Phaser HTML5 Game Tutorial: Build A Pong Game 6: Scoring and Resetting

August 7, 2015 By Leave a Comment

In part 5[1] of this of this series, we added keyboard controls to the paddles and collision detection to return the ball. Now we'll look at adding the scoring and game reset features.

The original Pong game had a winning score of 11. Whichever player won, the game would then return to the demo mode.

Let's start by adding the score limit in the `gameProperties` object:

gameProperties

screenWidth

screenHeight

dashSize

paddleLeft_x

paddleRight_x

paddleVelocity

paddleSegmentsMax

paddleSegmentHeight

paddleSegmentAngle

ballVelocity

ballRandomStartingAngleLeft

ballRandomStartingAngleRight

ballStartDelay

scoreToWin

Enabling and disabling the game world boundaries

Our current game doesn't allow the ball to go beyond the game world boundaries. We will need to remove the left and right boundaries to allow the ball to pass through so either player can score.

To do that, let's add a new function called `enableBoundaries` after the `enablePaddles` function:

enablePaddlesfunctionenabled

paddleGroupsetAll'visible'enabled

paddleLeft_upenabledenabled

paddleLeft_downenabledenabled

paddleRight_upenabledenabled

paddleRight_downenabledenabled

enableBoundariesfunctionenabled

Similar to the `enablePaddles` function, we will pass in a boolean argument to determine whether the left and right boundaries will receive any collision response. Here is the code to add into the function:

enableBoundariesfunctionenabled

physicsarcadecheckCollisionenabled

physicsarcadecheckCollisionrightenabled

We will call this function in the `starDemo` and `startGame` functions:

startDemofunction

ballSpritevisiblefalse

resetBall

enablePaddlesfalse

enableBoundaries

inputonDownstartGame

startGamefunction

inputonDownremovestartGame

enablePaddles

enableBoundariesfalse

resetBall

If you were to test this out now, the ball would be able to go beyond the left and right game world boundaries… then disappear forever. Not exactly what we're trying to achieve. Once the ball leaves the game world, we need to do 3 things:

1. The game needs to reset the ball so it reappears in the middle of the screen.
2. The ball must then move towards the paddle that failed to return the ball.
3. The winning side must score a point.

Resetting the ball

To reset the ball, we need to create a new function that will be called whenever the ball leaves the game world. We'll add the `ballOutOfBounds` function after the `collideWithPaddle` function:

ballOutOfBoundsfunction

resetBall

So far we only have one line of code within the function that will call our the `resetBall` function. To test it out, add the following line of code in the `initPhysics` function:

initPhysicsfunction

physicsstartSystemPhaserPhysicsARCADE

physicsenableballSpritePhaserPhysicsARCADE

ballSpritecheckWorldBounds

ballSpritecollideWorldBounds

ballSpriteimmovable

ballSpritebounce

```
ballSpriteeventsonOutOfBoundsballOutOfBounds
```

This line will add an event listener that waits for the `onOutOfBounds` event to trigger then calls the `ballOutOfBounds` function. If you test it out now, it should work fine except that when the ball resets, it chooses a random direction instead of heading towards the paddle that missed.

To keep track of the side that missed, lets add the following code to the `mainState` object:

```
mainStatefunction

backgroundGraphics

ballSprite

paddleLeftSprite

paddleRightSprite

paddleGroup

paddleLeft_up

paddleLeft_down

paddleRight_up

paddleRight_down

missedSide
```

We will be adding either the “left” or “right” string values to it. This will be done in the `ballOutOfBounds` function:

```
ballOutOfBoundsfunction

ballSprite

missedSide'left'

ballSpritegamePropertiesscreenWidth

missedSide'right'

resetBall
```

If our ball goes beyond the left game world boundary (line 225), we set the `missedSide` property to “left”. If the ball goes beyond the right boundary (line 227), we set it to “right”.

One final change before we test this.

In our `startBall` function, we need to add on to our existing code:

```
startBallfunction

ballSpritevisible

randomAnglegamePropertiesballRandomStartingAngleRightconcatgamePropertiesballRandomStartingAngleLeft

missedSide'right'

randomAnglegamePropertiesballRandomStartingAngleRight

missedSide'left'

randomAnglegamePropertiesballRandomStartingAngleLeft

physicsarcadevelocityFromAnglerandomAnglegamePropertiesballVelocityballSpritevelocity
```

We set the `randomAngle` variable to choose a value from the `ballRandomStartingAngleRight` array (line 153) if `missedSide` is “right” or `ballRandomStartingAngleLeft` array (line 155) if `missedSide` is “left”.

