this one.





#### Step 1: Add extra components!

To extend this app you will need to add three components—a HorizontalArrangement, Label, and a Button. Place the HorizontalArrangment *below* the Canvas, and the Label and Button *inside* the HorizontalArrangement.







#### Your screen should now look like this:







#### Step 2: Set properties

1. Click on Label1, set Text to "Game

over!" and uncheck the Visible box.

2. Click on Button1, set Text to

"Restart". Then uncheck the

Enabled and Visible boxes.

Properties	Properties
Label1	Button1
BackgroundColor	BackgroundColor Default
FontBold	Enabled
FontItalic	FontBold
Cant Size	FontItalic
24	FontSize
FontTypeface default •	FontTypeface default •
HTMLFormat	Height Automatic
HasMargins 🖉	Width Automatic
Height Automatic	Image None
Width Automatic	Shape default •
Text	
Game over!	Text Restart
TextColor Black	TextAlignment center : 1 * TextColor Default
Visible	Visible



**MIT App Inventor** 

Display hidden components in Viewer Check to see Preview on Tablet size.	<ul> <li>Screen1</li> <li>Canvas1</li> <li>ImageSprite1</li> <li>ImageSprite2</li> <li>ImageSprite3</li> <li>HorizontalArrangement1</li> <li>Label1</li> <li>Button1</li> </ul>
	<sup></sup> <sup>™</sup> Clock1
Non-visible components	Rename Delete Media Balloon.png Upload File



#### Step 3: Switch to the blocks window to write code!

Now that all new components have been added to the app, switch to the blocks window by clicking the "Blocks" button in the upper right corner.





When a balloon reaches the bottom of the screen, we would like all three of the balloons to re-set in the same way we had them reset when touched. Because we are doing the same thing, we can reuse code!

1. In part 1 of this tutorial, you wrote the following code to reset each balloon

if it touched the bottom of the screen:



2. Duplicate each of these groups of blocks by right-clicking on the "when" block and choosing "Duplicate"





1. Remove the green interior blocks from all three of the duplicated blocks, and stack all these green blocks

together. Then you can delete the duplicated "when" blocks. Make sure not to change the originals!





The huge green block you just put together will form the basis for what we want to happen when an Image Sprite reaches the bottom edge of the device.

- 1. Click on ImageSprite1 and drag out a block that looks like this:
- 2. Now grab the stack of blocks you made in the previous slide and click them into place within the "when" block you just dragged out.
- 3. Right click on the "when" block and duplicate it twice to get three copies of this group of assembled blocks.
- Click on ImageSprite1 in the ImageSprite1.EdgeReached gold block in one of the duplicates to get a dropdown menu and change ImageSprite1 to ImageSprite2.
- 5. Change ImageSprite1 to ImageSprite3 in the other copy.

**MIT App Inventor** 

You should end up with one "when" block each for ImageSprite1, ImageSprite2, and ImageSprite3







# The code you have just written should look like this:



whe	ImageSprite2EdgeReached
ec	je
do	set ImageSprite1 Y - to (10)
	set ImageSprite1 🗸 . Enabled 🔽 to 🔰 false 🗸
	set ImageSprite1 Visible - to false -
	set ImageSprite2 Y - to (10)
	set ImageSprite2 Enabled - to 🖡 false -
	set ImageSprite2 Visible - to false -
	set ImageSprite3 Y - to (10)
	set ImageSprite3 Enabled - to 🖡 false -
	set ImageSprite3 Visible - to 4 false -





Awesome! Now when a balloon reaches the bottom of the screen, it will restart at the top of the screen. We want to do a little more than that, though. When the bottom of the screen is reached, the balloons should stop falling, and a game over message and restart button should appear.

- 1. Click on Clock1 and drag out a block that looks like this:
- 2. Click on Label1 and drag out a block that looks like this
- 3. Click on Button1 and drag out a block that looks like this:

Enabled -

to

And this: set Button1

4. Snap them all together.

set Clock1 . TimerEnabled to set Label1 . Visible to set Button1 . Visible to set Button1 . Visible to set Button1 . Enabled to set Button1 .







Now, we will fill in the stack of code we just created.

- 1. Click on Logic and drag out a False block. Snap it into place beside the "set Clock1.TimerEnabled" block from the previous slide.
- 2. Click on Logic and drag out a True block. Duplicate the True block two times. Then snap them into place beside the bottom three blocks in the stack.

