**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>

David:

Code/Releases/Release2

Random World Generation (Feb 7 - March 2)

* randomly generate a 2D terrain
* Use Perlin Noise to create organic look and feel to terrain
* store map in json file so we do not run into memory issues
* seed the noise function
* Draw array of boxes under the top terrain layer
* Randomize the block type under surface layer intelligently
* Scale values based on tile width and height
* Draw map to screen based on players position using orthographic camera (initially)

Mining and placing (March 2 - now)

* Make blocks breakable by switching their contese in the array to null
* Make only blocks within players direct vicinity affectible
* Get data from json as to hitpoints of specific block you are breaking

Orthographic Camera(March 8 - March 14)

* Get orthographic camera to follow player on move, not just initially set
* Look into cameras and viewports and possibly both with the scalability they provide

Fixed Background

* Get a background image that will scale with window size
* Background does not move with camera

Clean and restructure code (March 2 - March 10)

* Clean Matt’s first few scratches - variable names, file names, unused code, etc.
* Clean my release 1 code, - variable names, remove matts scratches from the release, (they are not final features yet, should not be in the game)
* Restructure code - use Abdullah’s multiple projects in one repository approach so that we do not create a billion repositories, and so that we do not step on one another's toes
* Solve all git issues, learn the bash command prompt so that we may push and pull a repository that is not a project
* Clean my release 2 code

Add screen switching(Feb 23 - March 2)

* Use arrow up and down keys to toggle between screens

Matthew:

Side Hit Detection (Feb 7 - March 5)

Code/Scratches/HitWithSides

Implemented hit detection where the player and the walls have sides. Messed around with many different ways of doing this including having sides on just the walls, just the player, both, changing the sizes of sides, and changing the positioning of sides.

* Tried using this hit detection with an array of walls and things fell apart. There was always at least one awful glitch no matter how simple or complicated I attempted to make the side detection.
* Abandoned this form of hit detection because it falls apart in the corner cases.

Discrete Hit Detection (March 5 - March 19)

Code/Scratches/DiscreteHit

* Implemented a form of hit detection where you check to see how far you should move before you do. It is handled through a recursive function that shortens the distance you are going to move by one if you would hit something when you move.

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

David:

- Screen switching required me to rearrange pieces of code such as putting the set input processor into show instead of the constructor so that the input processor gets updated upon screen change, took a few days to figure out the interesting problems I was having as I never actually got any errors, just not the results I wanted (Keydown of all screens that display)

- Solving git issues - we had many partial conflicts and other odd things happening with the git initially when we went to the new format

- Orthographic camera is setting to the position of the player initially, but will not follow the player (fixed)

- figuring out what part of our code was not being used anywhere and deleting it.

- separating matts scratches from the game and vice versa proved tedious, a lot of file moving was required

- somehow with orthographic camera, updating one classes instance of a camera passed in updated all classes instances of that camera that was passed in - decided to give each screen its own camera (changed, happens when I tried to pass one camera to multiple classes)

- orthographic camera; needs to have batch.setProjectionMatrix called every frame, for a long time I was under the impression once it was called it stayed and did not need to be called again (ScrPlay - updateCam method)

-background image was scaling on window resize poorly as it was double scaling; the gdx.graphics.getWidth / Height is dynamic, and changes on resize, and the viewport expects it not to resize as it will do all of the scaling for us (ScrPlay - drawMap)

-mining based on json values for durabilities is heavily tied to inventories and hotbars, making this a huge task

Matthew:

* The hit detection with sides wasted a lot of time because I always had at least one major issue with it. It always felt close to complete so I stuck with it for longer than I should have.
* Making the recursive function for discrete hit detection (in the SpriteDiscrete class) took longer than I would have liked. I couldn’t figure out why the program was getting stuck for awhile.

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

David:

* No links really necessary spent much of the time restructuring project and renaming and cleaning different projects

Matthew:

* [**https://katyscode.wordpress.com/2013/01/18/2d-platform-games-collision-detection-for-dummies/**](https://katyscode.wordpress.com/2013/01/18/2d-platform-games-collision-detection-for-dummies/)I learned about the theory of discrete hit detection from there.

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

David: The new parts of my program include the ability to toggle between screens using the up and down arrow keys, this part taught me that the game screen does not run in the background and to do the screen toggling properly, the code to flip screens must be replicated over every screen. As well my new program will allow the user to break blocks in the vicinity of the player on the screen by clicking and dragging that click. I learned how to use a try catch statement for this part of the code. I used this statement to make short work of bounds checking, if you click to break a block outside of the populated array, eg negative array indexes, the error is simply ignored. This is a much more readable way to check for these errors as opposed to creating more complicated code to check for this. My code also now has a movable player that is followed by the camera; the background scrolls behind the player. Control the player using WASD. From this I learned that set projection matrix only sets for the upcoming batch draw not all draws using that batch.

Matthew: I learned about recursion from doing discrete hit detection. The loop for that can be found in the SpriteDiscrete in CommonClasses in the scratch for discrete detection.

With each status report, you will be submitting EVERYTHING. Organization is key. When I go to the groupwork folder**, I should see your project submitted in the following format:**

YourLastName: Under this folder will be the following folders:

**Documents**: It will hold all of your documents: status report, list of sources, and all the other documents that will be submitted in your final project.

**Programs**: There will be a folder for each project.