- Step 1 - Setting Up The Scene
- Step 2 - Creating the Ball Actor
- Step 3 - Set up the Scene Walls
- Step 4 - Adjust the Ball's Movement
- Step 5 - Add A Paddle
- Step 6 - Add the Game Over Screen
- Step 7 - Add a Replay Button
- Step 8 - Add The Break Actors
- Step 9 - Programming the Breaks
- Step 10 - Add Buttons to Control Paddle Movement

Step 1 - Setting Up The Scene

In this step, we create the main game scene where all of the action happens. The players will spend most of their time in this scene.

What are Assets?

The first step in making games is gathering the materials. Any sounds or images you would like to use in creating the game must first be uploaded to the Actimator environment so that they can be accessed when you need them. In the case of this game, all of the necessary images have been designed and added for you, they can be found in the `Assets` tab.

Change the Screen View

- Click the `gear` icon in the bottom left corner.
- Change the value of `Device Orientation` to from *landscape* to *portait*.

Create a Scene

- In the gallery, click on the `Scenes` tab.
- Press the `Add Scene` button.
- Enter a name for the scene, such as “Main”.
- Press the `Create` button.
- Click on the “Main” scene in the gallery to open it.
- Optional: On the top right part of the scene, adjust the `Viewport` to *Google Nexus 7*. It’s useful to make it as large as possible where you can still see the whole scene, without having to scroll.

Add background image

- In the gallery, click on the `Assets` tab. Choose `Images`.
- Drag the **Main Background** image onto the scene.
- Move the image so that it covers the whole scene.
- You can use `Property Editor` to set the image’s **Left** and **Top** properties to “0”.

Step 2 - Creating the Ball Actor

What is an Actor?

In Actimator, images do not perform any action. A good example of this is the background image we added before. To make a game character that can perform actions, we need to create an actor.

Create the Ball Actor

- From the gallery, choose the `Actors` tab.
- Click on the `Add Actor` button. It will open a window to add an actor.
- Set `Actor Name` to “Ball”
- Under `Choose a Default Image`, choose the Ball image.
- Press the `Create Actor` button.
- Drag the Ball actor to the scene. This creates an `instance` or `entity` of the Ball actor.
- Run the Scene. As you can see, nothing happens yet.

Make the Ball Move

- Click the Ball entity on the scene. The actor’s properties will open to the right.
- Under the `Physics` section change the **Linear Velocity** from “0” to “10”
- Run the scene again. As you see, the ball will now move straight up to the top of the scene.

Change the direction of the ball

Click on Ball entity. Go back to `Property Editor` and change the **Velocity Direction** to “90” and run the scene. Do the same thing for “180” and “270”. This gives you a sense of how to set the direction in which the ball moves. In the next step, we will add walls to keep the ball in the scene.

Step 3 - Set up the Scene Walls

To keep the ball on the screen, you need to put some walls around the scene.

Create Horizontal Walls

Go to `Actors` and add a new actor and name it “HWall” and use the *horizontal wall* image for it. Drag and drop the **Hwall** on to the scene and move it to the right position, centered at the top of the scene. Remember, you can also adjust the **Top** and **Left** properties to help posititionthe wall accurately.

Make the wall motionless

For the ball, set the **Linear Velocity** to “20” and **Velocity Direction** to “0”. Run the scene. As you see, the wall will move when it is hit by the ball. To keep the wall from moving, we need to make the wall motionless. To do so:

- Select the wall on the scene. Go to `Property Editor` under the `Physics` category and set the **Type** to “Static”. Run the scene again. This time, the wall does not move when it is hit by the ball.
- To make all the horizontal walls motionless, go to the **HWall** (on left side of screen, select 2nd icon called “change Image and/or Properties”) `Image or Property Editor`, and set the **Type** to “Static”. Click “Update Actor.” Drag and drop the second **HWall** to the scene. Run again.

Create Vertical Walls

Add another actor and name it “VWall”. Use the *vertical wall* image for it and set its **Type** to “Static”. Drop two of the **VWalls** on the scene and put then in the correct positions on the left and right edges of the scene.

