11.15.14

* Researching vectors, velocity, movement and acceleration (Nature of Code)
* Begin watching Shiffman videos
  + <http://vimeo.com/60601613>

11.17.14

* Made a player class that extends Shiffman’s Box
* Added foot sensor and jumping

11.18.14

* Met with Brandon to discuss sprite formatting
* Figured out how to access JBox2D JavaDoc
* Set player transform rotation to 0 to prevent spinning
* Implemented basic left/right movement

11.24.14

* Added Xbox controller config!
* Set friction = 0, fixing glitch in which player would “stick” to walls

11.25.14

Plan to implement scrolling:

* Make a final float SCROLL\_X. Initialize it = width \*.8 or width \* .9

This will determine the point on the screen at which the background will scroll and the player will advance

* Make a boolean scrolling. When player.xPos == SCROLL\_X, we will set scrolling = true
* When scrolling is true:
  + Player’s xPos is set equal to SCROLL\_X at the end of update() method
  + ­Move platforms and enemies left by an amount equal to player.MOVESPEED
  + Scroll background left by amount equal to player.MOVESPEED

11.26.14

* Scrolling detection is now implemented
* Dimensions have been updated to reflect tentative sprites
* Added a placeholder sprite
* Added a placeholder bg
* Added pseudocode for scroll() method

11.27.14

* Received some assets from Brandon
* Created method to display both pixel coordinates and Box2D world coordinates

11.28.14

* Going to take a different approach to scrolling
* Keep camera centered on player 1
* GOT CAMERA WORKING!!
* To do:
  + Implement bullets
  + Implement enemies (probably going to extend Box class)
  + Possibly implement keyboard controls
  + Map editor – Use bitmap or pixel
* Implemented sprites

12.2.14

* Added running and shooting animations
* Adjusted dimensions; Hitbox and character sprite match up
* Plan to implement pixelMap:
  + Make a png
  + For 1024x768, each screen will be 64 by 48 tiles
  + Create a (length)\*64 x 48 png
  + Zoom in and color pixels black for tiles
  + Read tiles and put a block there
* PImage levelMap;
* levelMap = loadImage(“map.png”);
* for (int x = 0; x < levelMap.width; x++) {
* for (int y = 0; y < levelMap.height; y++) {
* if (levelMap.get(x,y) == color(255,0,0) {
* //pixel at x,y -> make tile taking into account the size/spacing
* }
* }
* }