“Mild Goose Chase” Video Game, Version Log

**Kyle, John, Timothy, Victor, Chris, Matt, Frank**

Version 1.0 (4/6/2020)

* World Generation Algorithm implemented
  + Will create a randomize level upon clicking “Play / New Game”
  + Uses dummy prefabs
* Basic Player Sprite created
  + Implemented walking animation
  + Implemented attack animation with a sword (press spacebar to attack)
* Basic Player Movement
  + Use the WASD/UpLeftDownRight keys to move up, left, down, right respectively
* “Fixed” Camera implemented; will follow the player as to always lock them in the center of the screen
* Wall collision implemented (to prevent the player from clipping through the walls)
* UI elements added
  + Pause Menu
    - Resume
    - Options
      * Volume
    - Menu
    - Quit
  + Start Menu
    - New Game
    - Options
    - Quit
* AI pathfinding/sightseeing algorithm implemented
  + \*Please see below for images on how this works

Please see below for visuals and a short demo video on how the AI sightseeing algorithm works:

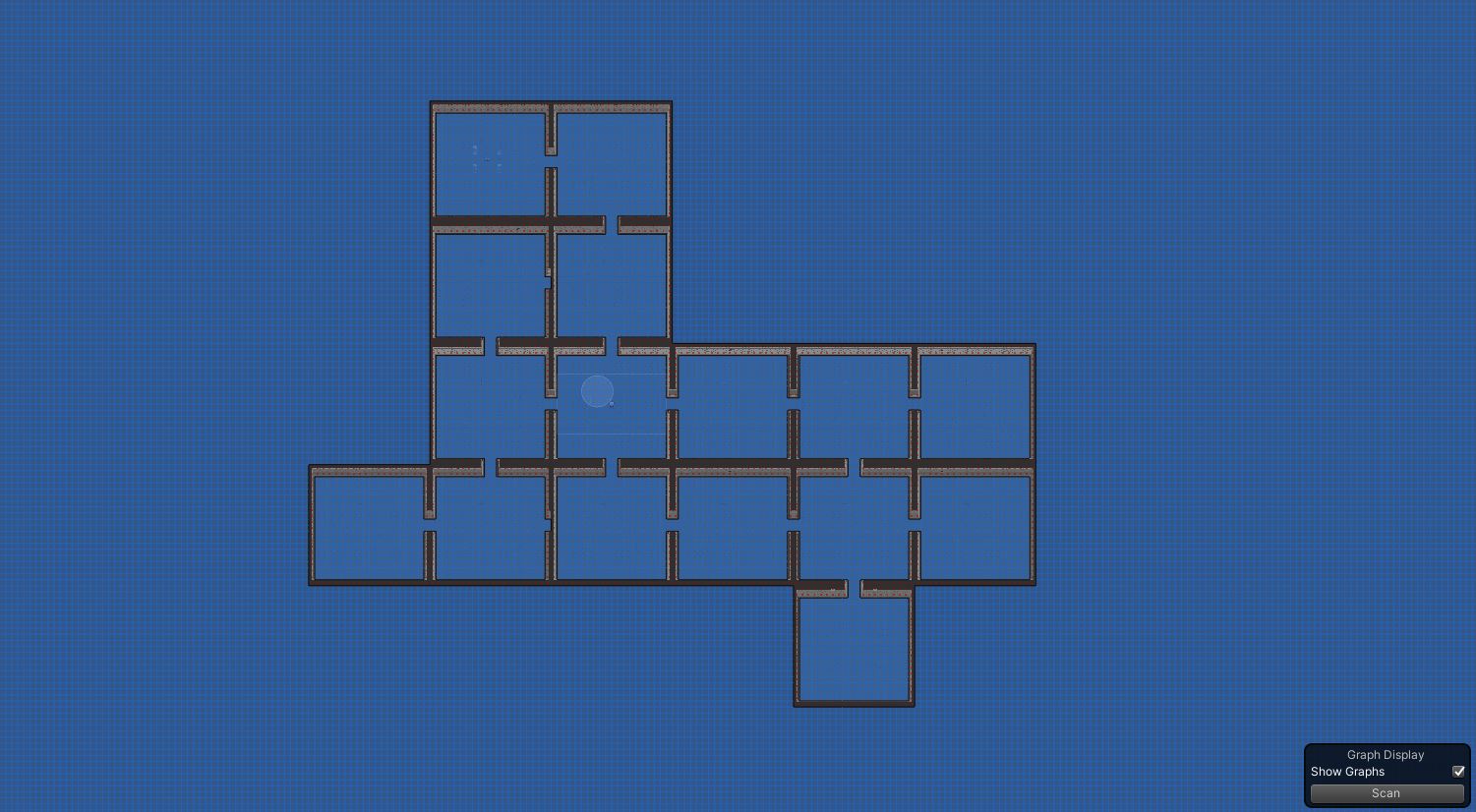
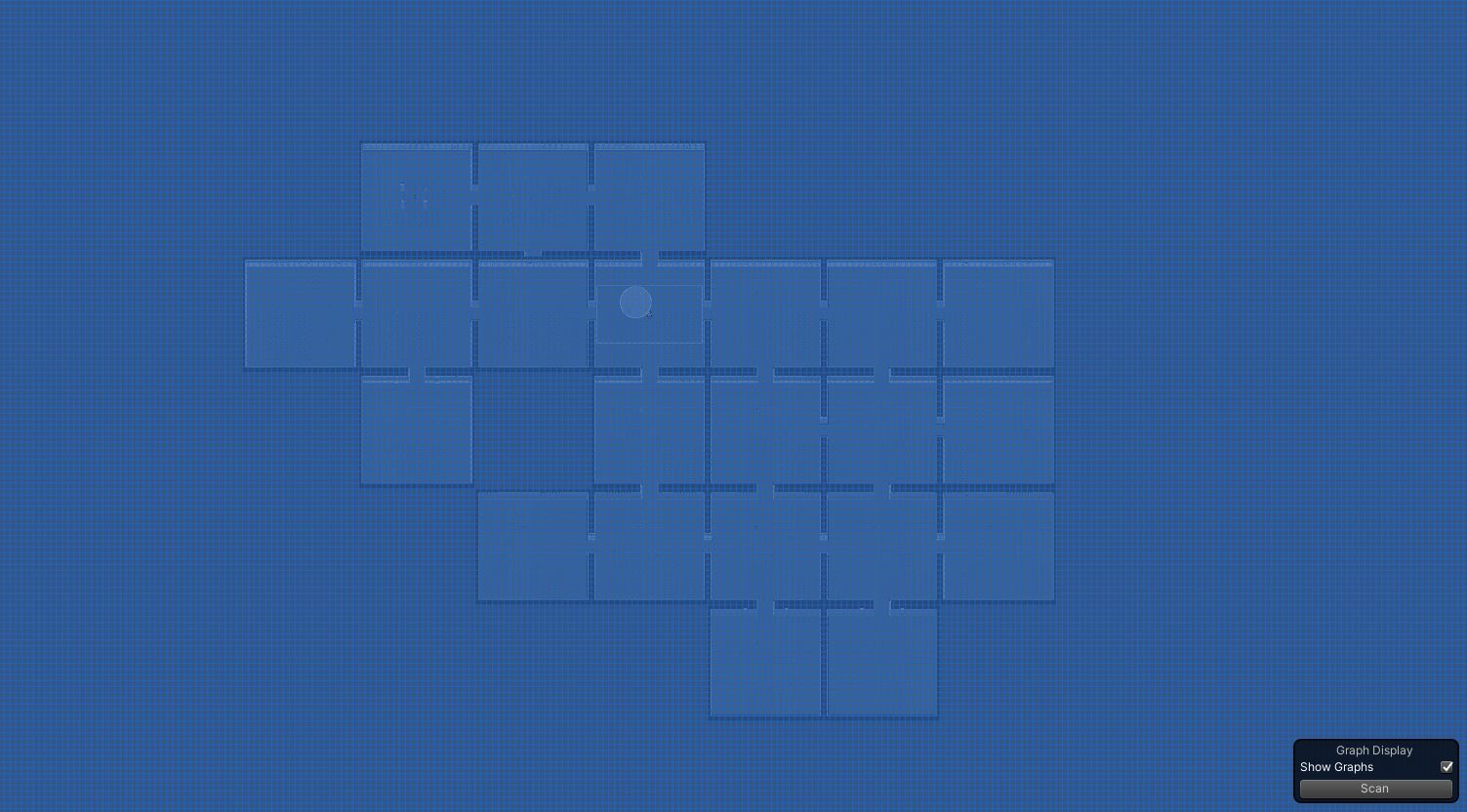
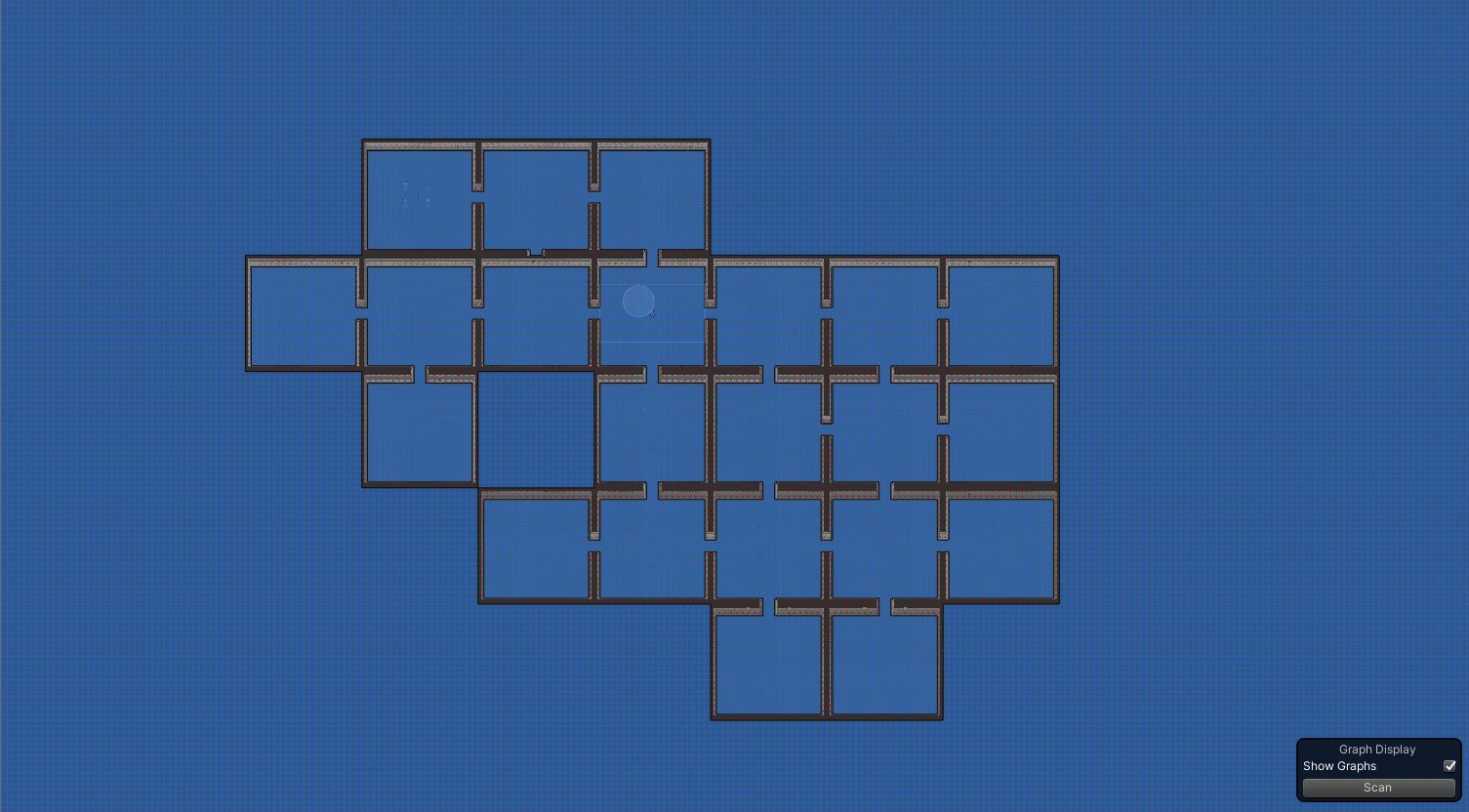


