**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/Release6 Code/Releases/Release(beta)

Matt: Code/Scratches/Enemies Code/Releases/Release5.1

David:

Mining and placing (April 6 - April 15)

* Add blocks to players inventory when they are broken intelligently. Add first to any index that already contains that type of item if it is below stack limit. If that is not possible add to next null index
* Placing blocks by checking if the tile index the mouse is on is null, then converting our item to a tile and adding it to the array

Inventory set up (April 6 - April 10)

* Give player items to begin the game

Add active area to hotbar(April 6 - April 11)

* Essentially create a hotbar and have an active slot of your hotbar so that you can only perform actions with what you are holding
* Modified to be simply an index in hot bar instead of its own item
* Give ability to switch between hotbar slots by using the scroll wheel

Created large External textures and Json Class

* Only load these once
* Access from this static class

Fix bugs(April 11-now)

* Placing blocks inside yourself
* Blocks stack going down when click on non null tiles
* Bounds checking on placing
* Change array of tiles subset origin so the hit detection area is evenly distributed around the player
* Decrease from stack of an item properly, prevent scrolling away and back to a now null item gives you more of it

Set up ScrInventory and ScrMenu (April 15 - April 18)

* Make them valid screens that we can toggle to and from
* Make inventory accessible by pressing ‘i’
* Change display on screen switch so we know we are getting there

Clamp world (April 16)

* Clamp world to all sides
* Prevent player from leaving the window

HUD(April 15 - April 26)

* Create Health and skill / Mana bars
* Appear fixed on screen based on character health and Mana
* Draw Hotbar on screen with an icon depicting which is active

Change to polling click hit detection instead of event driven (April 23)

* Convert the logic into a function called on both touch down and touch dragged
* Add left mouse to array, add it in key action so it is checked every frame in a polling way

Efficient drawing of map(April 23)

* Make map only draw in general vicinity of player
* Tried making it draw based on camera x and y, did not work when scaled
* Did the math for bounds so the entire screen has blocks being drawn
* Had to do bounds checking for trying to draw outside of world as it was heavily slowing down the game

Efficient hit detection(April 23)

* I looked at Matts hit detection code, I did not like how we seemed to have to arbitrarily loop through the checks 5 times. I think if it is doing its check properly, and is being recursively called properly, one time should be sufficient
* Changed architecture slightly, the recursive calls were in the wrong spot
* Now definitively works when it only does each check once

Rework Screen setup (April 20 - 21)

* Add menu screen, make space flip to next screen
* Screens create next screen in sequence, this allows for us to die and have the option of respawning in the current game, or to return to the main menu to start a new game

Add death screen(April 20)

* Add screen that is toggled to when player health reaches 0
* Gives options to respawn in current game, or to go to the menu and create a new game

Matthew:

Bullet Through Paper Problem

* Looked into improving hit detection further with Speculative Hit Detection to solve the bullet through paper problem.
* After reading up on this for about two weeks and trying to implement it in scratches I decided it is really not worth the effort. I might get back to this later because it is very neat but I’m done with hit detection for now.
* <https://katyscode.wordpress.com/2013/01/18/2d-platform-games-collision-detection-for-dummies/>
* <https://wildbunny.co.uk/blog/2011/03/25/speculative-contacts-an-continuous-collision-engine-approach-part-1/>
* <https://github.com/mattleibow/jitterphysics/wiki/Speculative-Contacts>

Enemies

* I created an enemy that extends SpriteDiscrete. It randomly moves and jumps. Can be seen in code/scratches/enemies
* After creating the enemy I reduced how much SpriteDiscrete does and now have a class for the hero called SpriteHero. I think splitting it up this way will be more helpful if in the future we want more types of enemies or if our current enemy and hero get more functionality. Can be seen in code/releases/release5.1 The enemies do not work in here yet
* To make the enemy work I also had to stop infinite jumping in SpriteDiscrete this happened in code/releases/release5.1

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

David:

* The issue of storing my array of tiles and having to convert an item into a tile(ScrPlay clickdown())
* A Lot of my checks for the tool type being right to mine a tile were causing crashes (ScrPlay clickdown())
* Could anchor yourself in place by placing a block inside yourself(not solved no specific area)
* Stack for blocks being placed was going down, but when it hit null, it wasnt updating the hotbar location. Therefore scrolling away and back gives you more of the item (InventoryObj addto())
* Bounds checking when trying to mine or place outside of the world(ScrPlay clickDown())
* Origin for creating our subset array was off, still bottom left. This created problems when doing hit detection on the op of our sprite as the array didn't actually reach the top(ScrPlay createSubset())
* Stopping player from going outside world bounds got aggravating. Was not setting players position same as Matt, lead to him eventually falling through the map.(ScrPlayer boundsCheckPlayer())
* I still cannot figure out why the snappy camera is a thing
* A Lot of annoying efficiency problems, ex hit detecting with loops, drawing the entire map every frame(ScrPlay createSubset())

Matthew:

* I got set back on enemies by trying to get them to intelligently move based on what tiles were near them. I scrapped that idea and have them move based off of a randomly generated number.
* I made a pile of Speculative Hit Detection scratches that all had very interesting flaws. The hardest part of Speculative Hit Detection is scaling it up to work with more tiles. I tried using a bunch of slightly different methods that broke when scaled up. I think if I want to get it to work in the future I would use a 2D vector to store my position and another one to store my direction.
* I copied a pile of wrong code into release5.1 when I started creating the new classes that extend SpriteDiscrete. I thought something was going wrong with how I was doing the extensions but I just old versions of functions that didn’t work.

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

David:

* <https://gamedev.stackexchange.com/questions/74926/libgdx-keep-camera-within-bounds-of-tiledmap>
* <https://libgdx.badlogicgames.com/nightlies/docs/api/com/badlogic/gdx/graphics/glutils/ShapeRenderer.html>
* <https://stackoverflow.com/questions/886955/breaking-out-of-nested-loops-in-java>

Matthew:

* <https://katyscode.wordpress.com/2013/01/18/2d-platform-games-collision-detection-for-dummies/>
* <https://wildbunny.co.uk/blog/2011/03/25/speculative-contacts-an-continuous-collision-engine-approach-part-1/>
* <https://github.com/mattleibow/jitterphysics/wiki/Speculative-Contacts>

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

David:

My code now includes, a player that can place blocks he has mined by clicking and by dragging, with proper limitations in place. The player now also hit detects properly on all sides. The inventory screen may also be reached using ‘i’ or the arrow keys. From the item to tile problem I have learned about the transient keyword that essentially prevents that property from being saved. I have also learned the benefits of having a external static textures class. I have also learned a little bit about how to approach large tasks like setting up an entire inventory for a player. In addition, the problems I ran into for placing blocks taught me more about being able to read through your code step by step, as well as using print lines and the debugger to find your error. I had an issue with breaking out of nested loops which taught me how to break to labeled loops, not just break. One main thing to note: I did change how screen changing works, now each screen creates the subsequent screen, therefore you must travel through the screens in the correct order, screen binding in the first two screens is space. I did this because to take an inputted seed, we need the load screen to be created after the menu screen has gotten the seed.

Matthew:

My code has an enemy randomly jump around. I also broke the old SpriteDiscrete class into SpriteDiscrete and classes that extend it in release5.1. This will ideally make making new types of enemies easier in the future and reduces the size of each class. I also learned I need to break up my functions more. I wanted to start adding hit detection between the SpriteHero and Enemy but it would be easier if my hit detection was more broken up.