Work in Progress Report 2

by: Tim and Thomas

GitHub repository: https://github.com/NoFlintGrey/SBF--Wip2.git

Major developments/breakthroughs(reference specific code please):

1. multiple files one for the main game code and one for input (one for mobile and one for desktop)
2. fixed the controls for desktop, no longer can the mobile joystick popup when running in desktop mode, and holding down keys will now move continuously.

Code for DeskTop controls contained in Desktop\_launch.java

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

@Override

**public boolean** keyDown(**int** keycode) {

**if** (keycode == Input.Keys.***LEFT***){

**arbInput**[4] = **true**;*//left*

}

**if** (keycode == Input.Keys.***RIGHT***){

**arbInput**[3] = **true**;*// right*

}

**if**(keycode == Input.Keys.***UP***) {

**arbInput**[1] = **true**;*// up*

}

**if**(keycode == Input.Keys.***DOWN***) {

**arbInput**[2] = **true**;*//down*

}

**game**.input(**arbInput**);

**game**.game();

**return false**;

}

@Override

**public boolean** keyUp(**int** keycode) {

**if** (keycode == Input.Keys.***LEFT***){

**arbInput**[4] = **false**;*//left*

}

**if** (keycode == Input.Keys.***RIGHT***){

**arbInput**[3] = **false**;*// right*

}

**if**(keycode == Input.Keys.***UP***) {

**arbInput**[1] = **false**;*// up*

}

**if**(keycode == Input.Keys.***DOWN***) {

**arbInput**[2] = **false**;*//down*

}

**game**.input(**arbInput**);

**game**.game();

**return false**;

}

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

Code for the Touch To Move Scratch

**camera**.position.set(**fPx** -212,**fPy**-117,0);

**camera**.update();

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

**tiledMapRenderer**.render();

Gdx.gl.glClearColor(1, 0, 0, 1);

Gdx.gl.glClear(GL20.GL\_COLOR\_BUFFER\_BIT);

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

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

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

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

**fTouchY** = (touchPos.y);

}

**batch**.begin();

**batch**.draw(**imgDude**, **fTouchX**-(**imgDude**.getWidth()/2) , **fTouchY**-(**imgDude**.getHeight()/2));

**batch**.end();

}

Major Challenges/setbacks( reference specific code please):

We faced a number of setbacks this past 2 or so weeks, mainly down to again trying to get objects from a tiled map file, trying the implement scene2d and a file management system similar to the Deep Dark Taurock, and view ports, as of now we don’t have objects from out tiled map file, we do not have any scene2d , but we do have a file system that is more like the Deep Dark Taurock now, that make scratch integration far easier. We also don’t have a view port enabled in our code, but for some reason it seems to work on multiple screen sizes, and resolutions without the viewport. Devices tested: 1080p nexus 5 (emulator), oneplus one (real device) ,and 1440p nexus 6p (emulator). In the Touch To Move scratch(when it worked) the y was flipped around the x-axis of the viewport and so it would not go where we touched. Also the code doesn’t work after we tried to isolate it from the larger project.

Bugs:

there are still a large number of known bugs in our game:

1. You can move the camera off the map and and encounter the “right of Stage bug” this bug causes the joystick to remain on screen after you stop touching the screen, whenever you move out of the map and bring up the joystick,
2. Sometimes nothing happens we just get a black screen and we have to restart.
3. Sometimes nothing happens and it shows yellow lines or a just a yellow background, and we have we have to restart.

Things we need to watch in the future:

Just a little side note for use be we need to start watching our file sizes of our assets our game is up to ~50MB in size with almost nothing done, so for now on we will have to use compressed images and not uncompressed images to save on space.

Any modifications to your specifications/release schedule:

No longer considering more than one level, as we still don’t have walls, and a player infact we almost went backwards in progress since the last WIP but we have a far better foundation to build more elements to our game then we did before.

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*

*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:

1. Desktop controls
   1. git for this is here https://github.com/NoFlintGrey/Desktop-Scratch-.git
2. scene2d intergradation and file management similar to the Deep Dark taurock
   1. Git for this is here : <https://github.com/NoFlintGrey/scene2d-scratch.git>
3. A touch to move scratch where the x and y change on where you touch
   1. Git for this is here :https://github.com/Ticken/Touch-To-Move-Scratch.git

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

The Deep Dark Taurock: <https://github.com/MrGFHCI/TheDeepDarkTaurock> (this was used for Scene2d, and a bit if it was used in the Desktop controls)

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*

(used for the touch to move it came from the drop code)

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

Lessons learned :

1. Multiple for a game files make scratch integration a lot easier
2. We can have the controls/ input listers located in a file out of the main game file
3. The file system in the Deep Dark Taurock might be a little much for what we have now, but we should continue to keep it in the back of our minds when moving forward
4. The Input Processor requires lots of different methods many of which don’t need to do anything
5. Vector3’s are used yet the z-axis is unused, which may allow for cleaner code should Vector2’s be used.
6. Using Tiled Maps is much cleaner than using normal Maps

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

The desktop controls were really easy to integrate now that we are using multiple files, and we had no challenges, scene2d was on the other hand a complete failure and is not integrated at this point in time, mainly due to the fact that the scene2d scratch based off the The Deep Dark Taurock is not working at this moment,and now uses a different file management system. But what we summited as a our wip 2 (better desktop controls and splitting the game into more than one file) runs fine on a number of devices.

The Touch to Move scratch was working well except that the y was on the opposite side of the screen, we suspect the tiled orientation to be the problem. However during the moving off the code from an older version of the Final Project to a scratch (the current one) the scratch now doesn’t run on the kobo.