Jeffard's Dungeon Adventure
A program proposal by Jason Tran

Project Description
For my culminating project, I plan to make a top down dungeon battle game. Top down in this context means that the player is viewed from above in an overhead view. When the controllable main character moves off of the screen, the next part of the map appears in that direction. Progress through the dungeon is made through the general direction of right and downwards.

When the program is run, you are taken to a menu screen that displays 3 options; Start, Program instructions and Controls, End program. The setting of the game is an underground dungeon that is guarded by monsters and spikes that are trying to kill you. The objective of the game is to get to the end of the dungeon while earning items and buffs along the way to make you stronger.

The character is controlled with the wasd keys to move in 4 directions. The character only moves while the keys are being held down. The character has 3 statistics; health, damage and movement speed. All of these start at a base number but can be improved through collecting items throughout the game. Items are found from predetermined chests throughout the map.

When you press the K key (placement on the keyboard is convenient for ergonomic gameplay), the character swings his sword dealing a certain amount of damage, the sword swing has a cooldown. The character also has an ability that you can activate every so often (again with a cooldown) by pressing the L key. This ability does a considerable amount more damage than your regular attacks. Attacks between enemy and player are only landed when the enemy is in appropriate range (predetermined value).

There are a few types of enemies all of which have different health, damage and speed statistics. They are generated based on the part of the dungeon you are on; at the start of the dungeon are the weaker enemies and at the end of the dungeon are the strong enemies. There are mini bosses along the way that give you unique items once you defeat them. At the very end of the dungeon, there is a boss that has high statistics and depending on how good your own statistics are, you may have an easy or hard time fighting the boss.

Enemies will become locked on to the player when the player enters a certain radius from the center of the enemy and wont stop chasing until they defeat the player or are defeated. It is possible to have multiple enemies locked on to a player at once. The enemy attacks have a cooldown as well.

If you beat the boss or you are defeated by an enemy, you are taken to a game over screen for a few seconds and then back to the main menu.

Programming Concepts

- Loops (while, for, do while): This will be very important in my program. Animation and constant gameplay will require loops. I will need loops to repeat segments of code either a certain number of times or until a condition is met. For example, the gameplay code being inside of the menu loop because when the game is over, you return to the main menu loop.
- Random Numbers: This will be utilized very heavily in my program. Random numbers determine how many enemies spawn and which enemies spawn as well as item drops.
- Arrays: This will allow me to simplify variables and make them easier to access because there will be many variables of the same data type.
- Hsa_ufa: This will allow me to access the console and make my game visual.
- Console Graphics: This will allow me to illustrate my game. The draw commands will allow me to make obstacles, walls, and the setting of the game. The image insertion commands will allow me to import my sprites of the different characters, items and graphics that I want.
- Animation loops: These are important because the illusion of animation is just many images being drawn and erased in the console so if all of the animations are in a single loop, they can all run concurrently and the game will run as intended.
- Synchronize function: Important to remove flashing of animation.
- Return and void methods: These will allow me to organize my program by invoking large amounts of complex code in a single line. A return method I can use is a method for checking collision between objects, a void method I can use is a random number generator.
- Character key inputs: Used for gameplay mechanics of the player; movement, attack, interaction.

- Math functions: Collision detection between enemies radius of player, collision between enemy and player, collision detection for appropriate attack range.
- RGB numbers: Used for creating custom colours to fit the colour palette of the project.
- Mouse Tracking Functions: Used to select options in the main menu.
- Music and sound functions: Used to implement appropriate sound effects and music.

Must Have vs Like to Have

Must Have	Like to Have
A main menu with the game title, program designer's name, option to start new game, instructions and controls page, and option to end game	Background music in the menu, background music in the dungeon
Dungeon background setting	Special ability attack being activated by pressing the L key
Map that is large and expands whenever the player moves outside of the intended bounds => the screen going to the next part of the map	Possibly figure out a moving camera to follow the player
Enemies that follow and attack player when player enters their range	Ideally only moving up, down, left and right
Character statistics: Health, speed, damage	
Sprites of player, enemies, items	
A visible health bar	
Attacking and being damaged mechanic	Attacking and being damaged sound effects
Working movement in all 4 directions	
Randomization of enemies spawning locations	A mix of stronger enemies where they are less common (at the start) but not too many. Just to make the gameplay dynamic

Item upgrades that improve either your health, damage and speed statistics	

Research:

1. I will need to find out how to change the screen to the next part of the map and keep my player in their rightful position on the overall map.
2. I will need to find out how to make the enemy mobs follow the player (ideally while moving only up, down, left and right).
3. I will need to find out how to implement sound files into my program.
4. I will need to find out how to make the character damage an enemy or be damaged by an enemy and how to update both of their health bars.
5. I will need to find out how to identify items that the user can use. I will also need to find out how to drop items, and how to pick them up.

My game will look something like this: