

## App 6

### Rock, Paper, Scissors App

#### Overview

The **Rock, Paper, Scissors App** will allow the user to play the Rock, Paper, Scissors game. This App allows for only 1 player as he will be playing against the iPhone. The game will be played with the following images:



lion0.png    cobra.png    rabbit.png

Note that the rabbit image is facing right while the lion and cobra are facing to the left. Four total images are provided for the lion that we will animate when the lion image is dragged (complete description of this property is described in later section)



The game is played by the user selecting one of the animals above. The iPhone will then randomly select one of the animals as well. The rules for our game are as follows:

1. Lion defeats rabbit (lion can catch rabbit)
2. Cobra defeats lion (cobra can bite lion)
3. Rabbit defeats cobra (rabbit can outrun cobra)

Our game will extensively utilize animation blocks and timer blocks. **The main objective of our game is to utilize the animation requirements as outlined in this document and the game will be graded primarily on these animation properties.**

---

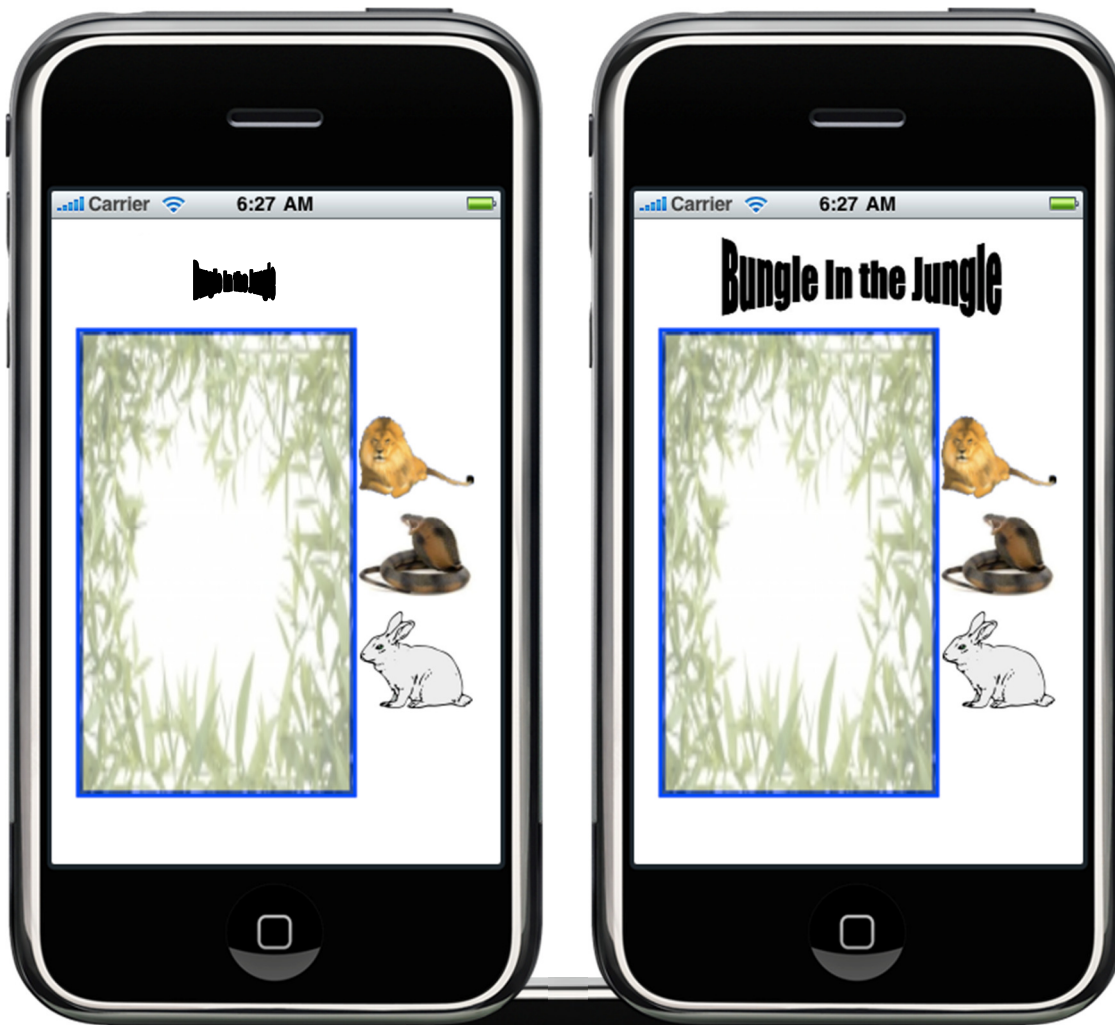
#### Opening Display Screen

The following image contains the title that we want to display when this App starts:

