**UML Diagram**

**Explanation:**

firstName 🡪 User’s first name

lastName: 🡪 User’s last name

age: 🡪 User’s age

score: 🡪 User’s game score

username: 🡪 Player’s username

password: 🡪 Player’s password

level: 🡪 Player’s level

experience: 🡪 Player’s experience

perks: 🡪 Player’s available game perks

achievements: 🡪 Player’s gained achievements

items: 🡪 Player’s items

Account() 🡪 Creates object with all parameters

setScore() 🡪 Sets user score

setUser() 🡪 Sets username

setPass()🡪 Sets password

setLevel()🡪 Sets account level

setFname()🡪 Sets user’s first name

setLname() 🡪 Sets user’s last name

addPerks() 🡪 Adds perk to account

addAchieve() 🡪 Adds achievement to account

addItem () 🡪 Adds item to account

setAge() 🡪 Sets user’s age

getLevel() 🡪 Gets account level

getPerk() 🡪 Gets account perks

getAchieve() 🡪 Gets account achievements

getItem() 🡪 Gets account item

getScore() 🡪 Gets user score

getUser() 🡪 Gets username

getPass()🡪 Gets password

getFname()🡪 Gets user’s first name

getLname() 🡪 Gets user’s last name

getAge() 🡪 Gets user’s age

toString() 🡪 Displays all account info

|  |
| --- |
| Account |
| -firstName: String  -lastName: String  -age: int  -score: int  -username: String  -password: String  -level: int  -experience: int  +perk: Perk(5)  +achievement: Achievement(10)  +item: Item(1) |
| Account(fn: String, ln: String, ag: int, scor: int, user: String, pass: String)  +setScore(scor: int): boolean  +setUser(user: String): boolean  +setPass(pass: String): Boolean  +setLevel(exp: int): boolean  +setFname(fname: String): boolean  +setLname(lname: String): boolean  +addPerks(perk: Perk): boolean  +addAchieve(achievements: Achievements): Boolean  +addItem(nam: String, func: int): boolean  +setAge(ag: int): Boolean  +getLevel(): int  +getPerk(): String  +getAchieve(): String  +getItem(): String  +getScore(): int  +getUser(): String  +getPass(): String  +getFname(): String  +getLname(): String  +getAge(): int  +toString(): String |

**Explanation:**

perkName 🡪 The Perk’s name

levelRequired: 🡪 The level of difficulty the perk requires

perkNum: 🡪 Assignment to perk function

Perk()🡪 Creates object with all parameters

setPerkName() 🡪 Sets perk name

setLevelRequired() 🡪 Sets level of difficulty

setPerkNum() 🡪 Sets number to assign perk function

getPerkName() 🡪 Gets perk name

getLevelRequired()🡪 Gets level of difficulty

getPerkNum()🡪 Gets assigned perk number

toString() 🡪 Displays all info

|  |
| --- |
| Perk(Aggregation) |
| -perkName: String  -levelRequired: int  -perkNum: int |
| Perk(perkName: String, levelRequired: int, perkNum: int)  +setPerkName(pn: String): boolean  +setLevelRequired(lvl: int): boolean  +setPerkNum(pNum: int): boolean  +getPerkName(): String  +getLevelRequired(): int  +getPerkNum(): int  +toString(): String |

**Explanation:**

achieveName 🡪 The achievement’s name

experience: 🡪 Amount of experience for achievement

Achievement() 🡪 Creates object with all parameters

setAchieve() 🡪 Sets achievement name

setExp() 🡪 Sets amount of experience

getName() 🡪 Gets achievement name

getExp() 🡪 Gets achievement experience

toString() 🡪 Displays all info

|  |
| --- |
| Achievement (Aggregation) |
| -achieveName: String  -experience: int |
| Achievement(nam: String, ex: int)  +setAchieve(nm: String): boolean  +setExp(ex: int): boolean  +getName(): String  +getExp(): int  +toString(): String |

**Explanation:**

itemName: 🡪 Name of item

itemNum: 🡪 Assignment of item function

Item() 🡪 Creates object with all parameters

setItemName() 🡪Sets the item’s name

setItemNum() 🡪 Sets the number to assign item function

getItemName() 🡪 Gets item name

getItemNum() 🡪 Gets assigned item number

toString() 🡪 Displays all info

|  |
| --- |
| Item (Composition) |
| -itemName: String  -itemNum: int |
| Item(itemName: String, itemNum: int)  +setItemName(iNam: String): boolean  +setItemNum(iNum: int): boolean  +getItemName(): String  +getItemNum(): int  +toString(): String |

