**Status Report**

David, Matt

**List of programs. Clearly describe the problem that you are solving. Please put the date that you worked on it:**

<https://github.com/DavidPeet8/Terarriag12>

<https://github.com/DavidPeet8/Learning-Racket>

David: <https://github.com/DavidPeet8/Learning-Racket>

<https://github.com/DavidPeet8/Terarriag12> code/releases/release7

Matt: <https://github.com/DavidPeet8/Terarriag12> code/releases/release6.1

David:

**Racket:**

Learn basic concept of racket (May 10 - now):

* One instruction per function
* No loops entirely recursive

Learn basic syntax of racket (May 10 - now):

* Comments
* Cond statements
* Defining variables
* Defining functions
* Idea of “Main” method

Look into idea of recursion (May 10 - now):

* Structural recursion
* Accumulative recursion
* Base case

Greatest common Factor:

* Recursive program to find GCF for 2 numbers

Fibonacci Sequence:

* Program to find the nth term in the fibonacci sequence

**Game:**

Inventory switching (April 15 - May 12):

* Add ability to switch spots in arrays of inventory and of hotbar by clicking on initial space and second space.
* Cannot move to an empty location

Matthew:

**Game:**

* I implemented Enemies in a release. This took longer than it should have because I forgot to draw it to batch and had no clue what was going on with it.
* I redid how all of the variants of Sprite worked. SpriteDiscrete and Enemy used to both extend Sprite and share a lot of the same basic functions. I made it so that SpriteDiscrete now has very limited functionality and Hero and Enemy now extend it. This should allow the making of new types of enemies easier in the future and cleans up the bloated SpriteDiscrete.

**Racket:**

* All the same stuff as David

**Major Challenges/setbacks( reference specific code please):**

David:

* Hard to gain some traction as they let you choose from one of too many different language variations on racket.
* Overall concept of racket is proving very annoying to program with → only one instruction per function, so many parentheses that you want to die
* Learning basic rules is annoying as testing takes multiple tries due to odd syntax
* REPL mode / window is very confusing as opposed to definitions window

Matthew:

* I forgot to draw my Enemies to batch and had no clue what was going wrong with them. I spend a week trying to debug them before figuring out I wasn’t drawing them.

**Source any web site/book that helped you with that concept:**

David:

* <https://www.youtube.com/channel/UChB6-Vdv00l5def5Iocwr4A?reload=9>
* Multiple of her videos
* Scheme book in classroom
* <http://www.ccs.neu.edu/home/matthias/HtDP2e/>

Matthew:

* <https://cs.uwaterloo.ca/~plragde/flaneries/TYR/> Waterloo site
* <http://www.ccs.neu.edu/home/matthias/HtDP2e/>

**Describe the code and the lesson that you learned from it:**

David:

**Old game:**

The code now has the added features of an inventory screen with background and menu screen with background. I also added the ability to switch item locations within the inventory screen. You can switch hotbar-hotbar, hotbar-inventory, inventory-hotbar, and inventory-inventory.

**Racket:**

The code I have does not do a whole lot, just created a few functions based on examples. I learned alot from this code though. I learned basic syntax of how to comment, define and populate a variable, define a function, call a function, and use built in functions. In addition, I learned the basic concept of recursion including the base case, and the difference between structural and accumulative recursion. As well I learned the general structure of a lot of control flow functions. These include if statements, cond statements, custom functions, and nesting.

Matthew:

I learned about pairs and lists in Racket. I also learned a variety of functions related to them such as cons to make pairs and car and cdr to get certain parts of them.