Your final result should look like this:





when ImageSprite1 .EdgeReached

set ImageSprite1 - . Y- to [10]

set ImageSprite2 - . Y - to 10

set ImageSprite1 - . Enabled - to I false -

set ImageSprite1 - . Visible - to | false -

edge

do

#### Step 3 continued

- Take the stack of code you just made and duplicate it 1. three times to get four three copies. Move one of them aside for now; you will use it later.
- Add one of the three remaining to each of the "when 2. ImageSprite.EdgeReached" blocks

set	Clock1 • . TimerEnabled • to false •
set	Label1 . Visible to true
set	Button1 . Enabled . to true
set	Button1 • . Visible • to true •

false 🔻

to (

true

true

true

to

to

#### set ImageSprite2 . Enabled to false set ImageSprite2 - . Visible - to ( false 🔻 set ImageSprite3 - . Y - to (10) set ImageSprite3 - Enabled - to | false set ImageSprite3 - . Visible - to false 2 3 when ImageSprite2 .EdgeReached edge 4. Put aside for later do set ImageSprite1 - . Y - to . 10 when ImageSprite3 .EdgeReached set ImageSprite1 - . Enabled - to I false edge set Clock1 v TimerEnabled • set ImageSprite1 - . Visible - to false do set ImageSprite1 - . Y - to (10) set ImageSprite2 - . Y - to (10) Visible 🔹 set ImageSprite1 - . Enabled - to 📜 false set Label1 • to false set ImageSprite2 - . Enabled - to set ImageSprite1 - . Visible - to false 🚽 set Button1 Enabled 🔹 set ImageSprite2 - Visible - to false set ImageSprite2 - . Y - to (10) set ImageSprite3 - . Y - to 10 set Button1 • Visible 🔹 set ImageSprite2 . Enabled to false -false 🔻 set ImageSprite3 - . Y - to (10) set ImageSprite3 . Visible to false set ImageSprite3 - Enabled - to 🔰 false -

set ImageSprite3 . Visible to

set ImageSprite2 - . Visible - to (

false -

false -





#### The code you have written so far should look like this:

whe	n (Im	ageSprite1EdgeReached
edge		
do	set	ImageSprite1 Y - to
	set	[ImageSprite1 -]. Enabled - to false -
	set	[ImageSprite1 - ]. Visible - to false -
	set	ImageSprite2 Y - to
	set	ImageSprite2 Enabled - to 🕴 false -
	set	ImageSprite2 Visible - to false -
	set	ImageSprite3 Y - to
	set	ImageSprite3 Enabled - to 🔰 false -
	set	ImageSprite3 - Visible - to false -
	set	Clock1 TimerEnabled - to false -
	set	Label1 Visible - to true -
	set	Button1 Visible - to true -
	set	Button1 (Enabled - to (true -

vhe	n (Irr	ageSprite2 .EdgeReached
edge		
lo	set	ImageSprite1 Y - to
	set	ImageSprite1 🚽 . Enabled 🚽 to 🌾 false 🚽
	set	ImageSprite1 Visible - to false -
	set	ImageSprite2 Y - to
	set	ImageSprite2 🚽 . Enabled 🚽 to 🔓 false 🚽
	set	ImageSprite2 Visible - to false -
	set	ImageSprite3 Y - to
	set	ImageSprite3 🚽 . Enabled 🚽 to 🔓 false 🚽
	set	ImageSprite3 - Visible - to false -
	set	Clock1 TimerEnabled - to false -
	set	Label1 Visible - to true -
	set	Button1 Visible - to true -
	set	Button1 Enabled - to true -

whe	n (Im	ageSprite3 .EdgeReached
edge		
do	set	ImageSprite1 Y - to
	set	ImageSprite1 Enabled - to false -
	set	ImageSprite1 Visible - to false -
	set	ImageSprite2 Y - to
	set	ImageSprite2 🗸 . Enabled 🚽 to 🔓 false 🚽
	set	ImageSprite3 Y - to
	set	ImageSprite3 - Enabled - to false -
	set	ImageSprite3 - Visible - to false -
	set	ImageSprite2 Visible - to false -
	set	Clock1 TimerEnabled - to (false -
	set	Label1 Visible - to true -
	set	Button1 Visible - to true -
	set	Button1 Enabled - to Ltrue -



Now we need to tell the app what to do when the restart button is clicked

1. Click on "Button1" and drag out a block that looks like this:



- 2. Take the stack of blocks that you set aside.
- 3. For this stack of blocks, set all Trues to False, and all Falses to True. Then snap this block into place inside the "when Button1.Click" block.

Your final result should look like this:





### Step 4: Testing and debugging!

Awesome! You're all done coding your app. Open the App Inventor Companion app and make sure everything is working properly. Remember, your app should:

- Reveal a new balloon at the top of the canvas every half a second
- Have that balloon drop to the floor
- Remove the balloon if the user clicks on it
- If a balloon reaches the bottom, Game over! and the Restart button appear
- Clicking the Restart button should start the balloons dropping again





## Here are some ideas to add to your Balloon Pop app!

- Keep score
- Speed up the balloons dropping as time goes by or as the score gets larger
- Make the balloons smaller as time passes to make it harder to click on them
- Add more balloons!
- What else can you think of?

