

CS 2340 A/GR Summer 2023 Project Outline

Introduction

One of the primary objectives of this course is to teach you some best practices for building large software systems. In order to do this, you must work as a team to complete a project in one semester. This semester you will be building a Game Suite that allows the user to select from at least three (3) games of different types (logic, board, card, etc.). You must utilize an Object-Oriented language (i.e., Java, C#, C++) to implement your application, and you may integrate with an approved framework; the list of approved frameworks is mentioned in the Canvas assignment.

Overall Requirements of the project

1. You must implement three games, including game logic, from at least two of the following categories:
 - a. Card Games (solitaire or multiplayer)
 - b. Board Games (checkers, battleship, etc.)
 - c. Logic Games (crossword, wordle, brain teasers)
2. You must have a graphical user interface for the user to select a game to play.
 - a. As a user, I want to select a game to play using a graphical user interface so that I can improve my game-playing skills.
3. You must provide the user with a way to end the game and return to the game selection screen.
4. You must provide the user with a way to exit the Game Suite application from within the GUI of the application.
5. You must include help text for each game, including instructions/rules for gameplay.
6. You must apply sound object-oriented design principles. At a minimum, this includes encapsulating functionality and leveraging hierarchies to reuse common functionality instead of copying and pasting at multiple places.
7. You do not have to implement networking for multiplayer games.
8. You must utilize iterative development to analyze, design, and implement your application.
9. You may use a game logic engine for one of your games. This is for the logic and rules of gameplay, not the game GUI itself.

Sprint Deliverables

Sprint 0.5

1. Team formation and organization
2. Version control setup (i.e., Git)

Sprint 1

Design

1. Requirements analysis of all three games.
 - a. OOA: Domain Model of any one game

- b. OOA: Use Case Diagram of any one game

Implementation

1. Explain the three games selected by your team for implementation across the three categories.
2. Implement Game Selection Main Screen – A way to select one of the three games
3. Implement GUI main screen – for all three games
 - a. Start Screen
 - A way to start the game
 - A way to quit the game
 - b. Initial Configuration Screen
 - An input for player name
 - A way to pick a character sprite to represent the player
 - A way to continue to the game screen
 - c. Game Screen
 - This screen is where the actual game will take place. This is where most of the functionality in later sprints is implemented. For now, the screen will:
 - Display player name
 - Display player character sprite depending on the selected character

Sprint 2

Design

1. Discuss the Software Architecture of your implementation of Game 1
2. OOA: Draw SSD for your Game 1
3. OOD: Draw SD for Game 1
4. OOD: Draw DCD for Game 1

Implementation

Complete implementation of Game 1. Game 1 should be fully functional as described in Sprint 1.

Sprint 3

Design

Show SOLID and GRASP compliance for Game 1 or 2 Implementation (Ref to activity in class on 6/21). If not, revise the implementation design to adhere to SOLID/GRASP.

Implementation

Complete implementation of Game 2. Game 2 should be fully functional as described in Sprint 1.

Sprint 4

Design

Show how your code complies with at least three design patterns, one of each type: creational, structural, and behavioral (Can be for Game 1, 2, or 3)

Implementation

Complete implementation of Game 3. Game 3 should be fully functional as described in Sprint 1.

Sprint 5

Design

Discuss Code Smells and Refactor code as needed for one of the three games.

No Implementation in Sprint 5. However, extra credit opportunities will be there if you add more features to Games 1, 2, or 3.

Example Games

Below are examples of simple games that can be chosen for each category. These are examples for your reference, but you can choose any games that fall within the categories.

Logic Game - "Sudoku"

Board Game - "Tic Tac Toe"

Card Game - "Blackjack"

Here is a brief description of each game. You need to describe the three games picked by your team and the corresponding categories similarly in Sprint 1.

Sudoku: Sudoku is a puzzle game requiring a player to fill a 9x9 grid with digits so that each column, row, and nine 3x3 subgrids that compose the grid contain all of the digits from 1 to 9. This game is a great way to learn about data validation and management in 2D arrays.

Tic Tac Toe: A simple game where the player and computer (or another player) take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. This game would be a good way to learn about AI logic and win conditions.

Blackjack: Blackjack is a card game aiming to beat the dealer's hand without exceeding 21. This can be achieved in one of three ways:

- Get 21 points with your first two cards (this is a blackjack) without a dealer blackjack.
- Reach a final score higher than the dealer without exceeding 21.
- Let the dealer draw additional cards until their hand exceeds 21.

These games also allow for a range of optional enhancements if you want to implement in Sprint 5 for extra credit. For example, a Sudoku game could have a hint system, or a timer to challenge the player. Tic Tac Toe could have different levels of AI difficulty. Blackjack could have a betting system.