VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

Poker Game

Report Name: Final Project

Course name: Fundamentals of Programming

CSC1002 - 24C03

Student name:
Lam Chi Hao
Vo Nhat Tien
Le Nguyen Anh Tri

Teacher name:
Nguyen Hai Minh
Tran Thi Thao Nhi
Phan Thi Phuong Uyen

07/12/2024

Contents

1 Group Information

	ID	**Name**	**Title**
1	24127034	Lam Chi Hao	Leader
~~2~~	~~24127557~~	~~Vo Nhat Tien~~	~~Member~~
3	24127567	Le Nguyen Anh Tri	Member

Table 1: Information of Group 02 member.

2 Table of Contribution

Problem	**Rubric**	**24127034**	**24127557**	**24127567**
Standard features	-			
Initialization	20	Score	Score	Score
Dealing	20	Score	Score	Score
Game play	20	Score	Score	Score
Player's information	20	Score	Score	Score
Advanced features	-			
Draw poker	30	Score	Score	Score
5-card stud	30	Score	Score	Score
Color / Sound effects	30	Score	Score	Score
Leader board	30	Score	Score	Score
Creations	30 / each	Score	Score	Score
Report	-			
	100	Score	Score	Score
	Total	Score	Score	Score

Table 2: Table of contribution

3 Acknowledgement

We would like to sincerely thank our theory teacher, Nguyen Hai Minh, and our lab teachers, Nguyen Thi Thao Nhi and Phan Thi Phuong Uyen, for providing the project and valuable resources for self-studying. Although their guidance was indirect, the materials helped us understand the fundamental concepts needed for this project.

We first consisted of a team of three; however, one of our teammates did not feel this was the correct field for him and left our team. Despite the obstacle that challenged us, we tried our best to give our best effort to finish the project.

We consider ourselves very lucky that this opportunity for the project was given to us and allowed us to learn by experience. Any limitations or shortcomings in this work are entirely our responsibility. We hope that this project will be a stepping stone for us to improve our skills and knowledge in the future.

4 Introduction

This report is our final report of the project *Poker Game* for the course *Fundamentals of Programming - CSC10012*. The project is a poker game application that allows users to play poker with the computer and other players locally. The game is developed using C++ programming language and SDL2 library for graphical user interface. The project is divided into two main parts: core mechanics and graphical user interface. The core mechanics of the game are implemented using classes and structures.

TODO: Add more information about the project, such as POKER RULE here.

The report provides an overview of the project, including the project structure, the flowchart of the game, the structure of the game using SDL2 library, the implementation of the game, and the conclusion.

5 Project Structure

For this part of the report, we will discuss the project structure, including the folder structure, file structure, variable, constant, class, and structure naming conventions.

The naming convention is a set of rules for choosing the character sequence to represent the name of a variable, constant, function, class, file, folder, or other entity in the source code. The naming convention is essential because it helps the reader understand the code more easily and quickly. In this project, we follow the naming convention below:

5.1 Folder Structure

- **Lowercase**: All characters in the folder name are lowercase.
- **Underscore**: If the folder name consists of multiple words, separate them with an underscore.

Our folder structure is as follows:

- **src**: Contains all source code files written in C++. There are two subfolders with two main categories of source code files.
  - **core**: Contains all the source files relate to the core mechanics of the game.
  - **gui**: Contains all the source files relate to graphical user interface of the game.
- **include**: Contains all header files. There are two subfolders with two main categories of header files.
  - **core**: Contains all the header files relate to the core mechanics of the game.
  - **gui**: Contains all the header files relate to graphical user interface of the game.
- **bin**: Contains all executable files, object files and dynamic libraries. There is a subfolder called obj. Inside the obj folder, there are two subfolders with two main categories of object files.
  - **core**: Contains all the object files relate to the core mechanics of the game.
  - **gui**: Contains all the object files relate to graphical user interface of the game.
- **assets**: Contains all assets used in the project. There are three subfolders inside our assets folder.
  - **imgs**: Contains all images used in the project. There are imgs for backgrounds, buttons, cards, cursor and icons for our game application.
  - **audios**: Contains all sounds used in the project. There are audios for background music, button click sound, card shuffle sound and card flip sound.
  - **fonts**: Contains all fonts used in the project. There are fonts for the game title, game buttons and game text.
- **libs**: Contains all libraries used in the project.

