Devin Murray
https://github.com/DevMur/Smoko

My project is a galaga style rpg. It will star a star fox like character as they rescue their friend from a computer ship. The game will start as an endless avoiding asteroids through space. Whether it be time or score there will be a mid-way boss fight followed by more vertical scrolling with some enemies implemented. Once another score, time, or other trigger is reached there will be a boss fight followed by more scrolling. After a third trigger the player will have either continue to reach a high score or they will fight the kidnapper followed by an endless mode. The final boss is only accessible if other game requirements are met... treasure, achievements, easter eggs, etc.

I plan on using C++ and most likely QT. The libraries include SDL which is a media/devices component to allow direct control of sprites through, in this case, wasd or arrow keys. The windows library will also be used to add music. Microsoft paint will be used for the sprites, enemies, and player ship.

My project will not work if it does not have a gui

Player input (controls of the character)

A board/map to play on

Enemies or obstacles to avoid (a game must have an objective)

A game cannot be played without controls and I do not have a backup plan if concurrent controls fail. I can take user input in a while loop but when events arise the delay on player input may render the game unenjoyable and potentially unplayable.

The map can in theory be a blank space but I plan for a black background with a layer on top of white red blue and green pixels moving slowly beneath the player to give a sense of direction.

Without a story and challenge there is no game. Just some code executing an arguably rough video and story. Incrementing through a 2d array and having vertical positions can add obstacles but some enemy movement based off the players position and speed would be more realistic for a game.

I do not need any outside data but I will need wav files as well as a boss fight theme. My boss fight theme is currently being composed regardless of its use for a performance in Macky auditorium and is based loosely on Williams “Duel of the Fates”. Laser and ship sounds will be made with my own synthesizer. High scores will be recorded in a text file.

The classes include player, ship, space, enemies, boss, achievements, and asteroids.

Player will maintain the number of lives left, the score, potentially the name (if high score is achieved) as well as ship parts/achievements for the ship class to inherit upon new lives or new checkpoints in the game reached (not gamestate checkpoints)

Ship will contain the ship sprite, it will inherit the ship’s state from the player, it will also contain the movement methods and outputs from movement input.

Space will contain the map in which the player resides. It will largely make up a background but its state will change slightly as a player begins to speed up as the game progresses or become static when boss fights occur. Its other job is to maintain time and call other classes member functions to display pertinent information like ship location and score. Its state will impact obstacles and enemy spawns.

Enemies will each have their own class and actions as will obstacles.

I will call obstacles the easiest, they have a linear direction and move at a linear speed with a linearly increasing spawn rate.

Medium strength will carry some set of predetermined move sets chosen at pseudo random with a similar concept to easy obstacles. They will also carry score to report to a player

Hard will make calls to find the players current or predicted location (based off the current location) and make more of an attempt at a players life. Carry a higher score

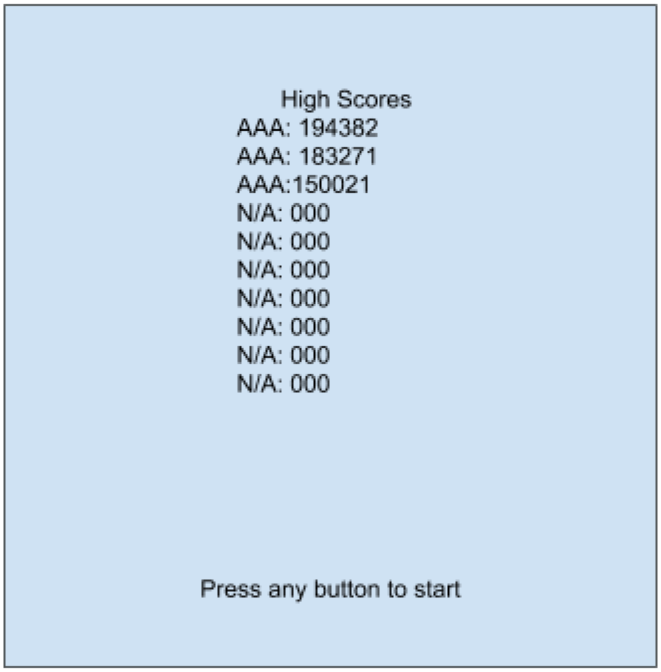
Achievements/treasure will be dropped in predetermined and pseudo random locations with certain hidden checkpoints the edge of these drops (if they don't spawn prior they must spawn at this point). They carry points and keys to more/different content.