In the next step, we are going to make the ball bounce off of the walls correctly.

Step 4 - Adjust the Ball’s Movement

In this step, we are going to make it so that the **Ball** actor bounces off of walls correctly.

Shape of the actor

If you look at the **Ball** on the scene, you will notice a blue square around it. This square is known as the actor's boundary, the area in which the actor interacts with other actors in the scene.

Test the effect of the actor's boundaries on movement

Set the actor's **Linear Velocity** to "10" and the **Velocity Direction** to "45". Press the `Play` button. The ball should hit the right wall, then the top wall and finally begin to move straight down. This is not how a ball would normally behave when hitting a wall at an angle. The ball is not moving correctly because the actor's boundaries are in the shape of a square.

Change the shape of your actor

Since our actor is meant to be a ball, it needs to have a round boundary. To fix this:

- Select the **Ball** actor.
- In the `Property Editor` change the **Shape** to "circle". Now the **Ball** actor will have a blue outline around it in the shape of a circle.

Change how the ball moves

Now that your ball has the correct shape, you can try to play the game. You can see that as the ball moves, it spins around and begins to slow down. This is happening partly because the ball has a force called friction acting on it. To prevent the ball from slowing down:

- Select the **Ball** actor.
- In the `Property Editor` under `Physics` change the **Friction** from "0.5" to "0". Run the scene again. The ball is no longer spinning as it moves but it is still slowing down. The ball is still slowing down because every time it hits a wall, it loses velocity.

Add bounce to the ball

Consider throwing a rock at the ground. When the rock hits the ground, it stops moving. The rock does not move back towards you once it hits the ground because it is not bouncy. The same is true for the ball. To get the ball to properly move on the scene, you need to give it bounce. To add bounce to the ball:

- Go to the **Ball** actor's `Property Editor`
- Under the `Physics` section, change the **Bounciness** value from "0.2" to "1". **Bounciness** has a minimum value of "0" and a maximum of "1".
- Change the **Linear Velocity** to "100". Run the scene. Your **Ball** actor should now bounce around the screen at the same speed until you press stop.

In the next step, we are going to add the paddle that the player controls.

Step 5 - Add A Paddle

In this step, we are going to add a paddle to the game that will bounce the ball upward.

Add the Paddle Actor

- From the gallery on the left side of the screen, choose the `Actors` tab.
- Click the `Add Actor` Button. This will open a new window where you can add your new actor.
- Set **Actor Name** to "Paddle"
- Under `Choose a Default Image`, choose the *Plate* image.
- Under `Default Properties` scroll to the `Physics` properties. Change the **Type** to *static*. This will prevent the paddle from moving when the ball hits it.
- Press the `Create Actor` button.
- Drag the **Paddle** actor onto the scene.

Understanding how the paddle moves

When adding motion to an object, you have to consider two things, what is supposed to happen and when is it supposed to happen? In this case, you want the paddle to move left or right when the left or right arrow keys on the keyboard are pressed. To figure out how to program this, begin by selecting the **Paddle** actor on the scene. Using the arrow keys on your keyboard, move the paddle to the right. As you do so, you will notice that its **Left** value in the `Property Editor` is increasing. When you move the paddle to the left, the **Left** value decreases. So, you know that to move the paddle to the left, you should decrease the **Left** value. When you want to move the paddle to the right, you should increase the **Left** value.

Adding controls to the paddle

To add controls to the paddle, we must program the paddle. To do so:

- Under the `Actors` tab on the left side of the screen, move your mouse over the **Paddle** actor.
- Click the `</>` (Program Actor's Behavior) Icon to the right of the actor's name or double click on the actor.
- Click the `Events` tab
- Under the `Graphics` section, click and drag `Draw [before render]` into the programming area (the large white area to the right of the Events tab).
- Drag `If` from the top of the screen into the `Draw` event in the programming area.
- Click on the `Commands` tab
- Under `Logic` find and drag `Key` to the condition box in the programming area.
- Click the `Key` type (green box) and press the "right arrow key", the word "RIGHT_ARROW" should appear.
- Add another `If` to the programming area (*BELOW* the first "If" block, not *INSIDE* the "If" block!). Add another `Key` to the new condition area and set it to the "left arrow key".
- Under `Assignment/Variable` in the `Commands` tab, find `Increase` and drag it to the white space below the `If` statement for the **RIGHT_ARROW**. Add another `Increase` in the `If` for the **LEFT_ARROW**.