# Bungle In the Jungle

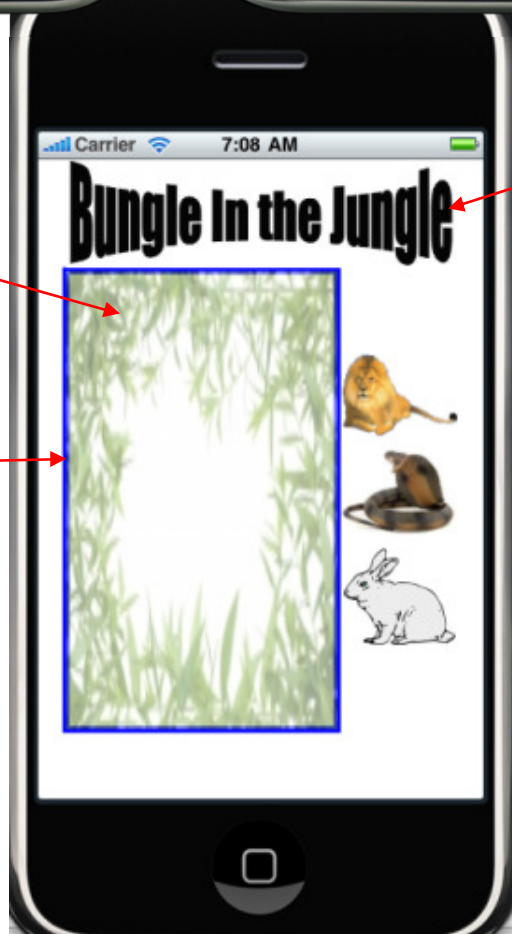
The image should start out very small and then expand to cover the upper portion of the app. This effect should be implemented as a zoom.

An example of the opening display for this **App** illustrating the zoom effect is shown below:



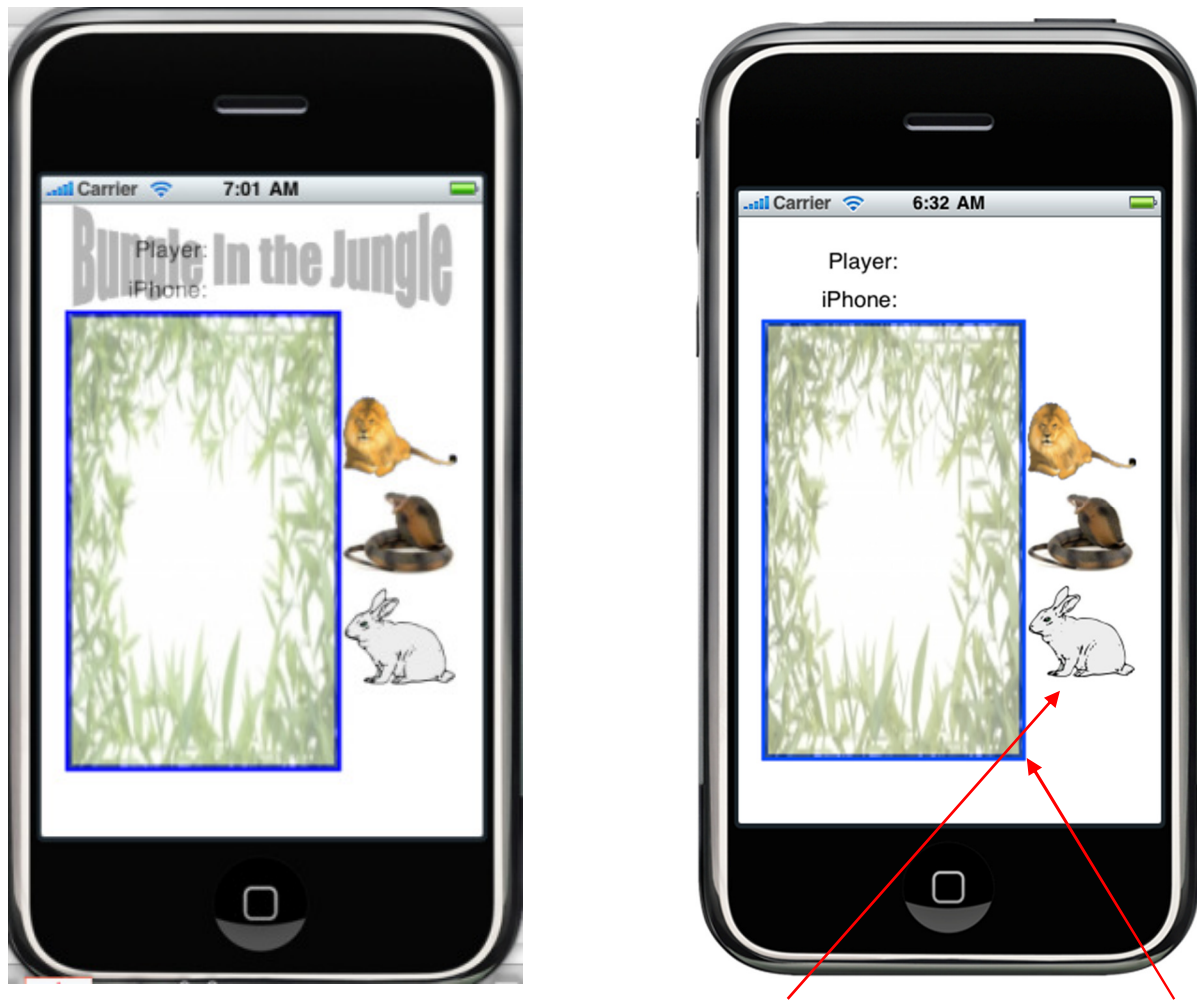
jungleframe.jpg

Draw the blue rectangle in  
your *drawRect* method



Title has finished zoom  
effect after a few seconds

The title should then fade out slowly and display the scoring labels as shown below:



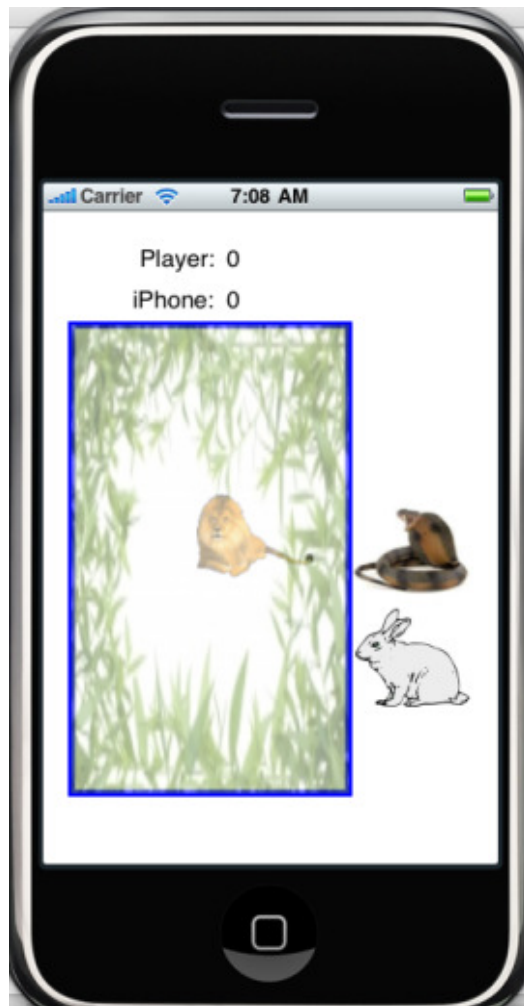
The user can play the game by dragging one of the animal images into the blue rectangular region. The following requirements are applied when dragging the animals.

1. The animal must be completely dragged into the blue region in order to play.
2. The animal is moved *gradually* back to its original location if the animal is dragged anywhere else other than the blue rectangular region.
3. Play *lionRoar.m4a* if the lion is touched (single tap only).
4. Continuously display the 4 lion images when the lion image is dragged.