Bosses are method heavy looking to beat the player with a set of moves, the moves may be called at random or the computer may lean heavily on a certain set once testing has been done.

Ship will inherit information from the player as will the bosses. When a new live is used it will receive any treasures points and velocity of the previous ship, which upon dying, will update the player of its final states so as to give the new instance of ship the proper attributes.

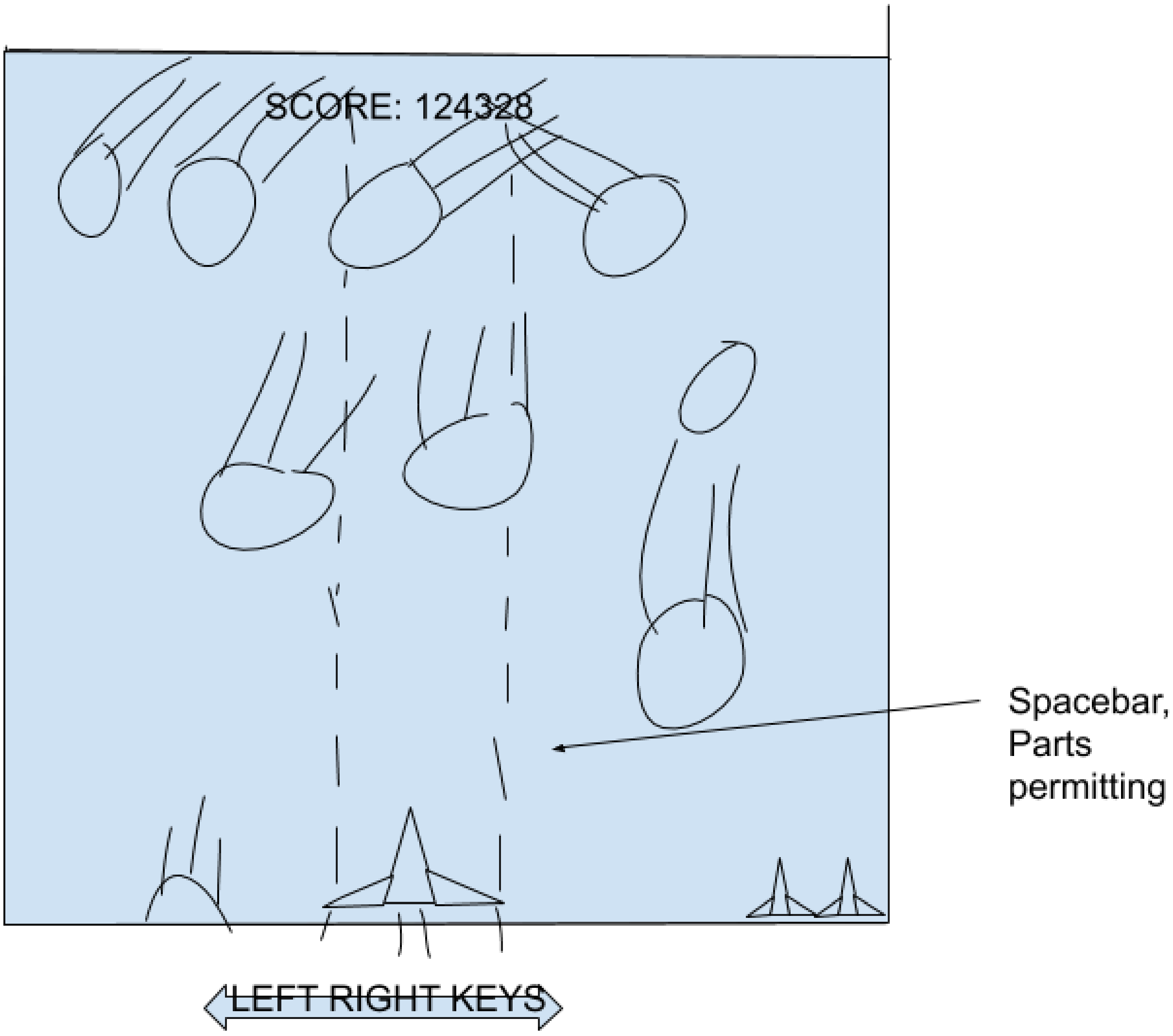
Singleton actually makes a lot of sense in this case. Since there can only be one ship at a time and it is inherited from the current player there is no reason to use multiple instances.

