Creating a Mole Whack Game with CoronaSDK

This sample project uses the Mole Whack artwork provided Vicki Wenderlich.

1. Download the starter project from the APPlied Club projects page. Unzip the project.
2. Let's start by exploring the contents of our new project. Open the project folder and then open the `images` folder. Just take a look at some of the images in there to get an idea for what we have.
3. Open the Corona Simulator and open the project. Then open the `main.lua` file in Notepad++
4. Let's add one of the images to our simulator screen. Let's start with our "friend" the Mole. Add the following code in main.lua:

   ```
   local myMole=display.newImageRect("images/mole_1.png",178,200)
   ```

5. Before I explain what this code means, save and relaunch the project. You should see a brown speck in the top left of the simulator screen. This is our mole.
   He's not really in a good place, so let's move him to the center of the screen for now so we can get a good look at him.
6. After the line we just added, let's add the two following lines to center our mole.

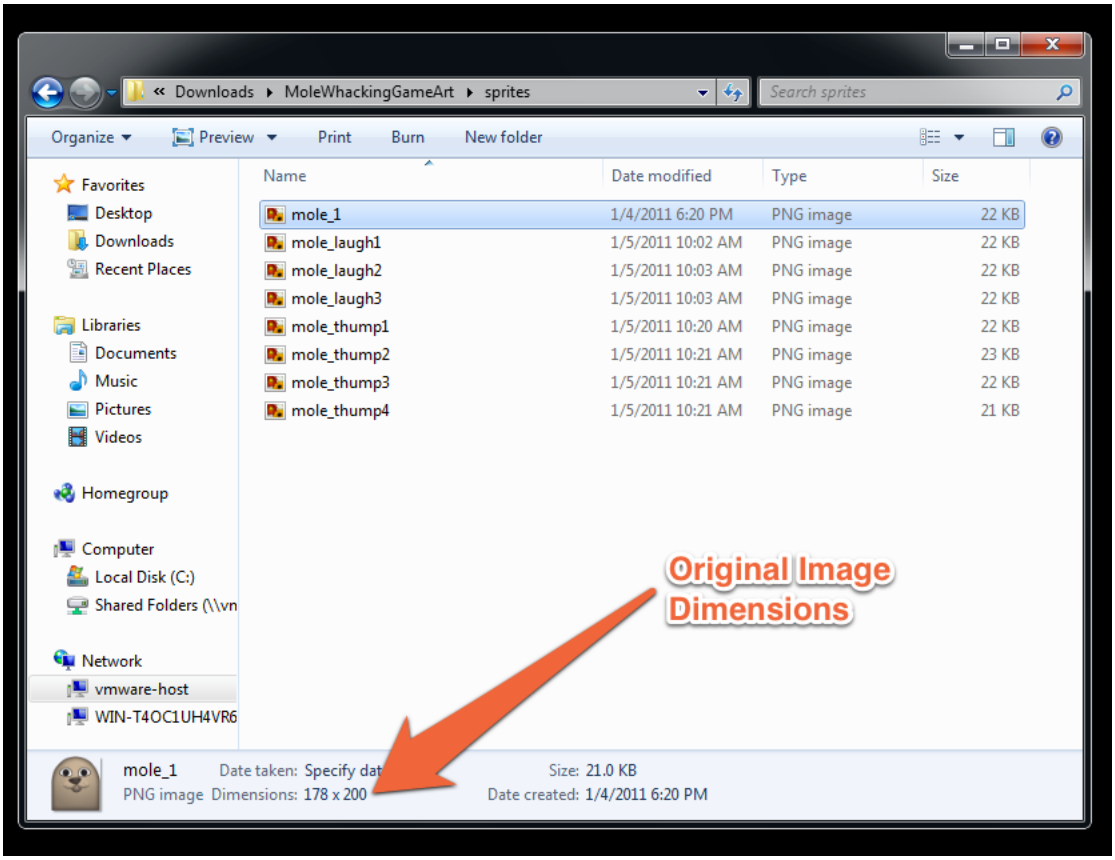
   ```
   myMole.x=display.contentCenterX
   myMole.y=display.contentCenterY
   ```

