**2048 Game Report**

**Introduction :**

Welcome to our Object-Oriented Programming (OOP) project on the classic puzzle game 2048! If you're not familiar with the game, the objective is to slide numbered tiles on a grid to combine matching numbers and reach the elusive 2048 tile. It may sound simple, but with each move, new tiles are added to the grid and the game can quickly become challenging.

In our OOP project, we have implemented the 2048 game using concepts such as inheritance, Aggregation, and encapsulation. Inheritance allows us to create a hierarchy of classes, where one class can inherit properties and behaviors from another class. This helps us to organize our code and avoid repeating ourselves.

Aggregation is a type of association that represents a part-whole or a has-a relationship between objects. It is a way to model a relationship in which one object contains or is composed of other objects.

Encapsulation is the practice of keeping the implementation details of a class private, and only exposing a public interface for interacting with the class. This helps to protect the integrity of the class and prevent outside code from making changes that could break the class.

We have also included intuitive controls and challenging gameplay in our 2048 game. Players can use the arrow keys on their keyboard to slide the tiles in the desired direction, and the game will automatically combine matching numbers and add new tiles to the grid. The game ends when there are no more moves left and the player has reached the highest score possible.

We hope you enjoy playing our version of 2048 as much as we enjoyed creating it! With its addictive gameplay and OOP design, it is sure to provide hours of entertainment.

**Classes Introduction:**

I have divided my project in 5 different classes. Here’s a quick introduction about the relationships between the classes.

**Tile Class :**

This class contains the value for each cell, and the information about the cell such that whether it is a mine, and it contains a special variable isCombined, which is used in the moving of the tiles, which holds the information if the cell is already combined/merged with other cell then it prevents from merging it again.

**Game2048 :**

This class provides a template for the whole 2048 gameplay, mainly managing its grid and the points. The methods like move up, move down, move left and move right are the essential part of the game.

**Record Class :**

It is a node class, serves as a basis for the Hall of Game data/records. As it is a collection of 2 things i.e name and points.

**HallOfGame Class :**

It is a whole collection/array of the records of the data from the file, it loads all the data of the file right at the beginning and shows everytime user demands it so that we don’t need to open and close file and access data directly form the fle.

**Game Class :**

It is the main class of this project which manages all the contents of the project including game and file handling. It is inherited from Game2048 and then the HallOfGame class is aggregated in it.

**UML Diagram :**