Prototyping will certainly be used as well. For each enemy and obstacle there does not need to be a new instance of it each time a trigger is tripped. Instead they can all be instantiated upon the game start and thrown at the player in different quantities. Especially since each enemy type is identical to each enemy of the same type.



High Scores
AAA: 194382
AAA: 183271
AAA:150021
N/A: 000
N/A: 000
N/A: 000
N/A: 000
N/A: 000
N/A: 000
N/A: 000

Press any button to start



submission 1: GUI and player input, the player will be able to run a page and input their information. This call to information won’t be used unless they set a high score but should allow

for testing of a key class. There will be little to no graphical assignment besides text and a custom window. For this submission, I will need to learn how to open a new window and take user input in that window. As for the rest of the class methods and variables, it should be nothing I have not seen or not used before.
Submission 2: The board will be created as well as player sprites and obstacles. They should be able to move horizontally, die, and compile a score. There will be no crash animations, no sounds, nor moving background. A player should be able to play a bare bones version of the game. I will need to learn concurrent player input but once movement is learned object interactions are well within my scope. I have little experience with it but I will need to interact with a time management function as time will impact spawn rates of many objects during this patch and in later patches.
Submission 3: Bosses, enemies, and items will be added. I will need to understand cloning and develop some algorithm(s)/method(s) to determine how ruthlessly the computer will use the enemies to defeat you. In-depth playtesting will be done and a visually crude space game similar to a mix between xevious and galaga.
Submission 4: Music will be added. Boss fights will be refined. Sound effects will be added. Key animations will be added. I have no experience implementing animation via c++ but I have used after effects and Microsoft paint for rough and clean animations. Worst case scenario and i can output multiple images in order upon an explosion or death (this may be the least computationally taxing method). I will need to learn to manipulate wav files but they should just be queued for events. Firing a missile, getting fired at, a boss fight, a death, the title screen, the endgame
Submission 5: RPG elements and a moving background. To avoid breaking a working game I will keep space static until the final submission. It should move completely independent of all other actors. I believe waiting until the end to move the space makes sense because it is easier to move sprites across a static coordinate plane rather than a moving one. I don’t want sprites sliding to new positions as the background moves but the background will also move at variable speeds. My current idea is to change their position in a similar manner as you would move something in a 2d array. If the elements of this array are changing at a rate faster than a sprite can move a single position either a new implementation of the display must be used or the background needs to move less variably. I would like to apply QT or unity type graphics to create a more interactive and engaging story. While text elements are a must a voice over or two may engage the player more. In this submission, I would need to learn 2d animation, it could be as simple as a picture with a moving text box, or multiple panels of story events. This will be time constricted but will separate a class project from a production-grade game.

I have used c++ on and off for the past 2 years but will not pretend to be above competent. Every lecture exercise and activity will be used as hands-on practice. I understand the objectives of QT and unity but i don’t have experience with either. I have already found numerous tutorials on youtube and google which can be used as introductions while class time can be used as another hands-on test of my own learning and my project can be my custom implementation.

My deadlines seem good, my classes contain most if not all of the required basic elements required for the game. Revision to the class structure and inheritance may be edited once gameplay is implemented. Between starting my project and now I believe the only factor that I overlooked or underestimated will be implementing movement to enemies and their weapons. Boss attack patterns can be very predetermined but having a player fire and having an enemy makes a move while otherworldly factors potentially impacting them may be difficult. Moving them around in a non-obviuos pattern not dictated by a grid may be difficult. My plan, if I cannot find a method to move them in a new and unique manner, will be to make a massive grid where each sprite takes up dozens of spaces and their center being moved vertically, horizontally, or diagonally will be minute compared to the size of the board. My only other revision is my use of SDL. I believed it would be plug and play but I will need to spend more than an hour studying it to implement it in a usable manner.