Code Documentation: Android Development Team

Jillian Andersen, Jordan Apele, David Ruhle, Kyle Wenholz

December 14, 2011

Due to the nature of web-development, not all of our code is “documentation worth”. There are, of course, comments throughout all of the code, but the pieces developed by our team that are worthy of a full document are below.

After collaborating with the iOS team for an entire semester, our team was in the end responsible for only two files: pong.js and replay.js. The remainder of the code was mostly styling and, therefore, fell into their territory. JavaScript has no official documentation procedure, so we have attempted to mimic JavaDoc comments as best as possible.

1 pong.js

This file deals with the VirPong game: drawing, communicating (server and inputs), and all other necessary utilities.

1.1 Drawing Code

- function initCanvas()
  Starts the pong game and grabs the canvas so that we can modify it in JS.
- function displaySelection(selection, options)
  Serves as a filter for various display options. Game canvases, buttons, input options, and even game end screens are initialized here.
  @param selection a string representing the desired display: gameCanvas, gameCanvasWithButtons, inputMethodSelection, selectRoom, newRoom, or gameEnd
  @param options any of various options that go with selection
- function draw()
  Draws the game state.
- function drawHalfCourt()
  Draws a half court line.
- function drawPaddles()
  Draws paddles based on the field positions.
- function drawRect(a,b,c,d,col)
  Draws rectangles on the canvas.
  @param a top-left x-position
  @param b top-left y-position
  @param c bottom-right x-position
  @param d bottom-right y-position
  @param col color of the paddle
- function drawBall()
  Draws the ball at current position.
- function drawScore()
  Draws current score on the canvas.

1.2 Input Code

- function handleInputSelect(method)
  Initializes the appropriate input methods for playing a game.
  @param method a string for the desired input method.
  ”K” = Keyboard

”T” = Touchscreen
”W” = Wii Remote
”A” = Local Accelerometer

- function movePaddle(e)
  Recieve the input and send it to changePaddlePosition(), which actually changes the paddle position.
  @param e the event passed by the keypress.
- function changePaddlePosition(actualKey)
  Change the value of leftPaddle or rightPaddle so that it will draw in the correct place. This method is for a string input. ”W” is down and”S” is up.
  @param actualKey The string value of the key pressed.
- function detectViableInputMethods()
  Figure out what platform we”re running on, return the correct choices for input selection.
- function setupLocalAccerometer()
  Set the frequency of refresh for accelerometer data. Start watching acceleration and point ot it with watchID.
- function onSuccess(acceleration)
  Contains the work done each time acceleration is audited. Right now, we display the raw data with a timestamp, as well as the calculated position, which is taken from the getPosition function.
  @param acceleration An object containing the current acceleration values. (x,y,z,timestamp).
- function onError(acceleration)
  Fires off an alert if there’s an error if the collection of acceleration.

1.3 Server Code

- document.addEventListener(”DOMContentLoaded”, function()
  The ’document.addEventListener’ contains reactions to information sent by the server.
- function connectToServer()
  Sets up the connection to the server and registers multiple event listeners.
- joinRoom(room, clientType)
  Select the room to join.
  @param room a string representing the room name
  @param clientType player or spectator as a string
- function createRoom(roomName)
  Tells the server to make a room just for me.
  @param roomName the name of the room to create.
- function updatePaddleToServer(position)
  Update our paddle position with the server.
  @param position the new position of the paddle.
- function performAuthentication()
  Loads in login information from local storage to send to the server.
- handleNewRoom()
  Grabs the room name from an element roomName.roomName and submits it as a new room to the server.

2 replay.js

This file takes care of backend events for the replay system (still in beta): drawing and server communication.

2.1 Drawing Code

- function initCanvas()
  Stars the pong game and grabs the canvas so that we can modify it in JS.
- function draw()
  Draws the game state.
- function drawHalfCourt()
  Draws a half court line.
- function drawPaddles()
  Draws paddles based on the field positions.
- function drawRect(a,b,c,d,col)
  Draws rectangles on the canvas.
  @param a top-left x-position
  @param b top-left y-position
  @param c bottom-right x-position
  @param d bottom-right y-position
  @param col color of the paddle
- function drawBall()
  Draws the ball at current position.
- function drawScore()
  Draws current score on the canvas.

2.2 Server Code

- document.addEventListener("DOMContentLoaded", function()
  The 'document.addEventListener' contains reactions to information sent by the server.
- function connectToServer()
  Sets up the connection to the server and registers multiple event listeners.
- function viewGame(gameName)
  Select the room to join.
  @param room a string representing the room name.
  @param clientType player or spectator as a string.
- function createRoom(roomName)
  Tells the server to make a room just for me.
  @param roomName the name of the room to create.
- function updatePaddleToServer(position)
  Update our paddle position with the server.
  @param position the new position of the paddle.
- function performAuthentication()
  Loads in login information from local storage to send to the server.
- function displaySelection(selection, options)
  Serves as a filter for various display options. Game canvases, buttons, input options, and even game end screens are initialized here.

@param selection a string representing the desired display: gameCanvas, gameCanvasWithButtons, inputMethodSelection, selectRoom, newRoom, or gameEnd
@param options any of various options that go with selection