7. Save and relaunch the project and you should get a much better look at our Mole. Awww, isn't he cute?
8. Ok, let's go back and look at what that code actually means:

   ```
   local myMole=display.newImageRect("images/mole_1.png",178,200)
   ```

   The first line we typed, adds the mole graphic to the screen. we create a local variable named `myMole`, it could be named anything like `mikeTheMole` or `margaretTheMole` or `molesRule` or `jake` or `francis` or, okay, you get the idea. We set this variable equal to the result of the `display.newImageRect` command.

   The `display.newImageRect` command takes three arguments (or parameters). The first argument is the name of the image we want to load, in this case `images/mole_1.png`. The next two parameters are the original width and height of the image. Corona will use the original width and height to determine how it should look on other devices (like iPhones or iPads or any number of Android devices). To find the original size of your image on Windows 7, you can click the image file and in the bottom window, it should give you the dimensions.



9. Let's look at the next two lines of code:

```
myMole.x=display.contentCenterX
myMole.y=display.contentCenterY
```

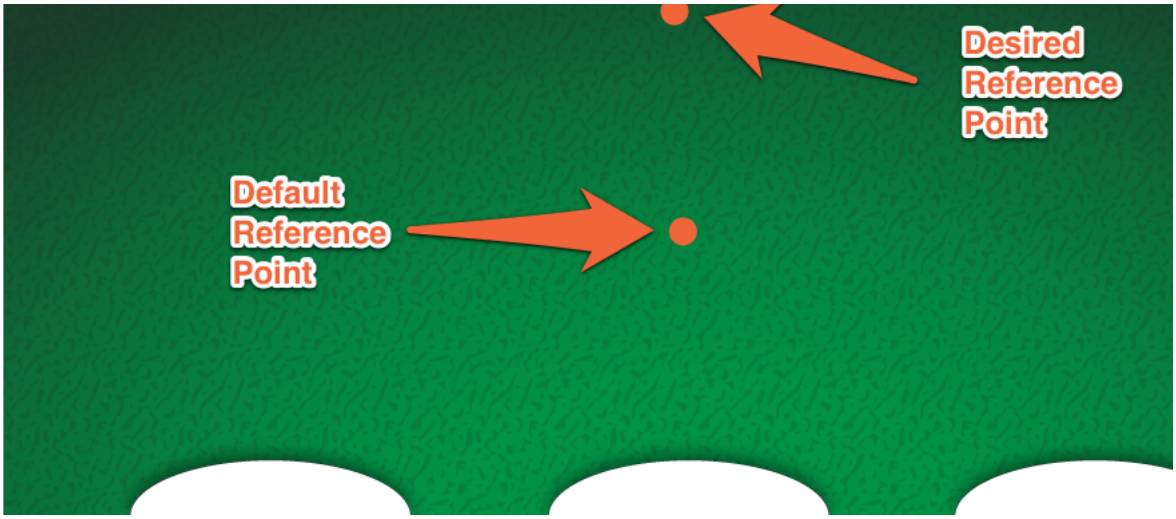
These two lines tell corona where to place our mole friend. The line referencing `myMole.x` tells Corona where to place the center of the mole object on the horizontal (or X) axis. The `myMole.y` tells Corona where to put the center of the mole object on the vertical (or Y) axis. The command `display.contentCenterX` is a special command that returns the numerical value for the center of the screen. So on an iPad device that is laid out sideways, the horizontal axis is divided into 1,024 points. The command `display.contentCenterX` returns the value 512. We could easily replace `myMole.x=display.contentCenterX` with this code: `myMole.x=512` and we would get the same effect. The benefit to using the `display.conentCenterX` command is that if we ever change devices (like to an iPhone), it will use the center of the new device. (In case you were wondering, an iPad turned sideways has 768 points so the center would be 384).

