Game development documentation

The first thing I did when improving the game was to make sure the player couldn't leave the screen. I did this by having it so that when any one of any SFAsset's directional methods get called, such as GoWest. I would get the location of the asset and add the vector of the movement and store it in a vector2. then ask if you can draw a line between the asset and the edge of the screen. If you can, move the asset by setting it to vector2. Then I realised if a player keeps firing up then the bullets will keep being processed when there off screen and if the player does this it will cause a memory leak until the level is finished. If the player fires off infinite amount of bullets the game would eventually crash. To deal with this issue, if its detected that the asset is off-screen and isn't a player then it will destroy the asset.

After this I further overhauled the movement to the game by completely changing how it was handled. I did this by having 4 boolean variables. The "heading" variables such as headingLeft. On key press it will set the variable to true for each corresponding key and on key release it will set it to false. Every game cycle of the main loop, it calls the "movePlayer" function witch moves the player accordingly to each of the 4 variables. This makes for a much smoother less stuttering system.

In the "movePlayer" function I move the player then test to see if the players bounding box hits a wall and if it does I move the player back. This process is done four times for each of the directions. This allows for wall collision. As all of this is done before the screen refreshes it means that the player will not see a visual jerky movement.

I added a destroy function for most types of assets. When called it will destroy all assets of the specified type. I also added an overall destroyLevel function that will call each of the destroy assets functions.

I placed my main game loop inside another loop that will make it so when the level count is higher than the number of levels in the game it will end. If it's not in calls the InitLevel function. This function is where all the level data is stored within a switch statement. Each level is stored in one of the cases. When I first started this these functions were really bulky so I made a makeWall, makeAlien and makeCoin functions that would allow you to put the coordinates of the items location.

In the game loop I had it go through each coin and if the player collides with the coin, it then adds 1 to the level count. And I do this again for every plane, if it hits the plane I set isAlive to false which closes the application and prints out game over and the players score to the console. And I have a 3rd cycle that goes through every bullet for every plane and if they collide then they both get destroyed. Furthermore, this will add 200 to the players score. Once every level is completed it will breake out the loop and print you win to the console and the players score.