Figure 1: The image above shows a visual for what the AI would see when a map is generated. Dark colors indicate walls. Blue areas indicate traversable areas. Red squares indicate barriers, and the AI cannot go through them (better picture of this below in Figure 5).

Note: The blue areas are outside of the playable area, which technically means the enemies can reach these points; but since these areas are off limits to both the player and the enemy, this will not break the game in any way.

Figure 2: This is a dummy map that has not been scanned yet. Please note the shape of the map and remember that every map generated for this game will be a different shape, with rooms of different sizes, openings, and walls.

 Figure 3: The same dummy map from Figure 2, but the sightseeing algorithm has finished. Notice how all walls, openings, and rooms have been accounted for. Notice how all areas that are traversable by both the player and the enemies can be successfully traversed.

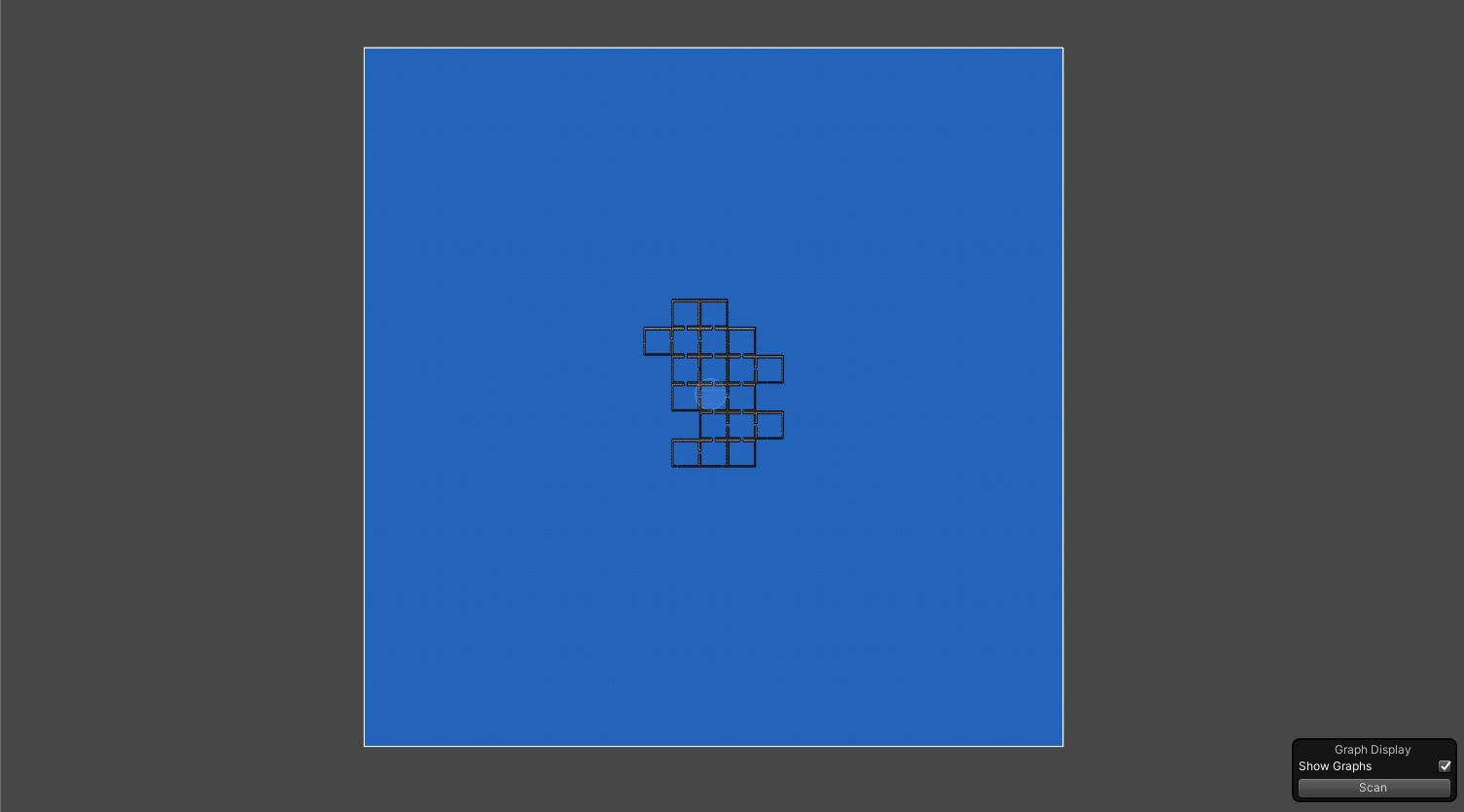
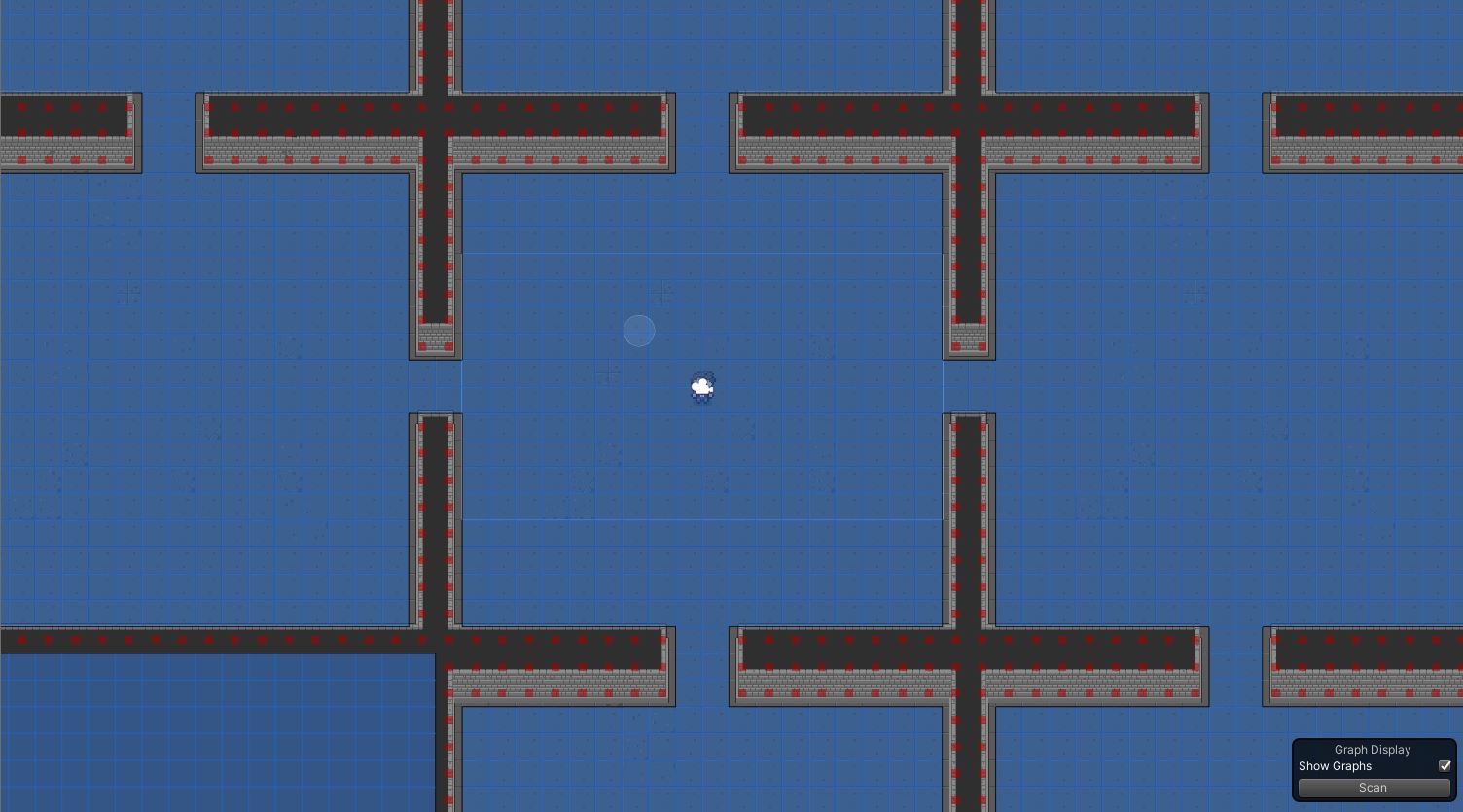


Figure 4: A more zoomed out view of another generated dummy map, post-scan. Notice how this map is different than the other two, and the sightseeing algorithm was able to mark everything successfully.