10. Ok our mole is pretty vulnerable out there with nothing to hide behind, so let's give him some cover...

11. Let's add the upper part of our grass:

```
local grassUpper=display.newImageRect("images/grass_upper.png",1024,384)
grassUpper.x=display.contentCenterX
```

12. Save and relaunch the project now. The code above gets the top of our grass centered horizontally but we need to place it vertically now. We don't want the upper grass to be centered vertically. Instead we want the top edge of the grass to be lined up with the top edge of the screen.
By default, Corona moves the center point of an object around but you can change this. This is called the object's referencePoint. Since I want to line up the top of the upper grass image, I would rather control where it is located by using the top of the image.



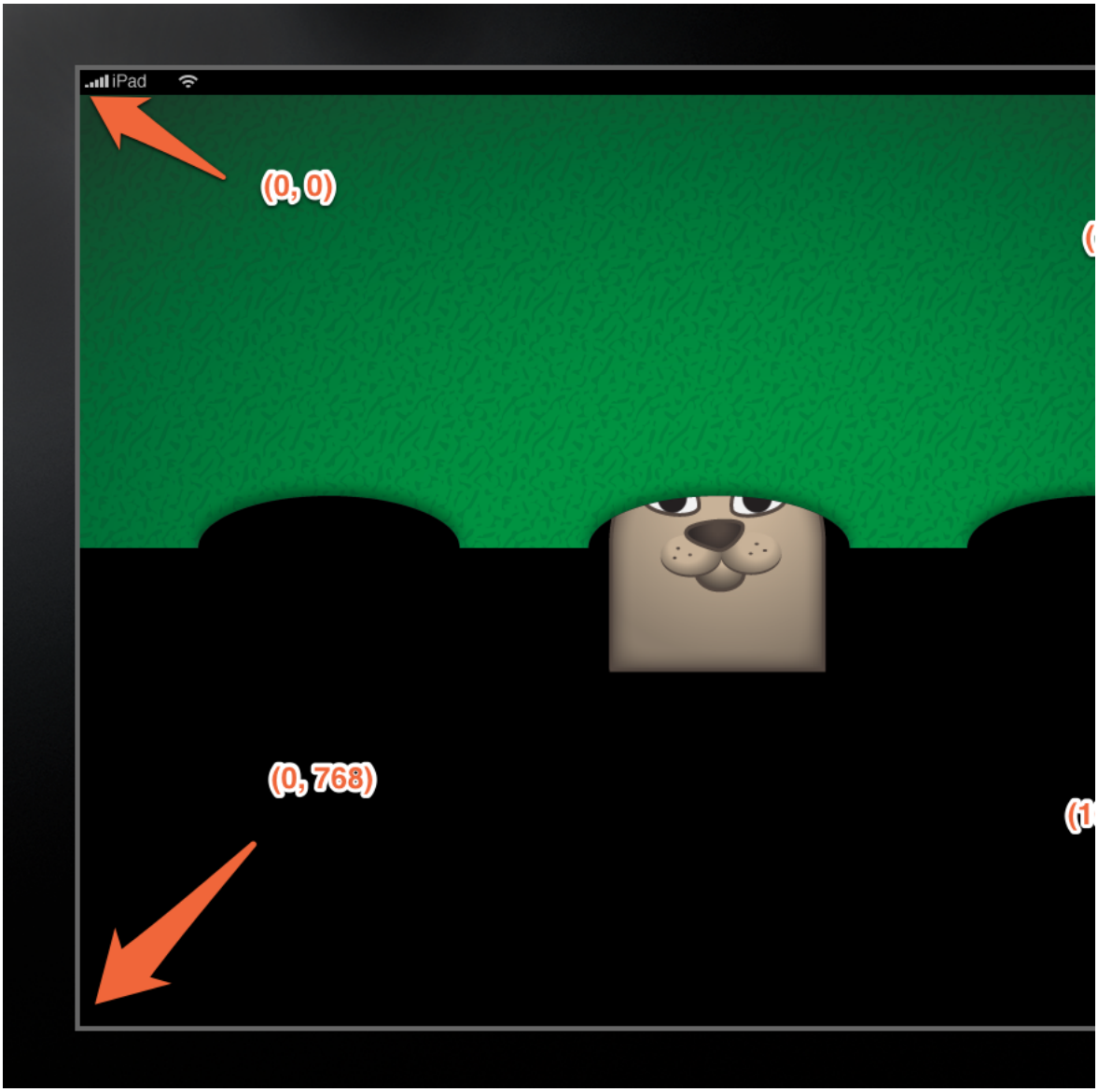
13. Here's how to change that reference point:

```
grassUpper:setReferencePoint(display.TopCenterReferencePoint)
grassUpper.y=0
```