Click here[2] to download the source codes up to this point. Here is our current work in progress:

Displaying player scores

We’ll start by adding a new object called `fontAssets` after the `soundAssets` object.

```
fontAssets
```

This is where we will be adding our properties related to all the text objects on screen. Add the following code:

```
fontAssets

scoreLeft_xgamePropertiesscreenWidth

scoreRight_xgamePropertiesscreenWidth

scoreTop_y
```

```
scoreFontStyle'80px Arial''#FFFFFF'align'center'
```

First we set the position of our score text fields on the screen. The left score text field will be about 160 pixels from the left side of the game world (line 45) while the right will be about 480 pixels (line 46). We also a small 10 pixel gap from the top (line 47) so the text fields are slightly below the top of our game world.

Next, we set the text field properties itself. A nested object called `scoreFontStyle` will be used to pass the properties to our text field when it is created. A nested object is basically an object within an object. You can easily identify it by its curly braces { }.

Our score text fields will be using the Arial font sized at 80 pixels with white text and aligned to the centre of the text field.

In the `mainState` object, add the following code:

```
mainStatefunction

backgroundGraphics

ballSprite

paddleLeftSprite

paddleRightSprite

paddleGroup

paddleLeft_up

paddleLeft_down

paddleRight_up

paddleRight_down

missedSide

scoreLeft

scoreRight

tf_scoreLeft

tf_scoreRight
```

To keep track of scores, we will use the `scoreLeft` and `scoreRight` properties. Both text fields will be added to the `tf_scoreLeft` and `tf_scoreRight` properties. I personally use tf_ as a prefix to easily identify the properties as a text field.

Now to add the text fields into the `initGraphics` function:

initGraphicsfunction

backgroundGraphicsgraphics

backgroundGraphicslineStyle0xFFFFFF

gamePropertiesscreenHeightgamePropertiesdashSize

backgroundGraphicsmoveToworldcenterX

backgroundGraphicslineToworldcenterXgamePropertiesdashSize

ballSpritespriteworldcenterXworldcenterYgraphicAssetsballName

ballSpriteanchor

paddleLeftSpritespritegamePropertiespaddleLeft_xworldcenterYgraphicAssetspaddleName

paddleLeftSpriteanchor

paddleRightSpritespritegamePropertiespaddleRight_xworldcenterYgraphicAssetspaddleName