- Under **LEFT_ARROW**, change the `Increase` to *decrease* since the value of the **Left** property decreases as the paddle moves to the left. Under **RIGHT_ARROW**, the increase menu will stay on increase since the value of the **Left** property increases as the arrow moves to the right.
- In the `Commands` tab, scroll down to the `Actor Entity` section. Find `This Entity` and drag one to both *increase*/*decrease* in your `If` areas. Change the name of the `This Entity` properties to *left* for each.
- Now, if the right arrow key is pressed, the **Left** property increases by 1 while the key remains pressed, moving the paddle to the right. If the left arrow key is pressed, the **Left** property decreases by 1 while the key remains pressed, moving the paddle to the left.

Testing the paddle

Click the `Scenes` button on the left side of the screen. Click the `Main` scene. Press play and press the left and right arrow keys. The Paddle actor should now move left and right. You will notice that your paddle is moving slowly. To fix this:

- Click on the `Actors` tab and then click the **Paddle** actor. Click the `</>` icon to return to the actor's program.
- Change the *increase* and *decrease* values from 1 to 20. Return back to your scene.
- Click on your ball actor. Change the `Linear Velocity` property from "100" to "50"
- Test your paddle again (select Scenes on left and click "Main" scene to return to game). It should now be possible control the paddle and bounce the ball with it.

In the next step, we are going to remove the bottom wall and add actions for my when you miss hitting the ball with the paddle.

Step 6 - Add the Game Over Screen

When you miss hitting the ball, the game should display a game over screen that lets you know you have lost. In this step, we are going to add a game over scene.

Remove the bottom wall

To remove the bottom wall, click the bottom **HWall** actor on your screen and press the `Delete` key on your keyboard. The bottom wall should disappear from the screen. Play the game again, now when you miss the ball, it falls off of the screen but nothing else happens.

Add the game over message

Once the ball goes below the paddle, the game should show the game over message that tells the player they've lost. To add a game over message:

- Create a new scene and name it "GameOver".
- Open the **GameOver** scene.
- Click on the `Assets` tab and select the image named "Background_Game_Over".
- Set the **Top** property of the image to "0".
- Set the **Left** property of the image to "0". The image should now be perfectly positioned on the screen.

When to show the game over scene

The game over scene should appear when the **Ball** actor falls below the **Paddle** actor. A good way to figure out exactly when this should happen is to return to Scenes, "Main" and drag the **Ball** actor so that it is sitting just below the **Paddle** actor. If you look at the **Ball** actor's properties, you will see that its top position is "_____". This means that you could program the **Ball** actor to show the game over scene when the **Top** property is "_____". To program this:

- Click on the `Actors` tab, then click on the **Ball** actor.
- Select the `</>` Icon.

- Go to the `Events` Tab.
- Select `Draw (Before Render)` and drag it into the programming area.
- Add an `If` to the `Draw` section of the programming area.
- Go to the `Commands` tab and under `Logic` and drag the `Comparison` to the `If` in the programming area. Change the = = to >.
- Under `Actor Entity` find `This Entity` and drag it to **Value 1** in the comparison. Set `This Entity` to *top*. Change **Value 2** to "____". This means that if the **Top** property is greater than "____", the ball is below the paddle.
- Under `Commands` find the `Scene` section and select the `Switch Scene`. Drag and drop the `Switch Scene` into the programming area in the `If` box..
- Set the `Switch Scene` to the *GameOver* scene. Now, if the value of the **Ball** actor's **Top** property is greater than "1140", the ball has fallen below the **Paddle** actor so the scene will change to the **GameOver** scene.
- Return to the main scene. Place the ball in the middle of the scene and play the game. If the **Ball** actor falls below the **Paddle** actor, the game should show the **GameOver** scene.
- Change **Ball** velocity to a slower value if it's too fast for you to test.