To change the reference point, you perform the command `:setReferencePoint` on the object you want to change (`grassUpper` in this case), and you tell it what the new reference point you want to use, in this case `display.TopCenterReferencePoint`. The `display.TopCenterReferencePoint` is a special value that tells it to use the top of the object and it's center (horizontally) to move it. All of the available reference points can be found here: CoronaSDK setReferencePoint Docs **Be careful, the reference point is tied to the object, it has nothing to do with the screen or display.**

The next line `grassUpper.y=0` tells Corona to place the reference point at the value 0. In Corona, the top point of the screen is 0 and the bottom point of the screen is the screen height, so with an iPad on its side, the bottom point would be equal to 768.

14. Save and relaunch the project now. You should have something similar to the below image (without all the arrows and numbers).

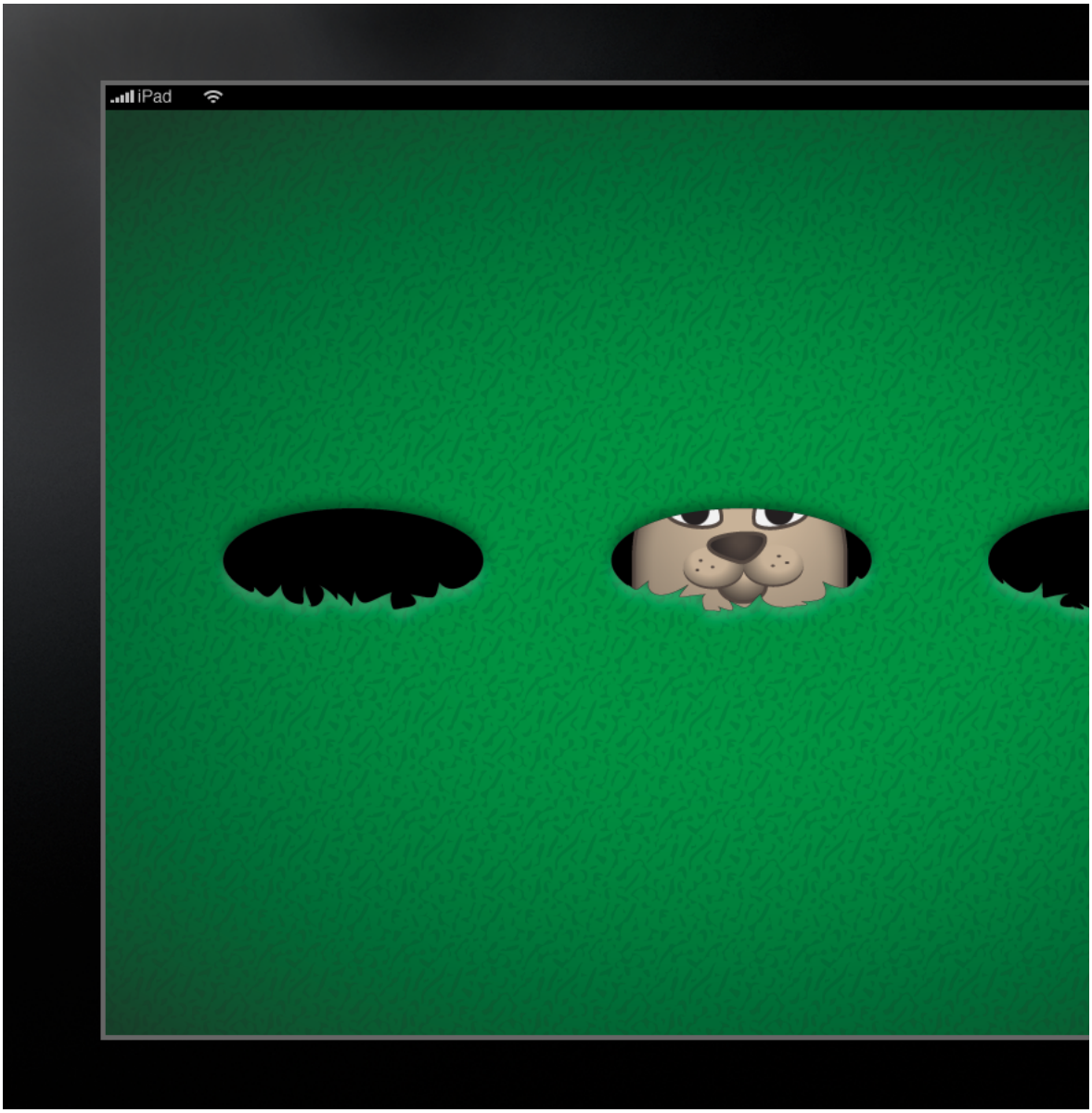


15. See if you can add the bottom grass on your own before going to the next step. The file is called `grass_lower.png`. You'll want to set the reference point to the bottom center of the object, or `display.BottomCenterReferencePoint`

16. Here is the code to add the bottom grass:

```
local grassLower=display.newImageRect("images/grass_lower.png",1024,384)
grassLower.x=display.contentCenterX
grassLower:setReferencePoint(display.BottomCenterReferencePoint)
grassLower.y=display.contentHeight
```

17. If you've noticed, our mole is now completely hidden behind our grass.



18. We can think of each of these objects as layers and as we create new objects, they are placed on top of older objects. This is the current layout of our screen:

    - Layer 1: Mole
    - Layer 2: Upper Grass
    - Layer 3: Lower Grass

19. It would look a little better if our layers were laid out like this:

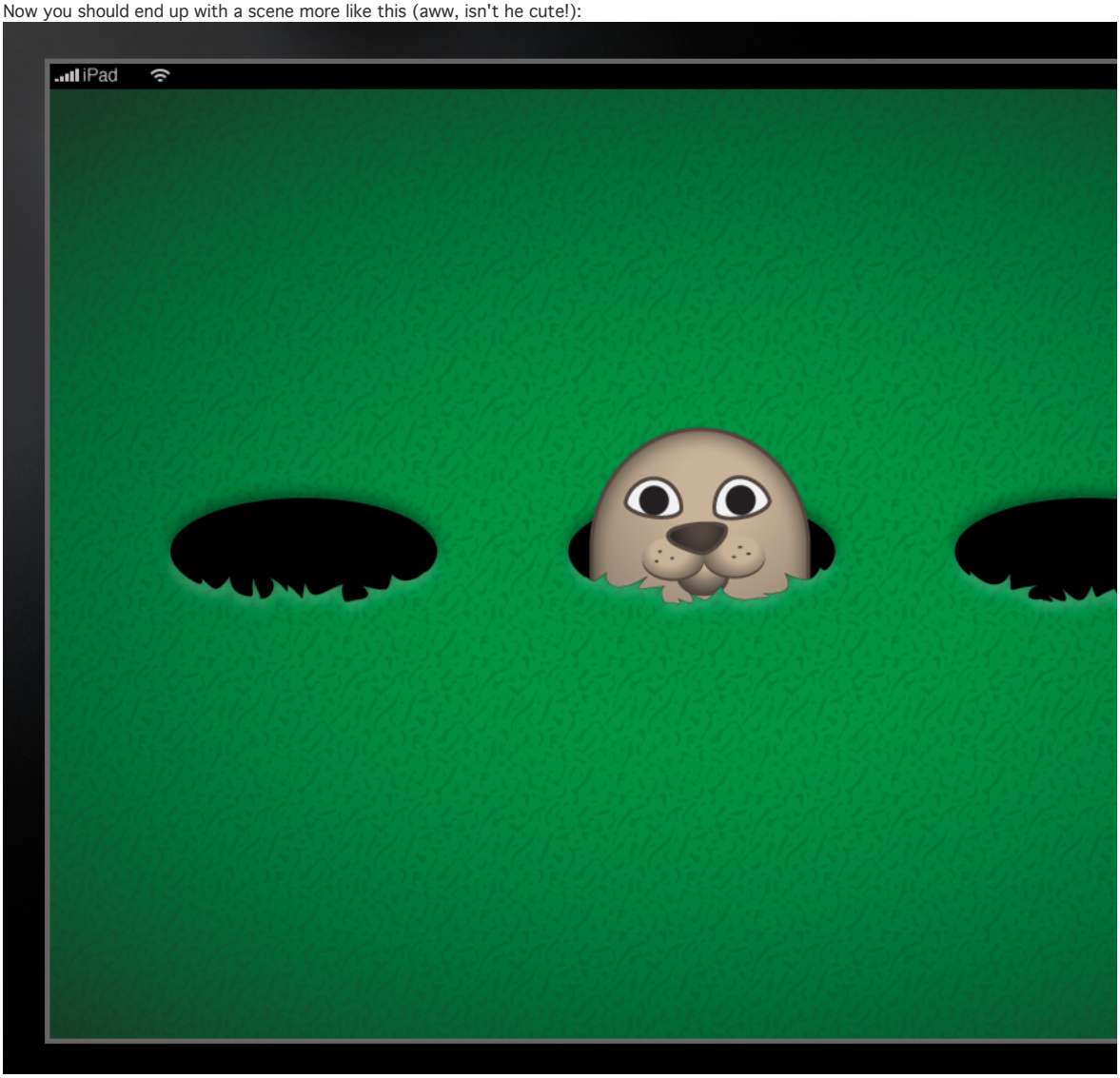
    - Layer 1: Upper Grass
    - Layer 2: Mole
    - Layer 3: Lower Grass

20. We can rearrange our objects but our options are a little limited. Basically, you can move an object all the way to the front, or you can send it all the way to the back. To send something to the back, we use the command `:toBack()` on the object. I bet you can't guess what we use to send it to the front? If you guessed, `:toFront()` you win!