paddleRightSpriteanchor

tf_scoreLeftfontAssetsscoreLeft_xfontAssetsscoreTop_yfontAssetsscoreFontStyle

tf_scoreLeftanchor

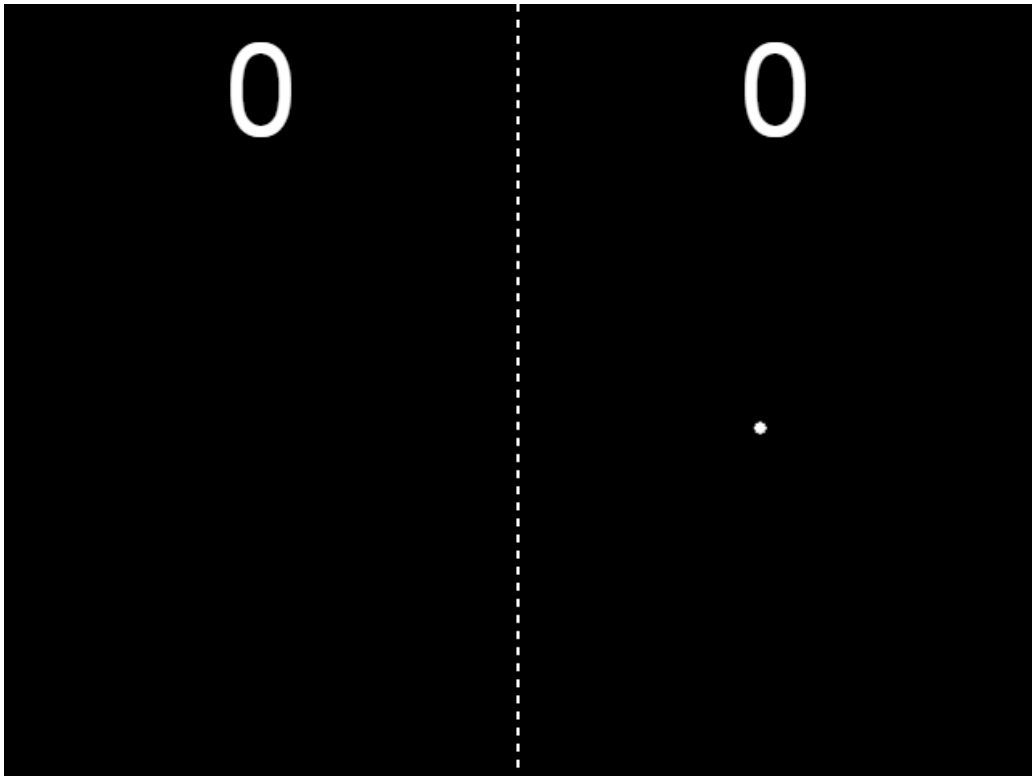
tf_scoreRightfontAssetsscoreRight_xfontAssetsscoreTop_yfontAssetsscoreFontStyle

tf_scoreRightanchor

Adding new text fields (line 114 and 117) requires 4 arguments:

- X position of the new text object.
- Y position of the new text object.
- The text that will be displayed.
- The style object containing style attributes such as the font, font size, colour, etc.

Also, the anchor point is changed to the horizontal centre and top of the text field (line 115 and 118). If you test it out now, you should see something similar to the screenshot here:



Keeping score

Now that we've got our score to display on the screen, we can start keeping track of it. After the `ballOutOfBounds` function, add two functions called `resetScores` and `updateScoreTextFields`:

ballOutOfBoundsfunction

ballSprite

missedSide'left'

ballSpritegamePropertiesscreenWidth

missedSide'right'

resetBall

resetScoresfunction

updateScoreTextFieldsfunction

The `resetScores` function will be used to set both player scores to 0 whenever a new game begins, while the `updateScoreTextFields` function is used to update our score text fields.

Add the following code:

resetScoresfunction

scoreLeft

scoreRight

updateScoreTextFields

Here we set our `scoreLeft` and `scoreRight` properties to 0. In the `updateScoreTextFields` function, we then assign these values to the `text` properties of our text fields:

updateScoreTextFieldsfunction

tf_scoreLeftscoreLeft

tf_scoreRightscoreRight

Next we need to update our `startGame` function to call `resetScores`:

startGamefunction

inputonDownremovestartGame

enablePaddles

enableBoundariesfalse

resetBall

resetScores

The `ballOutOfBounds` function needs to be updated to add points when the ball goes passed the left or right boundaries:

ballOutOfBoundsfunction

ballSprite

missedSide'left'

scoreRight

ballSpritegamePropertiesscreenWidth

missedSide'right'

scoreLeft

updateScoreTextFields

scoreLeft>=gamePropertiesscoreToWinscoreRight>=gamePropertiesscoreToWin

startDemo

resetBall

When the ball goes past the left boundary, the right player scores a point (line 255). Likewise when the ball goes beyond the right boundary, the left player scores a point (line 258). Once that happens, we update both score text fields (line 261).

Next we check to if either player has reached a total score of 11 to consider a winner.

At line 263, we have two conditions separated by a logical OR (||) operator. This means that if either conditions are true, the game will run the `startDemo` function. If neither player has reached the winner score, the game resets the ball and the next round begins.

Here's our current work in progress:

Click here[3] to download the source codes.

Our final step[4] will be adding sounds and making some minor tweaks so our game plays almost like the original Pong.

1. http://zekechan.net/getting-started-html5-game-development-pong5/
2. https://github.com/zekechan/phaser-html5-tutorial-pong/releases/download/1.0/6a-Resetting_the_ball.zip
3. https://github.com/zekechan/phaser-html5-tutorial-pong/releases/download/1.0/6b-Scoring.zip
4. http://zekechan.net/getting-started-html5-game-development-pong7/