Overview:

All code in Player.cpp pertains to:

1. Player movement
2. Player shooting and mechanism to change weapons
3. Player health and what happens should the player run out of health and die
4. Player initialization behavior at start of game and at each respawn after dying

Types of Classes:

1. **Player:** the main class for the player
2. **Explosion**: A class that serves as a subclass of the player class. Used for to control explosion behavior after playe death (Same as lecture shooting.cpp class)
3. **MissileStandard, MissileB etc**: Example missile classes for player to use (Same as lecture shooting.cpp class)

Player Class:

Member variables:

- X,y: location of the player
- Health: current health of player. Full health is 5, once this health is depleted to 0, player dies and once spawning is finished, health restored to 5
- State: 0 means player is either dead or currently respawning. 1 means player is active and be hit by enemy targets
- Respawn: boolean used to determine if a new player is to be respawned at the defined starting location
- Spawning: boolean used to determine if player is actively spawning or not
- E: explosion object of the player. Used to draw explosion when player dies
- Missile type: string used to determine which missile is shot when shoot() member function is called
- Initialize Count: Counter variable to determine how long player is invincible for during spawning

Member functions:

- Shoot(): Checks the missile type, and triggers a missile of that type to be launched
- Draw(): Draws player based on x and y coordinate
- ChangeMissileType(string newType): Changes missile type
- Initialize(): Checks if player is spawning, if they are, will draw ship from bottom of screen upwards to a define point. Ship will be invincible this whole time as denoted by its force field. Once ship waits a specified InitializeCount, the ships health is set at 5, state set to 0, and is now active.

Missile Standard and MissileB Class:

Both Missile Standard and Missile B from same class in shooting.cpp. The only difference is that missile B shoots in black and missile standard shoot in red

Explosion:

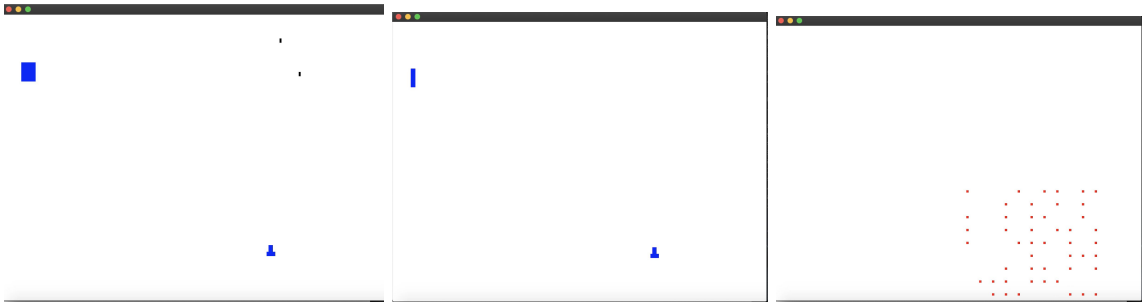
Same class as in shooting.cpp. However, only used as a member variable within the player class rather than an independent object

Debugging code:

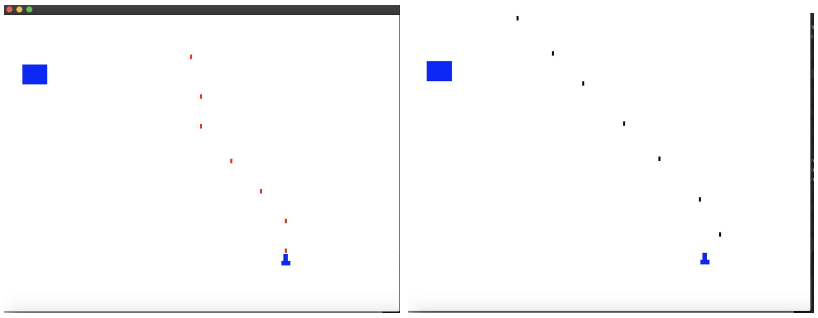
The functionality of the player class is tested in the main function which displays a visualization of all functionality.

Player movement is tested using a key matched switch case mechanism as done in class

Player health: Displayed by blue rectangle in top left corner of window. Simulated a hit from an enemy by hitting the “H” key. After 5 hits, health hits 0, player explosion begins.



Switching Missiles/ Shooting: In switch key mechanism, player can shoot by space bar, and switch to different missiles using “S” key to switch to missile Standard and “B” key to switch to missile B. To shoot, space bar key is used.



Player Initialization/respawn: player.Initialize is called every time step. At the end of an explosion, player respawn and spawning are set to true. This triggers the initialization function to draw the plate initialization behavior. Once initialization is finished, player respawn and spawning is set to false