Homework 5 Reflection

When implementing the script manager, I took a lot of the example code and reused and changed it based on what I needed for my engine uses and game. First off I allowed the ScriptManager to follow the design of my other Managers in my engine where it has a static get() function where the users are able to get the instance of the script manager anywhere in the engine. Then all of my scripts and functions were created and added in main to be run elsewhere in my code.

My logic for my character aka the gravity functionality was delegated to the character_logic.js script. This required me to expose my character's velocity to the scripts. I also had to expose some functions to check if the character was colliding with something, then the logic was made and changed the character's velocity as needed, then I exposed a move function to the scripts that took in a character and then just used the velocity which we had just messed with to move the character. Then I created a script to handle the pause events which required me to expose the paused boolean located in the Timeline, and the handling merely required me to change the paused boolean to the opposite value that it was currently at, then the Timeline handles the rest of it. Finally, I created a simple script to raise an event such as the pause event in the scripts.

Then when implementing the chord, I kept track of the time since either the esc key or the "p" key was pressed in, and then after that time has passed it checks if the other key had also been pressed since then or if no other keys have been pressed. Then based on that information it sends the appropriate event to the server so it can handle it accordingly. For the new chord it simply changes that client's character's fill color to a random color.

Finally, for my 2nd game I created a level from space invaders. The character now just has limited movement in that it can only move left and right with hidden "platforms" at the edges of the screen with the engine's collision detection to not allow the character to leave the screen. The character is able to shoot by creating a bullet (moving DeathBox) that takes in a boolean as to whether it was player created or not (which is important as enemies can also shoot bullets) as this decides how this bullet moves (up or down) and collides with other objects. Then I created a new Enemy class that merely has a set movement pattern that only happens every some amount of time. Once that time passes, I use a random number generator modded by the number of enemies and if that number is equal to one of the enemies then that enemy will shoot. The movement of the enemies is the regular down, left, down, right, continued. Finally I created events for shooting the bullet, which just creates the DeathBox with the boolean corresponding to if it was created by a player or not; the event for the bullet colliding with an enemy (and modified the one for the character) to delete the bullet as well as the object it collided with (given that the bullet is shot from an enemy and hits a player or vice versa) from the game world; and an event for deleting the bullet from the game world as soon as it leaves the window/play-area of the game. This creates a slightly dumbed down level of space invaders that feels pretty good (I think), with the win condition being that you've shot down all the enemies and now get to travel deep and empty space or your character ship gets shot and the game seg faults. :)