**Sketches**

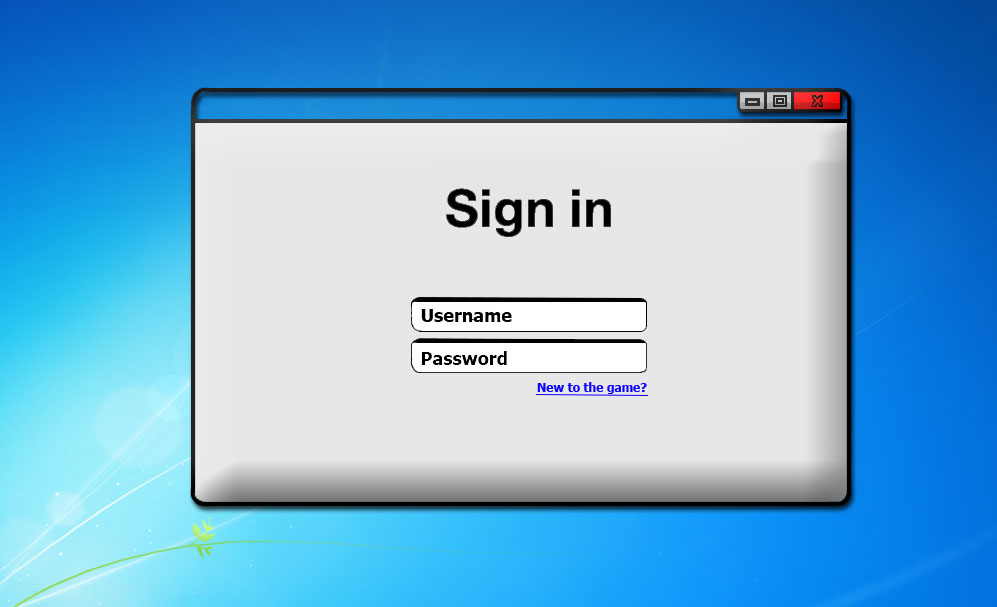


Figure 1: As soon as the game is launched the player will get the option to either sign in or sign up if it is their first time playing the number guessing game. You may sign in with a username as well as a password. On this screen there is an option to create a username via a button directly under the password text box.

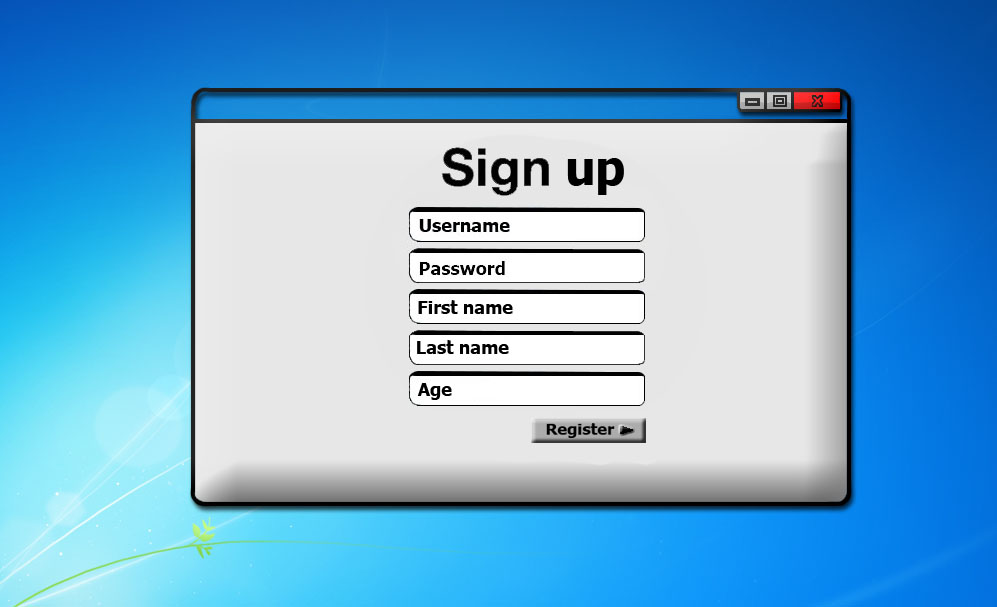


Figure 2: If the user chooses to create a new account rather than sign in, they may do so by entering a new username, password, first name, last name, and the age of the user. After entering all the required information, the register button near the bottom creates a usable account tailored to the user.

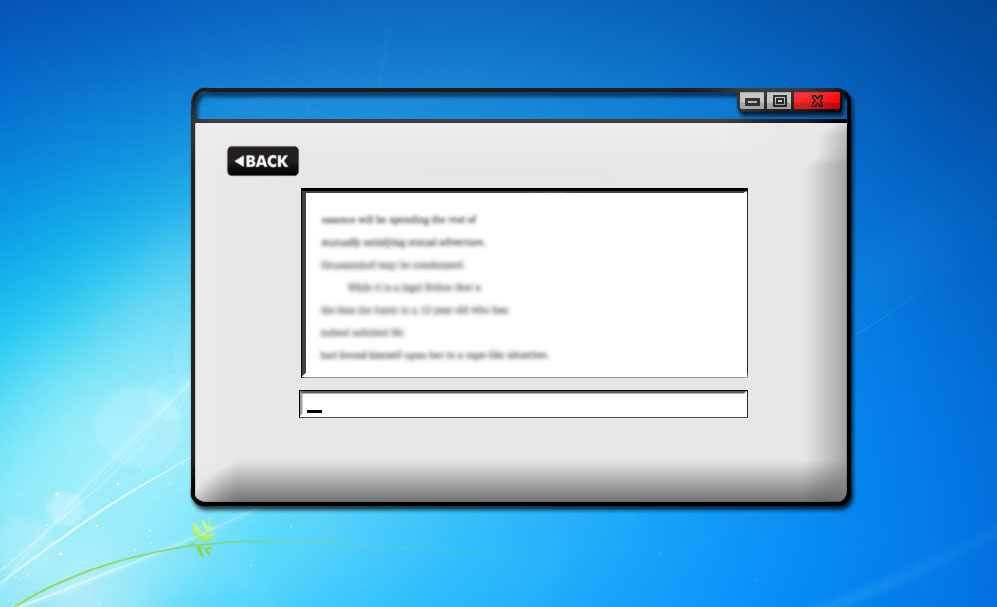


Figure 3: After entering account information, the game will begin. An option of 1 or 2 player vs will be given; if 2 player is chosen then the second player must sign in using the same procedure that player 1 went through. The difficulty is then chosen and the game is played until the correct number is guessed. Afterwards a leaderboard showing top scores is displayed.