In the next step, you will add a reset button to the **GameOver** scene that lets you restart the game.

Step 7 - Add a Replay Button

Once a player has lost the game, they should be able to restart. In this step, we are going to add a replay button to the game over scene.

Add the Replay button actor

For an image to have any action, it must first be created as an actor. This is true for buttons as well. To add a replay button, we have to:

- Go to the `Actors` tab on the left side of the screen and click the `Add Actor` button.
- Name the new actor “Replay” and select the *Replay_btn* image for the actor.
- Press create.
- Go to the **GameOver** scene, drag and drop the **Replay** actor onto the scene where you would like it.

Add action to the Replay button

For the replay button to work properly, it has to switch back to the **Main** scene when the button is pressed. To program this:

- Click on the **Replay** button actor in the `Actors` tab. Click the `</>` button next to it’s name.
- In the `Events` tab, find the section labled `EntityMouseTouch`. Drag and drop `Mouse Down` into the programming area.
- In the `Commands` tab, under the `Scene` Section, find `Switch Scene` and drag it into the **Mouse Down** section of the programming area. Set the scene to `Switch Scene` to *Main*.
- Return to the **Main** scene and press play. Once you have lost the game, press the **Replay** button on the **GameOver** scene. It should restart the game.

In the next step, we are going to add the objects that will break, to the game.

Step 8 - Add The Break Actors

Now that you have controls added to the game, you need breaks to add a challenge to the game. In this step, you are going to add breaks to the game.

Create the Break Actor

Since the breaks we are going to create will perform actions, they must be created as an actor. To create the **Break** actor:

- Go to the `Actors` tab
- Click `Add Actor`
- Name the new actor “Break” and select the *Break 01* image as the default image.
- Change the `Type` property to *static*
- Change the `Shape` property to *circle*
- Click `Create` to create the new actor.

Add breaks to the scene

Now that you have created your new **Break** actor, you can add it to your “Main” scene. You will need 7 total **Break** actors on the scene. Drag and drop the **Break** actor to the scene 7 times so that you have 7 copies(entities) of the actor on the scene. To arrange the entities correctly:

- Change the **Top** property to “100” for all entities of the **Break** actor.
- Change the **Left** property of the first **Break** entity to “100”
- The **Break** entities should all be 75 pixels apart. Set the **Left** property of the second **Break** entity to “175”. The second **Break** entity is now 75 pixels away from the first **Break** entity.
- The **Left** property of the third **Break** entity should “250”.
- The **Left** property of the fourth **Break** entity should be “325”.
- The **Left** property of the fifth **Break** entity should be “400”.
- the **Left** property of the sixth **Break** entity should be “475”.
- The **Left** property of the seventh **Break** entity should be “550”.
- All entities of the **Break** actor should now be arranged neatly in a row and the space between them should all be the same.

Change the images of the breaks

Not all of the Break entities are supposed to have the same image. To change the image of a Break entity, look in the `Properties`on the right side of the screen. There is an `Image` property.

- Click on the first **Break** entity, change the `Image` property to a different image, such as *Break 06*.
- Click on the next **Break** entity and change its `Image` property.
- Continue changing the `Image` property of each **Break** entity until they each have a unique image.

Add more rows of breaks

To play the game, we want more than one row of break entity. To add more rows of breaks:

- Press and hold the `Control` key on your keyboard.
- Click the first **Break** entity in the row, this should make a copy of the **Break** entity. The copy is sitting right on top of the original. Drag the copy below the first **Break** entity.
- Place another copy of the **Break** entity below the first copy, starting a third row of **Break** entity.
- Place a third copy below the previous copy. You should now have 4 rows of **Break** entity.
- Continue copying the rest of the **Break** entity so that you have 4 complete rows.

