Work in Progress Report 5

by: Tim and Thomas

GitHub repository: [https://github.com/NoFlintGrey/SBF-Wip5.git](https://github.com/NoFlintGrey/SBF-Wip4.git)

Major developments/breakthroughs(reference specific code please):

* Fixed screen switching
* Added character selection screen
* fixed main menu
* can no longer walk out of the map (we fully integrated the hit detection from a few wips ago)
* hud touched up
* sound integrated
* basic enemy ai
* more levels (one more)---

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

In main file: @Override

**public void** create() {

game = **this**;

setScreen(**new** MainMenu(game));

}

**public void** render() {

**super**.render();

}

in all other screen files

**public** MainMenu(Game game){

**this**.game = game;

}

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

fixed main menu

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

// button that launches the Charter Select menu;

**tbNewGame** = **new** TextButton(**"New Game"**, **tbsNewGame**);

**tbNewGame**.setSize(Gdx.*graphics*.getWidth()/3, Gdx.*graphics*.getHeight()/6);

**tbNewGame**.setPosition(Gdx.*graphics*.getWidth()/2-(Gdx.*graphics*.getWidth()/3)/2, Gdx.*graphics*.getHeight()/2-Gdx.*graphics*.getHeight()/6);

**tbNewGame**.addListener(**new** InputListener() {

@Override

**public boolean** touchDown(InputEvent event, **float** x, **float** y, **int** pointer, **int** button) {

**game**.setScreen(**new** Characterselect(**game**));

**return true**;

}

});

**stage**.addActor(**tbNewGame**);

@Override

**public void** render(**float** delta) {

**batch**.setProjectionMatrix(**camera**.**combined**);

**camera**.update();

**stage**.act();

**batch**.begin();

**spMenuBackground**.draw(**batch**);

**batch**.end();

**stage**.draw();

}

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

Added character select screen:

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

selection buttons

*//*

**tbLeft** = **new** TextButton(**"<"**, **tbsNewGame**);

**tbLeft**.setSize(Gdx.*graphics*.getWidth() / 5, Gdx.*graphics*.getHeight()/6);

**tbLeft**.setPosition(Gdx.*graphics*.getWidth() / 18, (Gdx.*graphics*.getHeight() /5)\*3 );

**tbLeft**.addListener(**new** InputListener() {

@Override

**public boolean** touchDown(InputEvent event, **float** x, **float** y, **int** pointer, **int** button) {

**if** (**nPlayer** > 1) {

**nPlayer**--;

**player**.playerset(**nPlayer**);

} **else** {

**nPlayer** = 4;

**player**.playerset(**nPlayer**);

}

**return true**;

}

});

**stage**.addActor(**tbLeft**);

*//*

**tbRight** = **new** TextButton(**">"**, **tbsNewGame**);

**tbRight**.setSize(Gdx.*graphics*.getWidth() / 5, Gdx.*graphics*.getHeight()/6);

**tbRight**.setPosition(Gdx.*graphics*.getWidth() - Gdx.*graphics*.getWidth() / 18 - Gdx.*graphics*.getWidth() / 5, (Gdx.*graphics*.getHeight() /5)\*3 );

**tbRight**.addListener(**new** InputListener() {

@Override

**public boolean** touchDown(InputEvent event, **float** x, **float** y, **int** pointer, **int** button) {

**if** (**nPlayer** < 4) {

**nPlayer**++;

**player**.playerset(**nPlayer**);

} **else** {

**nPlayer** = 1;

**player**.playerset(**nPlayer**);

}

**return true**;

}

});

Game launch and back button were done the same way we launched the character select screen.

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

Basic Ai (code was on desktop at home, so this is just a summary)

// checking if player is within aggro range in about a 20 x 20 box around the enemy

if (player.x< Enemy[I].x-20&& player.x> Enemy[I].x+41&&player.y< Enemy[I].y-38&& player.y> Enemy[I].y+38){

// use almost the same code as the joystick to get the direction to go just really basic and needs to be replaced with something better

**nDx** = ((**Player.x**) - (**Enemy[I].x**));

**nDy** = ((**Player.y**) - (**Enemy[I].y**));

**if** (**nDy** > 0) {

// set movement code

}

**if** (**nDy** < 0) {

// set movement code

}

**if** (**nDx** > 0) {

// set movement code

}

**if** (**nDx** < 0) {

// set movement code

}

if (nDx>100||nDy>100){

// tell enemy I to stop moving and go into a rest mode

}

}

Major Challenges/setbacks( reference specific code please):

* The code has yet to work in the Android Emulator, but it works in the Desktop.
* Ai is buggy
* Wall detection sometimes stops working

Any modifications to your specifications/release schedule:

* No story
* other classes will be just reskins of the main player class (same attacks and health)

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/*

*// tried to use this as well //http://stackoverflow.com/questions/4861859/implement-sound-in-android-application*

*//http://developer.android.com/reference/android/media/MediaPlayer.html*

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

**Basic AI attack within a engage range, move towards the player**

**git: https://github.com/NoFlintGrey/Basic-Ai**

Describe the generic concept you needed to test out:

Basic Ai

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

<http://www.raywenderlich.com/24824/introduction-to-ai-programming-for-games>

The rest came from already known knowledge related to building basic game enemy ai.(all it its checking if the player is within aggro range then following them by checking their position and setting our enemy to go to that spot

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

You need a max distance away that the enemy will still follow, or it will constantly get stuck on walls, Walls can be a difficult obstacle for ai to move around, while still being useful to attack. Using a large amount of weak enemies in a swarm works really well to get around some of our obstacle issue (because at least some of them will get through the doorway)

Describe any challenges that you enjoyed in integrating this scratch code into your major project: Originally we wanted the ai to be contained in the enemy's class file, but we were unable to get this to work while it was placed here, moving it to the main game screen file worked though, However we will still try to get it working while in the enemey class file over the next week.