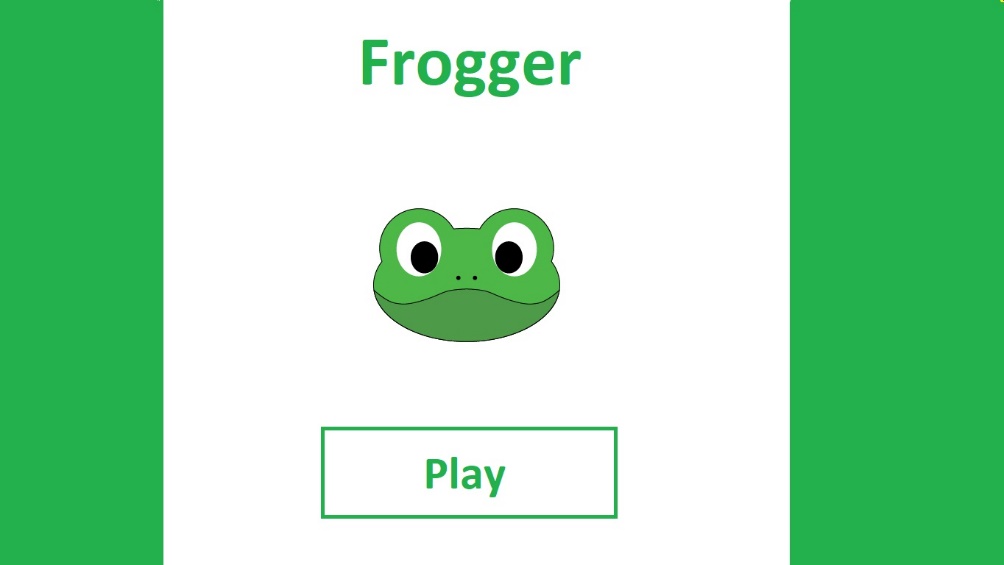
FRoggy!  
Project Scope

4/21/2021

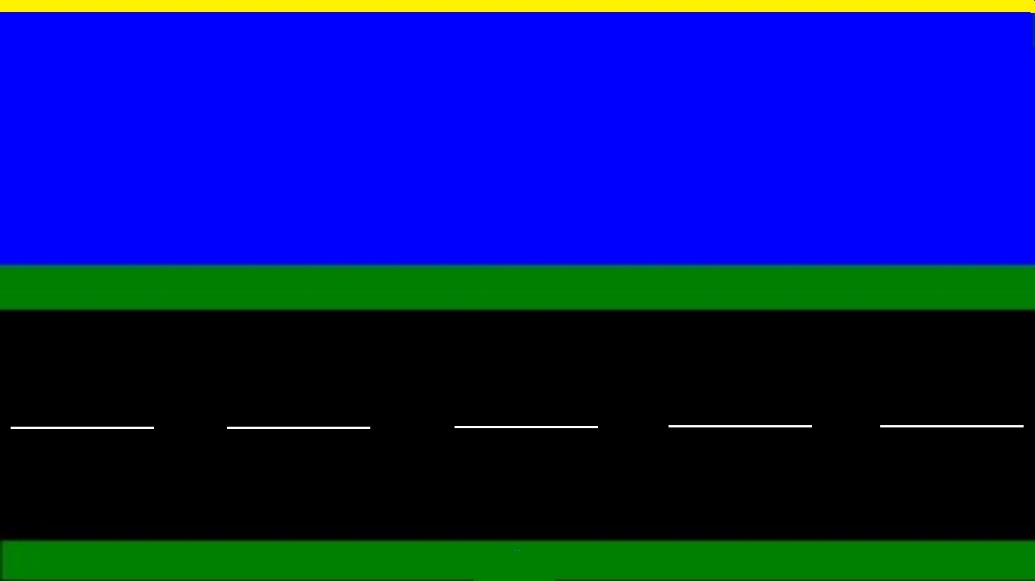
## High-Level Requirements

|  |  |
| --- | --- |
|  | The following use cases will be supported |

* When the application launches, the following initial page will be displayed:



* When the play button is pressed, the game screen will display:



* When the game begins, Frogger will start at the bottom of the screen facing north. Frogger should not be animating in any way and should display a single image of the frog facing north.
* Frogger will move with either the arrow keys or ‘W’, ‘A’, ‘S’, or ‘D’.
* When a movement key is pressed, Frogger will animate.
* Frogger must stay within the boundaries of the sketch.
* If Frogger collides with a car, truck or log, the skull image should be shown in its place and the sketch should freeze all animations
* If Frogger crosses the finish line, the sketch should freeze all animations and the words “You Win!” should display somewhere in the sketch.
* All cars, trucks and logs wrap [correctly] around.
* 4 lanes total. Two for the cars and trucks two for logs. Each lane goes in the opposite direction.
* You will make, 3 trucks, 5 cars for the street scene. You decide which goes where and in what direction.
* You will make, 5 logs or more if you scale them down.

## Technical Requirements

|  |  |
| --- | --- |
|  | The following functional requirements will be implements |

* All animation files will be stored in corresponding arrays. In other words, each direction the frog can move will have it’s own global array of image file names that it iterates through.
* noLoop() will be used to stop the sketch,
* The following functions will be created:
  + void setup( )
    - Initialize the size of the sketch.
  + void draw( )
    - Calls helper functions in sequence
      * move( )
      * animateCars( )
      * animateLogs( )
      * animateTrucks( )
      * boolean win = didWin( )
      * boolean lose = didLose( )
  + void move( )
    - Checks for movement keys
    - Moves Frogger a direction
    - Animates through the images
  + void animateCars( )
    - Create, move and wrap cars
  + void animateLogs( )
    - Create, move and wrap logs
  + void animateTrucks( )
    - Create, move and wrap trucks
  + boolean didWin( )
    - Check if you crossed the finish line
  + boolean didLose( )
    - Calls checkCollision once for each object in the game.
  + boolean checkCollision(float r1x, float r1y, float r1w, float r1h, float r2x, float r2y, float r2w, float r2h )

boolean checkCollision(float r1x, float r1y, float r1w, float r1h, float r2x, float r2y, float r2w, float r2h) {

// are the sides of one rectangle touching the other?

if (r1x + r1w >= r2x && // r1 right edge past r2 left

r1x <= r2x + r2w && // r1 left edge past r2 right

r1y + r1h >= r2y && // r1 top edge past r2 bottom

r1y <= r2y + r2h) { // r1 bottom edge past r2 top

return true;

}

return false;

}