Figure 5: A zoomed in image of a dummy map, post scan. Here, the red markers, which indicate barriers, are clearly shown. They each mark a single tile, which lets the AI know that these tiles are off limits. These markers cover all walls.

Version 1.1 Update Log – April 13, 2020

* Fixed issues with the UI to have it working properly across all our platforms (scaling issues, missing items/prefabs/sound issues, scene changes, etc.)
* Cleaned up some prefabs for neatness and functionality
* Added an animated damage text object
  + Currently doesn’t do anything but as soon as the combat is implemented properly it should show damage above an enemy on hit. (The scripts are on the master branch currently but won’t work until we know the combat works)
* Got initial combat mechanics (somewhat) working
* Changed the players movement to function as intended and fixed the issue with the player “drifting” after contacting an enemy
* Added new room prefabs to add variety to dungeon rooms
* Added new rooms to appropriate lists in the room templates script to allow rooms to be added to the rotation of randomly spawned rooms
* General updates to Player movement
* General updates to Player attacks
* General updates to combat (will be refined once the enemies die correctly)

Version 1.2 Update Log – April 20, 2020

* Fixed issues with combat from Version 1.1
* Many small and large updates to the UI such as new fonts, animated title, “help” menu on the main menu, and functional volume sliders
* Small UI fixes for a more friendly user experience
* Added music to the game. An extra volume slider controls the background music for the start screen and the main game.
* Separate working volume sliders for music and sound effects
* Added a small animation to the title of the game
* New UI for Health/Stamina bar
* Scripts added for Health/Stamina bar
* Creating an enemy health bar prefab along with scripts and sprites for it
* Implemented dashing mechanic for the player (Binding: Left Shift)
* Added our first item (heal potion) - walk over it to heal
* Reformatted the player control script to inherit off battler script
* Added sprites for swords in case we end up implementing items that can be picked up besides the potions

Version 2.0 Update Log – April 28, 2020

* Game now boots to the main menu upon death so the user can jump right back into a new game
* New enemies added:
  1. Mini Slime (part of the Slime’s desperate attack)
  2. The Butcher
  3. Sparkacus
  4. The Archer Nemesis
  + Added their own unique animations and attack patterns
  + Animations were added for idling, movement, attacks (except slimes, they just run into the player as an attack), and death of enemies
  + Audio added for enemy deaths
  + Added fixed-distance chasing, patrolling, searching and flee states for the enemies
  + Added death sounds for all enemies
  + Added health bars for all enemies
  + Enemies now have a chance to perform a “desperate” attack (attack made when they are low on health)
    - Slimes may split apart into smaller slimes (when slimes reach a specified fraction of their health, they will be spilt into ~1-6 baby slimes)
    - The Archer Nemesis may shoot two arrows at a time
  + Enemies now have a change to run away during an encounter
* There is now an enemy kill count for a level; several enemies need to be defeated before advancing to the next level
  + Transitioning to the next level of the game now works properly
* The player and enemies now regenerate health and stamina based on their attributes
* Enemy attributes evolve upon entering the next level
* Adjusted enemy sprite layers (depending on the player's position, enemy appearance now correctly shows being in front of or behind the player)
* Added more room prefabs to the rotation
* Reformatted player and slime hurt boxes for more accurate damage output/input tracking
* Refactored the slime script to be more accurate in attacking
* Removed conflicting audio listener (sounds effects overlapping and/or not working as intended)
* Added a menu to show the player's stats.
  + Also was added in the help menu on the main screen, controls for pause and for opening the skills menu (currently "K")
* General Player and Enemy balancing
* \*\*\*Plus, many backend changes made for readability/optimization purposes