

## **Requirements**

Your goal is to make a 2D game that has some sort of purpose/story. Your game can be any genre like RPG, puzzle, platformer, fighting, etc. (school-appropriate!) The game should run, allow the user to play toward some end goal, and feature a means of quitting the game. You are allowed to use other libraries (with permission first) but most of your game should reflect what we learned in class.

(This project is very open-ended, so you can decide on your game and story)

- 1) Main character
  - a) Controls, Movement, Interactions
- 2) There must be at least 3 moving shapes on the GUI. They could move by the user(s) or on their own. (NPCS/Monsters/Items)
- 3) Standard keyboard/mouse controls
- 4) Each sprite (shape) should interact with at least one other shape. (Collision/Object Placement)
- 5) There must be at least 2 sets of text written on the GUI.
- 6) Your game must have a winning condition or a losing condition or both. (can't be infinite)  
This can be anything from:
  - a) running out of time
  - b) being slain by monsters
  - c) collecting all the items
  - d) saving royalty
- 7) Your project should be properly indented and contains comments!!!
- 8) Zip your game files and upload the zip file onto Google Classroom.  
Please see HOW TO ZIP.doc for detailed instructions on how to zip your file if you don't know how to do so.

## **Grading:**

### **Part 1:** Planning (20 points)

You should first plan your project using the draft provided. The diagram (part 4) should show or explain how each image is going to move. You can draw your diagrams on a drawing application or on paper. If you make your diagram on paper, please submit a CLEAR picture.

### **Part 2:** Game (70 points)

Accomplishing the requirements will earn you 70 points.

### **Part 3:** Creativity (10 points)

The rest of the points will be determined by how difficult/creative your game is.

## Grading Rubric

Critical Elements	Exemplary	Proficient	Needs Improvement	Not Evident	Points
Planning	Plan is completed fully. Meets proficient criteria, developed a clear, well organized, and detailed plan	Plan is mostly completed and explains detailed points about the idea of the game.	Plan is filled out but not complete or does not explain points of the game	Plan is incomplete or missing	20
Creativity	Game has a clear goal and theme. The game has some level of challenge.	Game has a mostly clear goal, though some aspects may be out of place. The game may or may not have some level of challenge.	Game has a few consistent elements, but is mostly a collection of random objects.	Game is either incomplete or has only a few disconnected elements.	10
GUI: Interactions and controls	Your game includes multiple different interactions between sprites, and it responds to multiple types of user input (e.g. different arrow keys).	Your game includes at least one type of sprite interaction and it responds to user input.	Your game responds to user input through a conditional.	Your game includes no conditionals.	20
GUI: Movement	Complex movement such as camera movement, speed manipulation, moving in a curve, or jumping is included in multiple places in your program.	Your program includes some complex movements, such as jumping, acceleration, or moving in a curve.	Your program includes simple independent movement, such as a straight line or rotation.	There is no movement in your program, other than direct user control.	20
GUI: Win/Lose condition	Implements functional winning/losing (sentinel or flag values)	Implements functional exit conditionals with one or more errors	Implements functions exit conditionals with multiple errors	Does not implement exit conditionals	10

	conditions to exit the game				
Complexity	Exceeds typical technical complexity for course level	Meets typical technical complexity for course level	Meets typical technical complete but are simple	Below typical technical complexity for course level	10
Coding Convention	Your program code effectively uses whitespace, good naming conventions, indentation and comments to make the code easily readable.	Your program code makes use of whitespace, indentation, and comments.	Your program code has few comments and does not consistently use formatting such as whitespace and indentation.	Your program code does not contain comments and is difficult to read.	10