Version 1 updates & logic:

A Themis Game updates, renders, and implements handlers on the following at a standard interval (current defined at 60 fps) :

Main Game Camera

* + Characters:
    - Direction (for drawing sprite purposes)
    - x,y position
    - Sprite texture
    - name
      * Player is a character with:
        + Bag
        + Array of Objectives
      * Array of NPCs, each having:
        + boolean onMap – used to iterate across the Array quickly
        + boolean condition: decides whether an interaction with this NPC will trigger an Event
  + Tiled Map
    - Drawn in Tile
    - If using Tile to insert NPCs, then an NPC's position can be set using Tile (this is very convenient)
  + Any buttons, dialogue, not directly located or accessibly from the HUD

HUD Camera

* Contains:
  + Backpack
    - has an icon, x and y position (this stays constant on the HUD cam)
    - contains an array of items which are loaded from a textfile
      * each item either has an icon or string denoting it
      * icon, description and quantity are all optional
      * usable items also optional – not currently included in code
    - Methods add() and remove() which adds/removes an item from the bag
  + Objectives
    - contains an array of objectives loaded from a text file
    - checks if an event is no longer valid, if not valid remove from textfile.
      * This means that it has been completed
  + UDSIS
  + Other information that we may want to display, such as username, gametime, mostly optionals

A Themis game uses the following Handlers:

* TouchInputHandler
  + Checks whether the user has touched the screen, and also saves the x and y position of the last screen touch
    - is used by pretty much every class
  + CharacterHandler
    - Deals with character interactions with NPC's and other ingame objects
    - MainCamera
      * Basically an orthographic camera that is bounded by an x and y specified by us. Required because we don't want to go past our map/image
  + AnimationHandler
    - Needed for handling animation with sprites
    - not really difficult to implement just tedious therefore optional
  + GameStateHandler
    - handles gamestate changes based on whether certain conditions have been fulfilled
    - implemented using a stack

A Themis Game has currently two GameStates;

* A Play state
  + Contains the bulk of our game, loads everything
  + If the player has just started a new game, use CreatePlayer() to create a defaut player
  + Otherwise, create the player using stats written out in our textfile
* A Menu State
  + Just contains a play button right now, can possibly add more if we add more Gamestates

Themis Game Events:

* Haven't done much here because I am not really sure on what we are doing for game events.
* Would definitely need a Game Event handler if we want to write scripts such as making an NPC move in a certain direction for x frames, and then another direction for x frames
* Right now there are just two Events defined:
  + RandomEvents
    - A random quiz question that appears on some randomly generated interval of time, can easily be implemented without too much cost by loading from a Questions textfile
  + ScriptedEvents
    - Events that make up the core of the game.
      * Such as meet your advisor, and then talk to your science professor.
      * Get a new ID card
      * etc etc

methods that currently need to be updated for the prototype:

Player->handleInput()

* I've written some psuedcode just need to decide where on the screen the player needs to click to make him move in a certain direction. I.e we determine regions on cam for up left down right, or we create actual left right up down buttons
* Also need to check whether the players position is reaching the map bound or an impassable object

Buttons->handleInput()

* Same as above, had trouble finding the right logic for determining whether the cursor was within the button area...

Play->CreatePlayer()

* just need to create the default player for now since we dont need to implement I/O yet

The update, render, and handle input methods for HUD class

Creating a TileMap

Things to start thinking about:

start filling our assets folder with sprites, backgrounds, button images, sound effects, music, everything we might think of wanting to use in our game.

Write a script for how events will occur and in what order

Also implementing keyboard input would be really convenient for testing on the computer and isn't very hard, just annoying

Adding more gamestates to the game, such as making back pack, objectives, and UDSIS different states (like zelda). This is easier than drawing them on top of the play state since we can just preserve that.