# SUBMISSION: COURSE WORK 2

This document is a report on the program Dice Poker required for the course work submission.

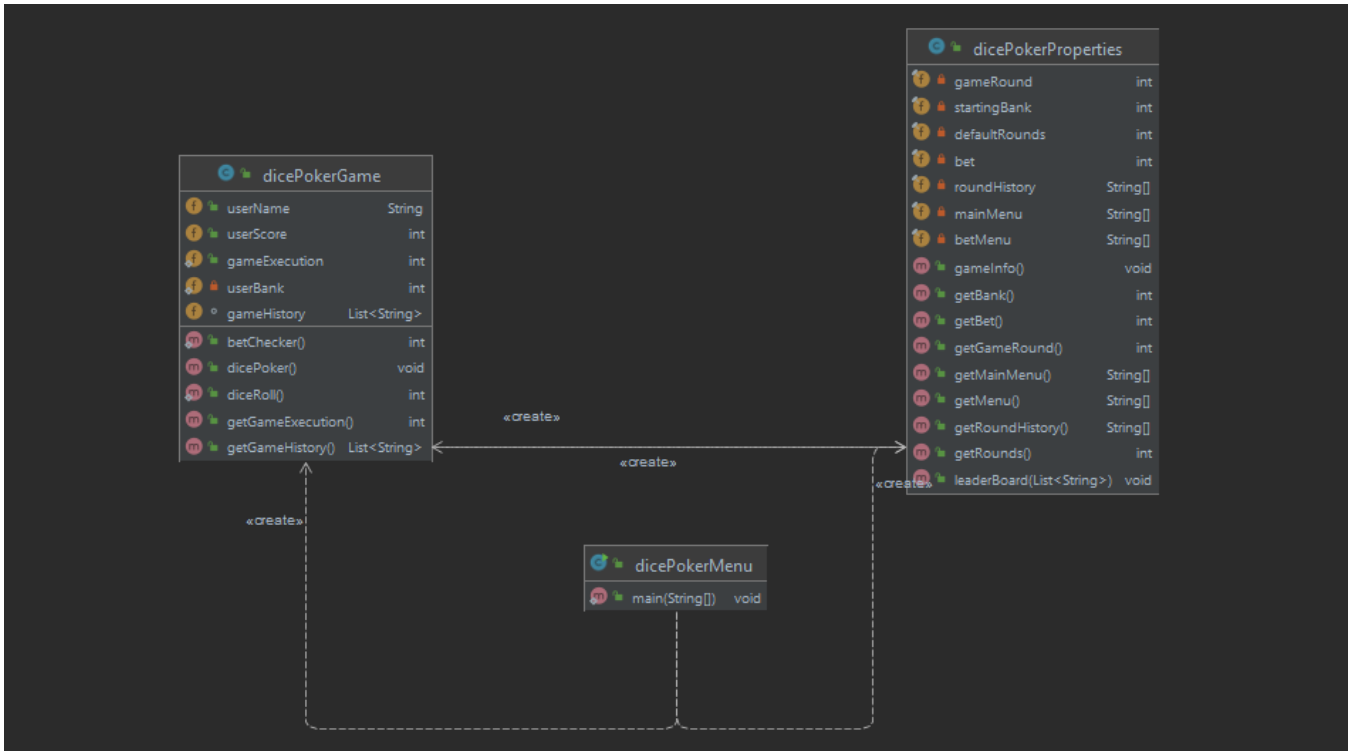
40524216

# Contents

# Version History

| Version | Date | Summary Of changes |
|---|---|---|
| 0.1 | August 19, 2021 | Initial Creation |

# UML Class Diagram




## dicePokerProperties:

The dice poker properties class contains most of the dicePokerGame variable data such as starting bank amount, rounds of the dice poker game, menu data etc. found in dicePokerMenu. The dice poker properties also contain methods used for the leaderboard functionality presented at the end of every game and preview at the menu with gameInfo used to provide details on playing the game.

## dicePokerGame:

The dice poker game class contains all the functionality of simulating the dice poker game. The dicePoker() method runs the game calling helper methods such as betCheker() to validate bets made on every round and diceRoll to simulate the action of rolling a pair of dice. All game details are grabbed from the dicePokerProperties class, at the end of the game the players' username and score are stored and passed into the dicePokerProperties class to update the leaderBoard list.

## dicePokerMenu:

The dice poker menu pulls both class' together to provide the player a main menu. A do-while loop is used to provide options declared in the dicePokerProperties that are used to class objects created from the dicePokerGame class & dicePokerProperties for running the game and checking the leaderboard before starting the game.

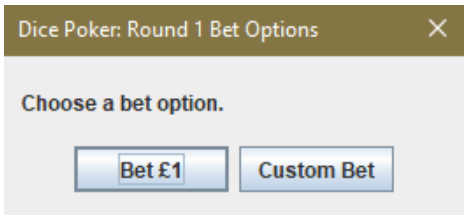
# Extensibility:

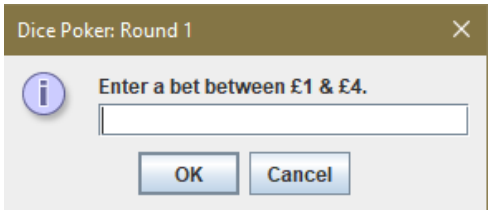
This program is extensible in the way each method and variable are used to account for one another. When changing the game rules (game rounds & starting bank amount), these can be easily manipulated in the dicePokerProperties class by changing the variables startingBank, &rounds. The sides of the dice can be manipulated in the diceRoll() method found in the dicePokerGame class.

# Advanced features:

## Variable bet amount

The variable bet amount was implemented used a betMenu to provide the player various options on placing their bet from dicePokerProperties. The player can make a bet of £1 or place a custom bet between £1 and 4£. This betMenu is utilized in the betChecker() method to validate the bet being entered is a valid one (a whole number between 1 & 4).




## High score table

The highs core table was implemented using a list in the dicePokerProperties class. The method leaderboard consists of default high scores stored in the list; this list then adds the players score after every game execution using a for loop to add the gameHistory into the list. This array list is sorted using the super class scoreComparator that split string entries (e.g. "19 Nicole") using *.split* to grab the score entry and compare each using *.compare* to validate each entry and return score. The class scoreComparator is then sorted using the .sort function in reversed order to list the scored from Highest to Lowest.