CRC Model

User	
username: username of User password: password of User gameMode: difficulty of User matchHistory: collection of each game this User has played (Mode, numMoves, time) getUsername: gets username of user setPassword: if user wants to change password getGameMode: current game mode user is in setGameMode: changes difficulty of gamemode	UserDatabase UserAccount

Entity –> Enterprise Business Rules

UserManager	
changeGameMode: changes the current game mode of User changePassword: changes password of User (requires getPassword and setPassword from User)	User

Use Case–> Application Business Rules

UserCreator	
createUser : creates a new User object with username, password, empty matchHistory for each of the modes checkUsername : helper for createUser that checks if username exists in userDataBase	User promptLogin UserDataBase

Use Case–> Application Business Rules

promptLogin	
User login (userLogin) User sign up (createNewUser)	UserCreator userLogin

Use Case–> Application Business Rules

InfoChecker	
password entered valid/invalid (from userDatabase) username exists/doesn’t exists	UserDatabase UserLogin

Use Case–> Application Business Rules

UserLogin	
Prompts login/signup Takes user input for username and password and calls infoChecker methods to see if information is valid	InfoChecker promptLogin

Command Line/UI

UserDatabase	
Stores a collection of Users and user data in a local file	User createNewUser

Database –> Frameworks and Drivers

HistoryUpdater	
updateHistory: adds the last match with data (gamemode, moves, time) played to User	User endGame

Use Case–> Application Business Rules

UserInputManager	
getInput : gets the user’s input for moves	runGame

Command Line/User Interface –> Frameworks and Drivers

Board (abstract)	
Shape: number of rows and column Size: rows x columns (number of tiles that are put on) numPairs: size/2	BoardGenerator TileBoard

Entity –> Enterprise Business Rules

Tile	
key : integer value getKey : gets key value of Tile Theme : theme of tile	TileBoard

Entity –> Enterprise Business Rules

TileBoard	Board
revealedTiles: keeps track of which Tiles are always flipped because they are matched boardState: revealed and unrevealed cards for the current state (even during a players move) numMoves: the number of moves for this board (2 cards flipped) getBoardState Time: time for this board game	Tile Board runGame User BoardGenerator

Entity –> Enterprise Business Rules

BoardGenerator	
Creates TileBoard with size based on User.GameMode, (num moves, time) = 0, 0	User TileBoard

Use Case–> Application Business Rules

BoardUpdater	
updateBoardState: updates board based on user inputs (eg. keeps matched cards flipped all the time) updateNumMoves: updates number of moves for this board after user’s second input per turn timeUpdater: constantly running to keep track of time	getUserInput runGame TileBoard

Controller –> Interface Adapters

runGame

generateBoard : create board based on user’s input difficulty (randomly place tiles)
getUserInput: constantly await for user input until all tiles are matched
showTile : userinput reveals tile on TileBoard
checkMatch : helper for changeGameSettings, check if 2 consecutive tiles are a match
changeGameSettings: changes settings
updateBoard: updates Board whenever User moves, calls checkMatch to see if second move matches first move

gameSettingsManager
BoardGenerator
BoardUpdater
UserInputManager
TileBoard

Controller –> Interface Adapters

Leaderboard	
highScores: ordered list for the highest scores for each game mode getLeaderboard: Contains/displays top 10 scores and username out of all users for a given difficulty setLeaderboard: changes list of high scores	LeaderboardManager

Entity –> Enterprise Business Rules

LeaderboardManager	
addToLeaderboard: for a gamemode, adds (User, gamemode, num moves, time) if it has higher num moves than one on the leaderboard. If num moves is tied with another, check time showLeaderboard: shows leaderboard	Leaderboard endGame

Use Case –> Application Business Rules

endGame	
checkLeaderboard promptNewGame Add in match history (gamemode, num moves, time) to User’s match history	BoardGenerator historyUpdater LeaderBoardManager

Controller –> Interface Adapters

gameSettings	
isOpen isMuted Volume setVolume wantsToExit	gameSettingsManager

Entity –> Enterprise Business Rules

gameSettingsManager	
Mutes the sound effects Can adjust volume Resets game for user Exits game for user	gameSettings runGame

Use Case–> Application Business Rules