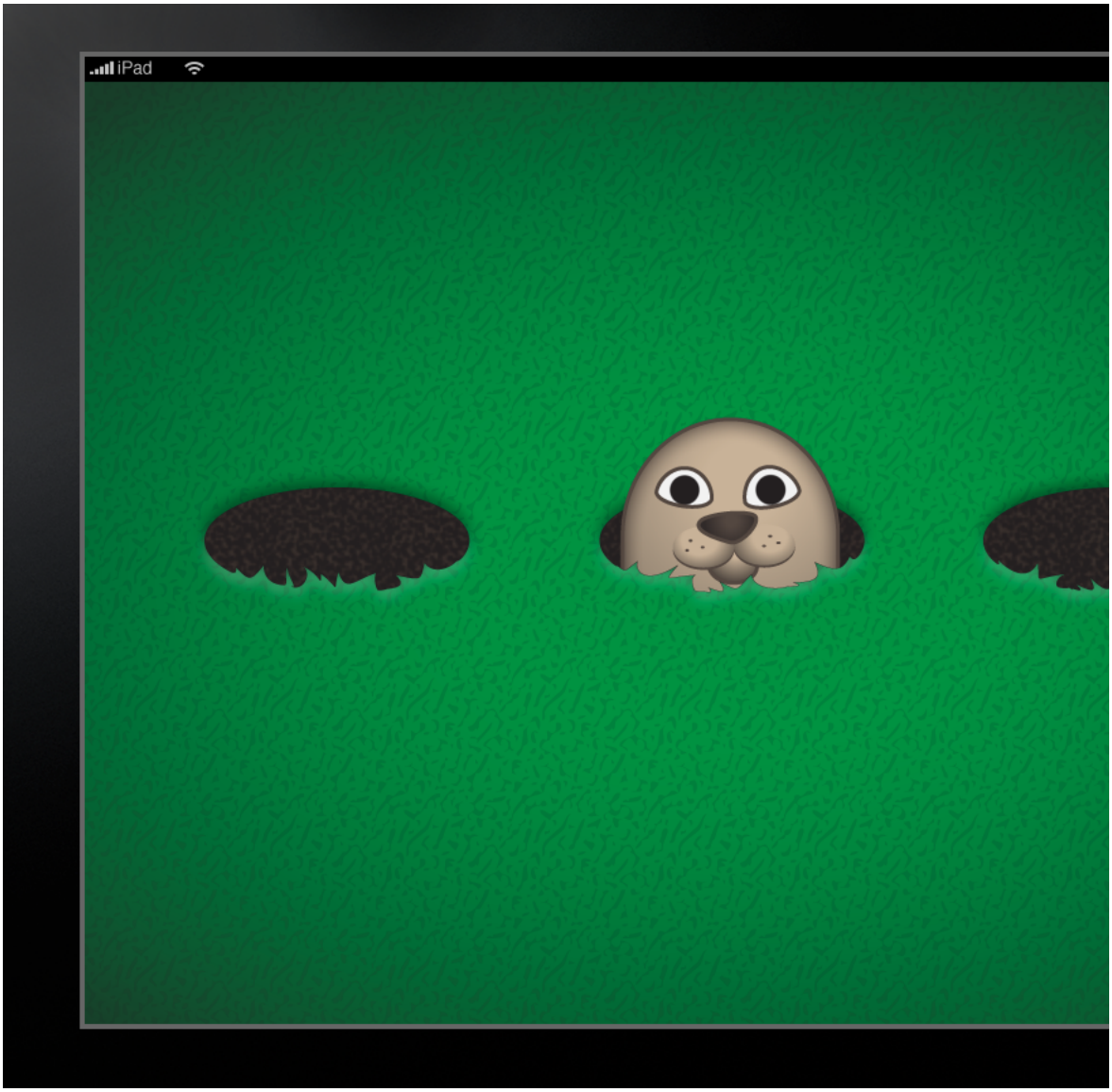
21. Let's rearrange our layers by adding the following lines of code:

```
grassUpper:toBack()
myMole:toFront()
grassLower:toFront()
```

22. Now you should end up with a scene more like this (aww, isn't he cute!):



23. Vicki gave us one more piece of artwork to add in. She gave us a file called `bg_dirt.png` that should go all the way in the back. Do you think you can add that yourself? Give it a shot. When you are done, your scene should look like this one (it looks similar to the step above but if you look closely, the holes have some texture to them now):



24. If you had any problems, here is the final code for the `main.lua` file:

```
-----------------------------------------------------------------------------------------
--
-- main.lua
--
-----------------------------------------------------------------------------------------

-- Your code here


    local myMole=display.newImageRect("images/mole_1.png",178,200)
    myMole.x=display.contentCenterX
    myMole.y=display.contentCenterY

    local grassUpper=display.newImageRect("images/grass_upper.png",1024,384)
    grassUpper.x=display.contentCenterX
    grassUpper:setReferencePoint(display.TopCenterReferencePoint)
```

```
grassUpper.y=0

local grassLower=display.newImageRect("images/grass_lower.png",1024,384)
grassLower.x=display.contentCenterX
grassLower:setReferencePoint(display.BottomCenterReferencePoint)
grassLower.y=display.contentHeight

grassUpper:toBack()
myMole:toFront()
grassLower:toFront()

local bgDirt=display.newImageRect("images/bg_dirt.png",1024,768)
bgDirt.x=display.contentCenterX
bgDirt.y=display.contentCenterY
bgDirt:toBack()
```