Work in Progress Report 3

by: Tim and Thomas

GitHub repository: https://github.com/NoFlintGrey/SBF-Wip3.git

Major developments/breakthroughs(reference specific code please):

* window scaling (using a Stretch Viewport)
* A player to move around the screen (only faces one direction right now)
* A static (non moving) enemy to kill
* Finally fixed the problem with the y coordinate being inverted, and got the Touch To Move Scratch working

Code for viewport

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*// set nVieW*

***nViewW****=320;*

*// get screen size and set aspect*

**dAspect** = Gdx.*graphics*.getWidth()/(Gdx.*graphics*.getHeight());

**nViewH** = (**int**)(**nViewW**/**dAspect**);

*// viewport*

**camera** = **new** OrthographicCamera();

**viewport** = **new** StretchViewport(**nViewW**,**nViewH** ,**camera**);

**viewport**.apply();

**camera**.**position**.set(500, 600, 0);

//

**public void** resize(**int** nWidth, **int** nHeight){

**dAspect** = nWidth/nHeight;

**nViewH** = (**int**)(**nVeiwW**/**dAspect**);

**viewport**.update(**nViewW**,**nViewH**);

**camera**.**position**.set(**camera**.**viewportWidth** / 2, **camera**.**viewportHeight** / 2, 0);

System.***out***.println(**"resized"**);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

code for player to move around and staying centered in the view port

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//

**nPx** = (**int**)500+(**int**)(**camera**.**viewportWidth**)/2;

**nPy** = (**int**)600+(**int**)(**camera**.**viewportHeight**)/2;

//

**camera**.translate(**nDx**, **nDy**);

**nPx**=**nPx**+**nDx**;

**nPy**=**nPy**+**nDy**;

//

**batch**.begin();

**batch**.draw(**imgPlayer**, **nPx**, **nPy**);

**spMenu**.draw(**batch**);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Code for enemy (might be different in git hub)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//

int nHealth =10, nAttack=5;

//

**batch**.begin();

**batch**.draw(**imgEnemy**,600, 600);

**spMenu**.draw(**batch**);

if (arbInput[5]){

// checking if you are close enough to attack

if(nPx> 600-20|| nPx<600+20+20||nPy> 600-20|| nPy<600+20+20){

nHealth=nHealth-nAttack;

}

}

if (nHealth==0){

**batch**.begin();

**batch**.draw(**imgEnemy**,-1100000, 600);

**spMenu**.draw(**batch**);

//

imgEnemy.dispose();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Code for reverting the y

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**if** (Gdx.*input*.isTouched()) {

Vector3 touchPos = **new** Vector3();

touchPos.set(Gdx.*input*.getX(), Gdx.*input*.getY(), 0);

**fTouchX** = (touchPos.**x**);

**fTouchY** = (**camera**.**viewportHeight**\*3)-(touchPos.**y**);

**sprPlayer**.**x** = **fTouchX** \* Gdx.*graphics*.getDeltaTime();

**sprPlayer**.**y** = **fTouchY** \* Gdx.*graphics*.getDeltaTime();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Major Challenges/setbacks( reference specific code please):

We have still yet to find a way to load in polygons from out tiled.tmx map file that works with our code, though we believe that we should be able to do this some time soon, whether or not it it loaded directly from the tiled map file, or if we will only be using that as a reference to create our own without using the tiled map file. Continuing with the theme of tiled map files causing us unnecessary harm, tiled map files become distorted if using a viewport mode other than the extended viewport (which is the same as using no viewport at all). Which why we are now looking at other map editors.

Any modifications to your specifications/release schedule:

Changes to the release schedule are:

* only one level
* no story
* programing art only
* only basic sound (background music and a few other (attack, dying, enemy))

Source any web site/book that we used for the final build for this release :

*http://www.gamefromscratch.com/post/2014/05/01/LibGDX-Tutorial-11-Tiled-Maps-Part-2-Adding-a-character-sprite.aspx*

http://www.gamefromscratch.com/post/2014/12/09/LibGDX-Tutorial-Part-17-Viewports.aspx

*the deep dark taurock*

[*www.gamefromscratch.com/post/2014/12/09/LibGDX-Tutorial-Part-17-Viewports.aspx*](http://www.gamefromscratch.com/post/2014/12/09/LibGDX-Tutorial-Part-17-Viewports.aspx)

[*http://stackoverflow.com/questions/3342651/how-can-i-delay-a-java-program-for-a-few-seconds*](http://stackoverflow.com/questions/3342651/how-can-i-delay-a-java-program-for-a-few-seconds)

*// tried to use this http://www.bigerstaff.com/libgdx-touchpad-example/*

**Description of your scratch/test program:**

Describe the generic concept you needed to test out:

Scale an image using a viewport.

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

http://www.gamefromscratch.com/post/2014/12/09/LibGDX-Tutorial-Part-17-Viewports.aspx

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

Lessons learned : that Scaling images to fit the screen is easier than we thought and it allows us to use a cleaner way of keeping the menu screen in the center.

Describe any challenges that you enjoyed in integrating this scratch code into your major project:

There were a few challenges in integrating the viewport code; scaling our menu image was not challenging, the challenges, came from the use of the tiled map loader and the view port, as it currently still distorting the image, and getting to resize properly (this was disabled for now will have a menu selection later )

Describe the generic concept you needed to test out:

Move an image around the map using a touch sensor

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

<https://github.com/Mrgfhci/Drop/blob/master/core/src/com/mygdx/drop/Drop.java>

https://github.com/Ticken/SBF-Drop/blob/master/core/src/com/mygdx/drop/Drop.java

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

Lessons learned : That the y inversion was a result of the Tiled Map orientation and if you took the height of the Viewport and took away the touch position of the y you could invert it back to normal.

Describe any challenges that you enjoyed in integrating this scratch code into your major project:

Attempted to make it work by manually changing the Tiled Map settings which caused a bug which caused the program to stop working.

We forgot to remove the Fill function and it made the background red

Thomas’ computer was changed to Linux and stalled progress as well as rendering the SDK incompatible

Using Sprites caused the .x, .y, .set functions to not work as it did with rectangles