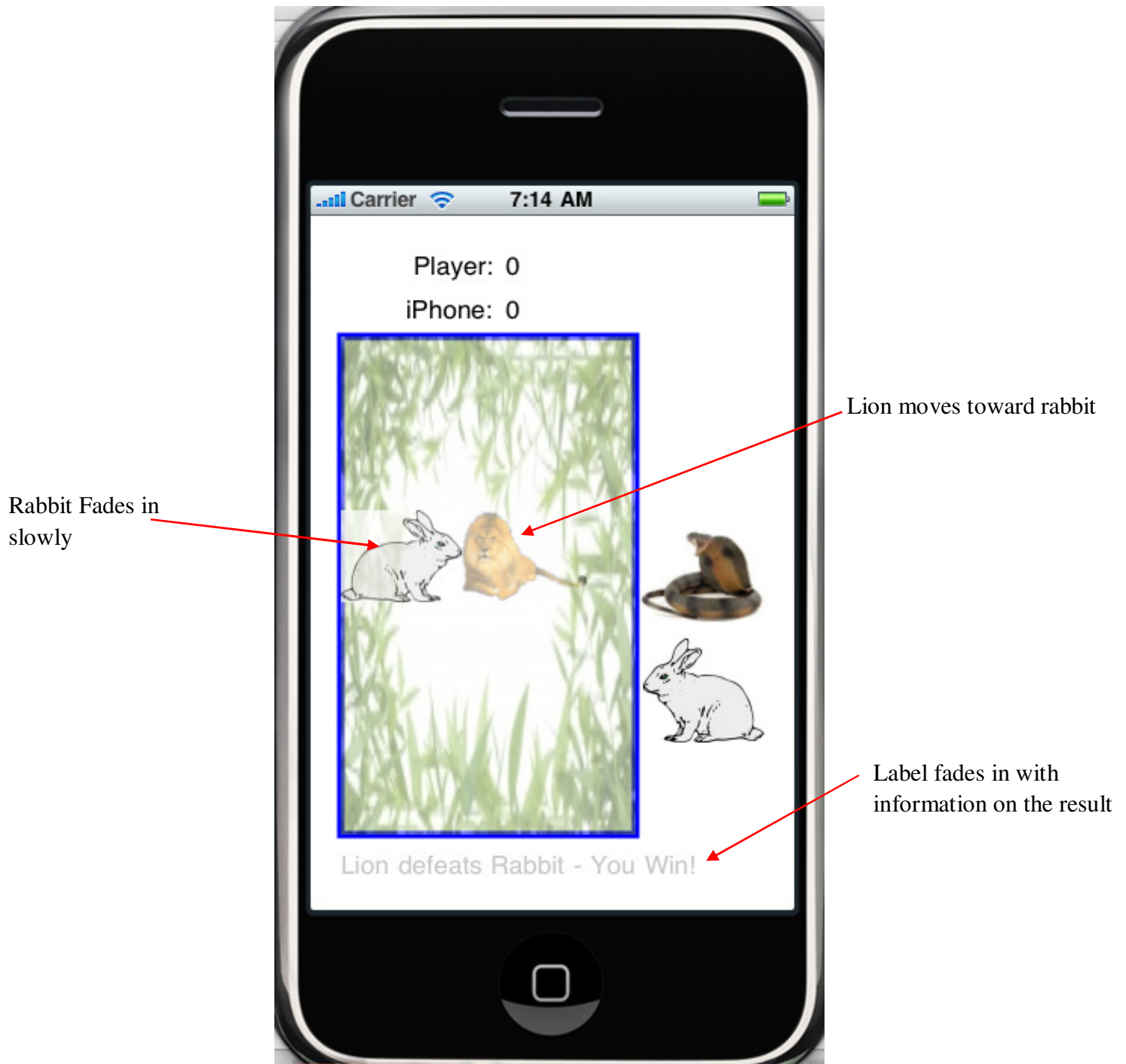
Example screen captures of dragging the lion outside of the blue rectangular region are shown below. Note how the lion is animated continuously through all 4 images as long as it is being dragged. Select a time interval that displays the lion images in a nice, continuous manner. The lion is gradually moved back to its original location if it is not completely placed within the blue rectangle. Be sure and display the original image by gradually moving it back to its original location when it is not dragged into the blue rectangle.



Suppose the lion is dragged completely into the blue rectangular region. An example of this is shown below:

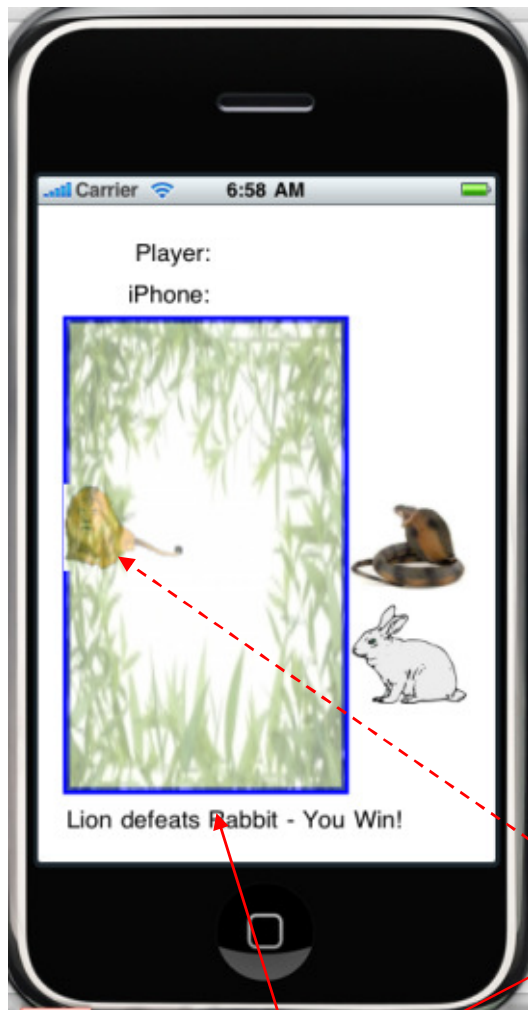


You will need to place the image at a fixed location once the dragging is complete - **Do not simply allow the image to reside wherever the player drags it!** As soon as the image is in place, the iPhone will randomly select another animal. An example of the iPhone selecting the rabbit is shown below (**Important:** *The animals should face each other*)



The rabbit fades into the blue rectangular region at a fixed location on the left side of the region. The lion defeats the rabbit, so it moves towards the rabbit and gradually covers the rabbit completely (see screen captures below). The Rabbit eventually fades out and the lion is the only object that appears.





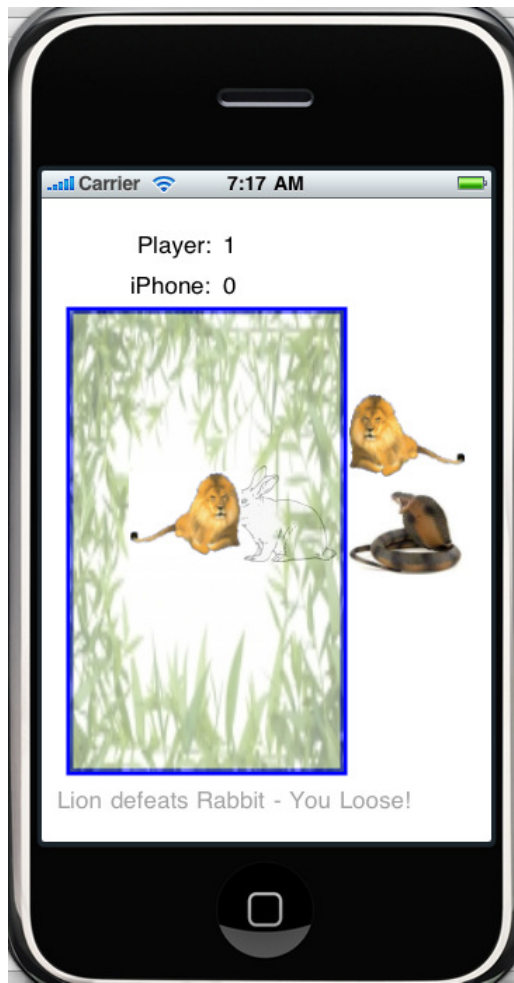
The lion then gradually moves back to the original location. The score is also updated indicating the player has won. The label indicating the result also fades out as the lion gradually moves back to its original location. A screen capture indicating this is shown to the right.

A tie results when the iPhone selects the same animal as the player. No points are awarded. An example of a tie for the lion or cobra is shown below. Both animals fade out, and the animal on the right will gradually fade back in while gradually moving back to its original location.

Note: The message below is **Tie!** for both cases. It does not appear in the case on the right because the screen capture was made during the fade out effect and the **!** did not appear in this screen capture.



The player loses if the iPhone chooses an animal that defeats the one that he selected. An example of a losing play is shown below:



The player chose the rabbit and the iPhone chose the lion. The lion should gradually move from the left and completely cover the rabbit while the rabbit fades out. The score is updated for the iPhone, the label indicating the result fades out, and the rabbit on the right fades back in while it moves gradually back to its original location.

---

## What you Submit

You should name your Xcode project **App6**. Zip up the entire Xcode project. Submit the zip file via Blackboard.

---

## Grading

It's important to provide the core animation functionality of the App as specified in this document. **You will not be graded on how identical your App's animation/graphics is to the example.** Feel free to use different backgrounds, different options, and different layouts too if you like. *Be sure and provide the core animation requirements as described in this document.* **Be sure and test your App on another machine to assure that it runs properly.**