

Object Oriented Programming Spring 2020
Semester Project



Course Instructors

**Dr. M. Aleem, Dr. Naveed Ahmed, Dr. Kifayat Ullah, Mr. Hassan Mustafa,
Mr . Jawad Hassan, Mr. Hassan Raza**

Design Deadline (Class Diagram): May 5th , 2020 before 17h30

Deadline Implementation: May 15 th , 2020 before 23h30

**Can also be submitted "late" till the start of the final exams without any
penalty**

Instructions

- Make sure that you read and understand each and every instruction. If you have any questions or comments, you are encouraged to discuss your problems with your colleagues (and instructors) on Piazza.
- **Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge) both of you will be punished. You will be awarded (straight zero in the project — which can eventually result in your failure) and appropriate action as recommended by the Disciplinary Committee (DC can even award a straight F in the subject) will be taken.**
- **Try to understand and do the project yourself even if you are not able to complete the project. Note that you will be mainly awarded on your effort not on the basis whether you have completed the project or not.**
- Divide and conquer: since you have many days so you are recommended to divide the complete task in manageable subtasks. We recommend to complete the drawing and design (i.e. number of classes and their relationships) phase as quickly as possible and then focus on the intelligence phase.
- Before writing even one line of code, you must design your final project. This process will require you to break down and outline your program into classes, design your data structure(s), clarify the major functionality of your program, and pseudo-code important methods. After designing your program, you will find that writing the program is a much simpler process.
- **No Marks will be given if you do not submit your class diagram and if you do not use the object oriented design principles you have learned during the course.**
- Imagination Powers: Use your imaginative powers to make this as interesting and appealing as you can think of. An excellent solution can get you bonus marks.
- Combine all your work in one .zip file.
- Name the .zip file as ROLL-NUM SECTION.zip (e.g. 19i-0001 B.zip).
- Submit the .zip file on Google Classroom.
- You must follow the submission instructions to the letter, as failing to do so will get you a zero in the Project.

The objective of project is to make a game in which Player swap one gem with an adjacent gem with the help of mouse(Drag & Drop) to form only horizontal or vertical chain of three or more gems of the same color, (an idea of game can be taken from the screenshot given at end). Bonus points are given to player when chains of more than three identical gems are formed, but also when two chains are formed in one swap. When chains are formed, the gems disappear, and gems fall from the top to fill in gaps. Sometimes chain reactions (referred to as cascades) are triggered, where chains are formed by the falling gems. Cascades are awarded with bonus points. Be creative as you can in drawing gem and game board.

Following features should be added to the game:

- 1) Take Name of player as input.
- 2) Drawing of game board in the start of game
- 3) You can match Rows and columns (No diagonal matches allowed)
- 4) There should be a Hint button, when the player clicks hint button, highlight the gems that can form a chain and the score & progress bar should be depleted.
- 5) Progress Bar should be displayed.
- 6) Next Level will require specific amount of points/coins.
- 7) A menu also be included optioning Levels, Level Modes, Highest scores and Settings etc.
 - a) Levels include List of levels (completed levels should be displayed bright and incomplete levels be dark)
 - b) Level Modes include difficulty of level (at least two modes)
 - i) **Normal Mode**

In Normal Mode, the player fills up the progress bar on the bottom of the screen by matching gems. The game starts with an empty progress bar and the game will not ends in between level in this mode. When the progress bar is filled up completely, the player goes to the next level. As the level progresses, more points are required to proceed to the next level. As the player levels up, they get more points by matching gems (example: Level 1= 100 pts., Level 2= 150 pts. etc.).
 - ii) **Time Trial Mode**

In this mode, the gameplay mechanics are like Normal Mode, but the progress bar starts half-way filled. The player must keep the bar filled by matching gems, and they will level up by filling the progress bar. The game ends if the progress bar is depleted completely.
- Note: Game difficulty will increase (means more points will be require for completing current level), with the level increase.
- c) Highest Scores include top 4 highest scores, the player has achieved till then
- d) Settings will include Music and sound option, Player profile and How to Play options etc.
- 8) Game must include Play and Pause Button.

- 9) There should be at least 5 types of gems (max 8 gems), each gem will carry equal points. When four gems chain are formed, player gets bonus triple points (x3). When Player forms five gems, points will be added 5 times more (x5).
- 10) Game Control will be Mouse only. While Play/Pause, Hint, Music On/Off, How to play, profile Buttons can be control using some keyboard keys (for example, H key for Hint)

File Handling

- 1) Player can resume game (after leaving midway). You need to store name and his current state of game
- 2) High scores will be stored in files with names.
- 3) Player profile will also be stored in files. In player profile you need to store the name and his highest level along with the game state of his highest level.

Marks Deduction

- 1) If more than 1 Global Variable has been used in coding (Maximum 1 variable allowed).
- 2) If code has been written without classes.
- 3) If coder has not applied Programming Practice.

Bonus Points

Bonus marks will be given based on

- 1) Creativity on Game Board.
- 2) Graphics of Game.
- 3) lives of players to clear game board/level.
- 4) Game Aesthetics.
- 5) Player Profile (can include Name of Player, number of levels completed both in normal and time trial mode, Language, completion rate etc)

Some guidelines from the given code:

In the skeleton code you are given game.cpp, util.cpp, util.h, install-libraries.sh and Cimg.h files

In game.cpp

You have been given “*MouseClicked*” function in which you can code for right click or left click.

Also a function named as “*MouseMoved*” which is called when the mouse cursor is moved and the coordinates of mouse will be stored in x and y variables.

But you need to drag the gems so another function named “*MousePressedAndMoved*” is given, which you can use and build your logic according to the functionality.

For game execution:

First you have to install libraries by opening terminal in project folder and run a command “ bash install-libraries.sh”

then run “make” on terminal and it will create object files and game as out file.

To run the game write “./game” in terminal.