Position the break entities

To position your new **Break** entity neatly, you will have the change the **Top** and **Left** properties for the new **Break** entity you placed on the screen:

- Change the **Top** property for all of the **Break** entity in the second row to “175”

- The first **Break** entity in the second row will have a **Left** value of “100”. The second entity will have a **Left** value of “175”. The **Left** value for all of the entity in the second row should be the same as the **Break** entity directly above it. Set the **Left** value for all of the entity in the row.
- The **Top** value for all of the **Break** entities in the third row should be “250”. The **Left** value for all of the entities in the row should be the same as the one above it.
- The **Top** value for all of the **Break** entities in the fourth row should be “325” and the **Left** property should be the same as the entities above them.
- Play the game. As you can see, when the ball hits the breaks, note what happens to the breaks.

In the next step, you will program the breaks to be removed when they are hit by the ball.

Step 9 - Programming the Breaks

Now that you have added the Breaks, we need to program them to disappear when they are hit.

- Go to the `Actors` tab and click the **Break** actor.
- Click the `</>` icon to open the programming area.
- Under the `Events` tab, find `Collision Start` and drag and drop it to the programming area.
- In the `Commands` tab, under `Actor Entity`, drag and drop `Destroy` to the `Start Contact` event.
- Drag and drop the `This Entity` command to the empty space inside the `Destroy` command you just added. Now, whenever the break collides with any other entity, the break that was hit will be destroyed.

In the next step, you will add buttons to control your paddle so the game can be played on touchscreen devices such as a phone or tablet.

Step 10 - Add Buttons to Control Paddle Movement

In this step, we are going to add buttons to the screen that will allow you to control the paddle's movement. Right now, you control the paddle using keys on your computer's keyboard. What if you wanted to play the game on another device, such as a phone? The buttons you are going to add will let you play the game on a mobile device such as a phone or tablet.

Create buttons

If you look under the `Assets` tab, you will see images for both the right button and left button. Since these buttons are going to have action (controlling the paddle), they need to be created as actors. To do this:

- Go to the `Actors` tab. Click the `Add Actor` button.
- Name your new actor "Left_btn" and click the *Left_btn* image as the default actor image.
- Since the button will not collide with any objects in the game, make sure uncheck **Has Physics**. Create your new actor.
- Create another new actor named "Right_btn", set the *Right_btn* image as the default image. Make sure to uncheck **Has Physics** for this button as well.
- Place the **Left_btn** and **Right_btn** actors onto the scene. The **Left_btn** actor should go in the bottom left corner, the **Right_btn** actor should be placed in the bottom right corner.

Program the buttons

To program the buttons to control the **Paddle** actor's movements:

- Click on the **Left_btn** actor in the `Actors` tab.

- Click the `</>` icon next to the **Left_btn** actor's name.
- Under the `Events` tab, find the event named `Mouse Down` under `Entity Mouse Touch`. Drag and drop `Mouse Down` to the programming area.
- Go to the `Commands` tab. Drag and drop a `Decrease` into the `Mouse Down` programming area. Unlike the when you programmed the paddle to move with the arrow keys, this time, you will not use `This Entity`.
- Find `Entity` and place it in the `Decrease` command.
- Return to your Main scene and click on your **Paddle** actor. Under `Properties`, there is the `Name` property. Copy this property name.
- Go back to the program for your **Left_btn** actor. Paste the **Paddle** actor name into the `Entity` name section.
- Change the decrease value to "40".
- Click on the **Right_btn** actor in the `Actors` tab.
- Click on the `</>` icon next to the **Right_btn** to open the programming area.
- Click on the `Events` tab, find the category named `Entity Mouse Touch` and drag `Mouse Down` to the programming area.
- Under the `Commands` tab, drag and drop `Decrease` into the `Mouse Down` programming area.
- Change `Decrease` to *increase** since the value of **Left** is going to increase as the paddle moves to the right.
- Drag and drop the `Entity` command into the `Increase` command. Paste the name of the **Paddle** that you copied earlier into the `Entity` name section.
- Change the increase value to "40".

Now that you have programmed your buttons, you should be able to control the **Paddle** actor with the buttons.

Your game is now complete. Publish and share it with your friends.