  - **SDL2**: Contains all SDL2 libraries. This is our main library for creating the graphical user interface.
  - **cmake**: Contains all CMake libraries.

- **report**: Contains all report files which are source code files written in LaTeX.
  - **contents**: Contains all the contents of the report. We want to keep the report organized, so we separate the report into multiple files and put them in the contents folder.
  - **figures**: Contains all figures used in the report. The figures are images, graphs, tables, etc.

5.2 File Structure

- **lowercase**: Only one exception is our main file, which is named **main.cpp**.
- **CamelCase**: We decide that all of the header files and source files are named in CamelCase.
- **Underscore & hyphen**: This is for files in SDL2 libraries, which are named in lowercase and separated by an underscore. Beside that, our files in assets are named in lowercase and separated by an underscore or hyphen.

5.3 Variable, constant, class, structure

- **camelCase**: We decide that all of the variables are named in camelCase. Notice that the first letter of the first word is lowercase, and the first letter of the following words is uppercase.
- **CamelCase**: For classes and structures in our project, we decide that all of the classes and structures are named in CamelCase. Notice that the first letter of each word is uppercase.
- **ALL CAPS**: We decide that all of the constants are named in ALL CAPS. Notice that if the constant consists of multiple words, we will separate them with an underscore.

5.4 Function Prototype and Definition

We decide that all of the functions are named in camelCase. Notice that the first letter of the first word is lowercase, and the first letter of the following words is uppercase.

6 Chart

This part will provide a chart of the project's structure. The chart will show the relationship between the classes and the functions of the project. The chart will be divided into two parts: the first part will show the relationship between the classes, and the second part will show the functions of the project.

TODO: Write more about the chart of the project's structure

6.1 Core class relationship

TODO: Write more about the class relationship

TODO: Add the chart of the class relationship

6.2 GUI function relationship

TODO: Write more about the function relationship

TODO: Add the chart of the function relationship

6.3 Program flowchart

TODO: Write more about the program flowchart

7 SDL2 Game Structure

In this section, we will explore the fundamental structure of our SDL2-based poker game. The game follows a standard SDL2 application architecture with the following key components:

- **Initialization**: Setting up SDL2, creating windows and renderers
- **Game Loop**: The main loop that handles:
  - Event handling (user input)
  - Game state updates
  - Rendering
- **Resource Management**: Loading and managing textures, fonts, and sounds
- **Cleanup**: Proper deallocation of resources

7.1 Core Components

The game is structured around several core classes:

- `Game`: Main class managing the game state and loop
- `TextureManager`: Handles loading and rendering of images
- `InputHandler`: Processes user input
- `Player`: Manages player data and actions
- `Card`: Represents individual playing cards
- `Deck`: Handles the deck of cards and dealing

Figure 1 illustrates the relationship between these components.

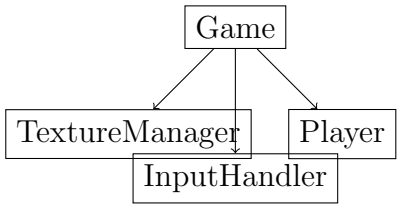

Figure 1: Basic game structure diagram

8 Implementation

The implementation of the project is divided into two main parts: core mechanics and graphical user interface. The core mechanics of the game are implemented using classes and structures. The graphical user interface is implemented using the SDL2 library. The project is developed using C++ programming language.

8.1 Core Mechanics

The core mechanics of the game are implemented using classes and structures. The core classes of the project are `Game`, `Player`, `Card`, and `Deck`. The `Game` class manages the game state and loop. The `Player` class manages player data and actions. The `Card` class represents individual playing cards. The `Deck` class handles the deck of cards and dealing.

- **Game**: Main class managing the game state and loop
- **Player**: Manages player data and actions
- **Card**: Represents individual playing cards
- **Deck**: Handles the deck of cards and dealing

9 List of References

- LaTeX template
- LaTeX on Visual Studio Code tutorial
- SDL-2.30.9 library
- SDL-2.30.9 installation tutorial
- CMake-3.21.3 library