Work in Progress Report 1

by: Tim and Thomas

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

Major developments/breakthroughs(reference specific code please):

Major developments for this WIP were:

1. the implementation of a onscreen pop-up joystick which controls both the player and the camera.

which because we could not get the build in touchpad class to work, had to be made from scratch, though there is a bit of camera drift (it does not keep the player in it’s center.),

1. The movement of the player and the camera, in this release we got the player and the camera to move together (ignoring the camera drift), however there seems to be considerable decrease in performance when moving in certain directions. As of yet we do not know the cause of this.
2. The last major development in this progress report was using tiled for map creation and loading the tiled.tmx file and its dependencies.

Code for number for the pop-up joystick.

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

@Override

**public boolean** touchDown(**int** screenX, **int** screenY, **int** pointer, **int** button) {

*// move joystick into view*

**if** (screenX<Gdx.*graphics*.getWidth()/2){

**nJoyStickX** = screenX-100;

**nJoyStickY** = Gdx.*graphics*.getHeight()-screenY-100;

**nJoyStickNubX** = **nJoyStickX**+64;

**nJoyStickNubY** = **nJoyStickY**+64;

}

**return false**;

}

@Override

**public boolean** touchDragged(**int** screenX, **int** screenY, **int** pointer) {

*// find and move the nub if it's on the left side of the screen*

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

**if** (screenX<Gdx.*graphics*.getWidth()/2) {

**nJoyStickNubX** = screenX - 36;

**nJoyStickNubY** = Gdx.*graphics*.getHeight() - screenY - 36;

*// limit the nub*

**if** (**nJoyStickNubX** > **nJoyStickX** + 164) {

**nJoyStickNubX** = **nJoyStickX** + 154;

}

**if** (**nJoyStickNubX** + 36 < **nJoyStickX**) {

**nJoyStickNubX** = **nJoyStickX** - 36;

}

**if** (**nJoyStickNubY** > **nJoyStickY** + 164) {

**nJoyStickNubY** = **nJoyStickY** + 154;

}

**if** (**nJoyStickNubY** + 36 < **nJoyStickY**) {

**nJoyStickNubY** = **nJoyStickY** - 36;

}

*// moving the player*

**nDx** = ((**nJoyStickNubX** + 36) - (**nJoyStickX** + 100)) / 50;

**nDy** = ((**nJoyStickNubY** + 36) - (**nJoyStickY** + 100)) / 50;

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

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

}

**return false**;

}

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

Code for moving the player and camera

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

from: touchDragged

*// moving the player*

**nDx** = ((**nJoyStickNubX** + 36) - (**nJoyStickX** + 100)) / 50;

**nDy** = ((**nJoyStickNubY** + 36) - (**nJoyStickY** + 100)) / 50;

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

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

and:

@Override

*// updating the camera and render objects*

**public void** render () {

**camera**.**position**.set(**nPx** -212,**nPy**-117,0);

**camera**.update();

**tiledMapRenderer**.setView(**camera**);

**tiledMapRenderer**.render();

*// joystick and buttons*

**batch**.begin();

**batch**.draw(**imgBackground**, **nJoyStickX**, **nJoyStickY**);

**batch**.draw(**imgNub**, **nJoyStickNubX**,**nJoyStickNubY**);

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

**batch**.end();

}

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

code for tiled:

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

from create:

**tiledMap** = **new** TmxMapLoader().load(**"MyMap.tmx"**);

**tiledMapRenderer** = **new** OrthogonalTiledMapRenderer(**tiledMap**);

Gdx.*input*.setInputProcessor(**this**);

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

Major Challenges/setbacks( reference specific code please):

The biggest challenge for this release was trying to get collision with the wall and player. This is still a challenge for us as we do not yet have a method for reading in the polygons from the tiled.tmx file, and will be our next task. Other Challenges were getting trying to get the touchpad based joystick to work, before we build our own from scratch.

code for trying to grab the polygons:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*// poylgons*

*/\*\* from https://www.youtube.com/watch?v=Mx\_vQU\_rMCQ*

*for (MapObject object: tiledMap.getLayers().get("Object Layer 1").getObjects()){*

*if (object instanceof RectangleMapObject){*

*Rectangle rect = ((RectangleMapObject)object).getRectangle();*

*} else if (object instanceof EllipseMapObject){*

*} else if (object instanceof PolylineMapObject){*

*} else if (object instanceof PolygonMapObject ){*

*Polygon Poly = ((PolygonMapObject)object).getPolygon();*

*}*

*}*

*\*\*/*

*// from https://github.com/libgdx/libgdx/wiki/Tile-maps*

*// MapLayer layer = tiledMap.getLayers().get("Object Layer 1");*

*// PolygonMapObject polyObject = new PolygonMapObject();*

*// Polygon wall1 = polyObject.getPolygon();*

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

Any modifications to your specifications/release schedule:

Release v 0.1 and v 0.2 were combined all other release remain the same though a bit behind schedule.

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

// unused for code but helpful

<http://gamedevelopment.tutsplus.com/tutorials/introduction-to-tiled-map-editor-a-great-platform-agnostic-tool-for-making-level-maps--gamedev-2838>

<https://www.youtube.com/watch?v=zckxJn751Gw>

<http://www.javacodegeeks.com/2012/05/android-game-development-with-libgdx_03.html>

<http://stackoverflow.com/questions/15185799/libgdx-get-swipe-up-or-swipe-right-etc>

<http://gamedevelopment.tutsplus.com/tutorials/parsing-tiled-tmx-format-maps-in-your-own-game-engine--gamedev-3104>

<https://github.com/libgdx/libgdx/wiki/Tile-maps>

http://www.java-gaming.org/topics/mapobjecttobox2dparser-parse-tiledmap-to-box2d/29937/view.html

// used

<https://www.youtube.com/watch?v=Mx_vQU_rMCQ>

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

<http://gamedevelopment.tutsplus.com/tutorials/introduction-to-tiled-map-editor-a-great-platform-agnostic-tool-for-making-level-maps--gamedev-2838>

<http://www.gamefromscratch.com/post/2014/04/16/LibGDX-Tutorial-11-Tiled-Maps-Part-1-Simple-Orthogonal-Maps.aspx>

<http://www.bigerstaff.com/libgdx-touchpad-example/> (used to help build my own joystick)

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

Describe the generic concept you needed to test out:

Input with touch

Sounds

Orthographic camera

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

https://github.com/libgdx/libgdx/wiki/Orthographic-camera

[*http://www.freesound.org/people/acclivity/sounds/28283/*](http://www.freesound.org/people/acclivity/sounds/28283/)

*http://www.freesound.org/people/junggle/sounds/30341/*

*https://app.box.com/s/605bvdlwuqubtutbyf4x*

*https://app.box.com/s/peqrdkwjl6guhpm48nit*

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

How to use setDeltaTime to make the movement as smooth as possible and how it sets it based on the time and speed to make it not stutter(raindrop.**y** -= (150 + 2\***nScore**) \* Gdx.*graphics*.getDeltaTime();)

How to loop music to make it continue(setLooping)

What an iterator is(lets you take control of the container ex.array)

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

There wasn't really any challenges integrating the main scratch code (camera, and touch a single point) the challenges came from things that were not it the scratches, such as Tiled, for creating the collision polygons.