Foundation Coding Assessment

The assessment is formed of three parts:

- Part I: Multiple choice

- Part II: Code modifications and debugging

- Part III: Coding exercises

Please ensure you follow the instructions carefully. To submit your work, complete Part I in this document and include the URL for your GitHub repository for Part II and Part III (more details below).

## Part I: Multiple choice

Please include a brief explanation for your choice if requested. Here is an example of the response format

|  |  |
| --- | --- |
| 0. What is your favourite colour?    a) red  b) green  c) blue  d) yellow | |
| *c* | *Because it’s the colour of the sky.* |

START OF ASSESSMENT

|  |  |
| --- | --- |
| 1. Which one of the following alphanumeric sequences allows you to specify a colour in JavaScript, and why?    a) #F3D3Z3  b) #F3D3E3  c) F3D3E3  d) #F3D3E34 | |
|  |  |

|  |  |
| --- | --- |
| 2. Which of the following is a valid statement, and why?  a) game.Stage.setBackgroundColour.("#F3D3A3");  b) game.stage.SetBackgroundColor("F3D3A3");  c) game.stage.setBackgroundColor("#F3D3A3");  d) game.stage.setBackgroundColor("#C3D3A3"); | |
|  |  |

|  |  |
| --- | --- |
| 3. Where would you expect text to be added if you saw the statement, and why?:  game.add.text(20, 20, "Welcome to my game");  a) Top-left corner  b) Top-right corner  c) Bottom-right corner  d) In the centre of the canvas | |
|  |  |

|  |  |
| --- | --- |
| 4. Which function would you use to make images available to the game, and why?   1. load() 2. create() 3. preload() 4. update() | |
|  |  |

|  |  |
| --- | --- |
| 5. Consider the following function:  function clickHandler(event) {  alert("The position is: " + event.x + ", " + event.y);  }  If you click in position x = 230, y = 250, the alert box will display:  a) The position is: 230, 170  b) The position is: +230+,+170  c) The position is: x = 230, y = 170  d) The position is 230, 170  Explain why. | |
|  |  |

|  |  |
| --- | --- |
| 6. Only one of the following statements is a valid instruction to preload sound in the preload() function, can you identify it? Give an explanation for your choice.   1. game.load.audio.("score", "../assets/point.ogg");   b) game.load..audio("score", "../assets/point.ogg");  c) game.load.audio(score", "../assets/point.ogg");  d) game.load.audio("score", "../assets/point.ogg"); | |
|  |  |

|  |  |
| --- | --- |
| 7. How many columns can you get in a bootstrap layout?   1. 7 2. 9 3. 12 4. 5 | |
|  |  |

|  |  |
| --- | --- |
| 8. In the code <img src="img/burger.jpg" class="img-rounded img-responsive"/> what is img-responsive used for?   1. To make a rounded image 2. To resize the picture uploaded 3. To adapt the picture to the different screen sizes 4. To enlarge the size of the picture | |
|  |  |

|  |  |
| --- | --- |
| 9. In the following RGBA colour code (10,50,127,60) the 127 is coding for:   1. Red 2. Blue 3. Black 4. Green | |
|  |  |

|  |  |
| --- | --- |
| 10. If you increase the second component of each pixel in the picture by 100, how does the colour of the picture change?   1. It becomes more green 2. It becomes more blue 3. It becomes more black 4. It becomes more pink | |
|  |  |

|  |  |
| --- | --- |
| 11. Which of the following commands will play a C?   1. play 20 2. play 30 3. play 60 4. play 22 | |
|  |  |

|  |  |
| --- | --- |
| 12. Which is the command to increase the volume of the sound of a note?   1. amp 2. release 3. sleep 4. increase | |
|  |  |

END OF PART I ASSESSMENT

|  |
| --- |
| For Part II and Part III, you will need to do some coding and edit the files we have provided you with in the folders (p2\_1, p2\_2, p2\_3, p3\_1, p3\_2, p3\_3) |

## Part II: Code modifications and debugging

1. In the p2\_1 folder, in the flappy.js file, you will find the following code. Modify the code so that it runs correctly.

// Adds a pipe part to the pipes array

function addPipeBlock(a, b) {

// make a new pipe block

var block = game.add.sprite(x, b, "pipe");

// insert it in the pipe array

pipes.push(blocks);

// enable physics engine for the block

game.physics.arcade.enable(block);

// set the block's horizontal velocity to a negative value

// (negative x value for velocity means movement will be towards left)

block.body.velocity.a = -200;

}

2. In the p2\_2 folder, in the flappy.js file, change the following code to obtain a three-square gap in each pipe:

// Generate moving pipe

function generatePipe() {

// Generate random integer between 1 and 5.

var gapStart = game.rnd.integerInRange(1, 5);

// Loop 8 times (8 is the height of the canvas).

for (var count = 0; count < 8; count++) {

// If the value of count is not equal to the gap start point

// or end point, add the pipe image.

if(count != gapStart && count != gapStart+1){

addPipeBlock(750, count \* 50);

}

}

// Increment the score each time a new pipe is generated.

changeScore();

}

3. In the p2\_3 folder, in the “syntax\_erros.txt” file, fix all the syntax errors.

## Part III: Coding exercises

1. Starting from the code in the p3\_1 folder, add the following features to the game:

1. Change the background when you get to 10 points;
2. Add some text on the top right corner of the game canvas and change its colour when you get 5 points;
3. If you haven’t done so already, make a sound play every time the score increments. Change the type of sound played when you get to 8 points.

2. Starting from the code in the p3\_2 folder, edit both the index.html and style.css files to achieve the following:

1. Change the page title into the name of your favourite movie
2. Change the colour of the heading h1 into red
3. Change the font of the heading h1
4. Add a link to a new html page displaying a picture of the movie
5. Change the list <ul>...</ul> into an ordered one
6. Change the font-size of the “veggie” items into 15px.

3. In the p3\_3 folder, build a game with the following features:

1. Every 3 seconds, one of a set of three objects, object A, object B and object C, is selected at random (but with each object having the same probability of selection) and enters the game canvas.
2. The objects enter the game canvas from the top, the right, and the left hand sides of the canvas.
3. Each of the objects has different scoring properties:
   1. If the player hits an object A, the score will increase by 5;
   2. If the player hits an object B, the score will decrease by 5.
   3. If the player hits an object C, the score will become the integer value closest to the square root of the current score.
4. When the score becomes negative, the game ends.
5. Move functions that enable the player to move in any direction in a grid environment, i.e. moveRight(), moveLeft(), moveUp() and moveDown().
6. When the player gets 12 points or more, the probability of object A entering the game canvas is halved, i.e. reduced to ⅙ and the probability of getting object B increases to ½.

|  |
| --- |
| To submit once your have completed everything, zip this entire folder again and send by e-mail to: [foundationcertificate@cambridgecoding.com](mailto:foundationcertificate@cambridgecoding.com) |