Name: Carmen Tam
Person ID: 50194532

CSE426/526

Top Game of 2019

Abstract

As technology grows, the world around the virtual world has also grown. More specifically, the era of gaming. As the popularity of gaming grew, the amount of games produced per year is exponentially greater. Annually, there are articles that flood across the internet listing the best games that came out that year. However, based on the writer or the website the article was published in, prejudice can play a huge role in their ratings. This causes many readers to get a biased opinion that might steer them towards a game that many have had bugs, problems, or even an overall bad experience playing. All this relating to social media influence as well as false advertising.

This project allows a voting system that lets the users choose their favorite game of the year themselves based on their experience. It revolves around a voting system that involves a point system that utilizes user inputted information. Upon start, the user will be given four initial functions that will prompt them to enter:

1. User's age
2. Amount of games the user has owned in 2019
3. Amount of years the user has played games for
4. User's gender

The inputted values will be entered as an integer. Depending on the data that the user enters, the system will calculate the amount of value their vote will count for when the user votes. For example, the user's vote will be weighted more if:

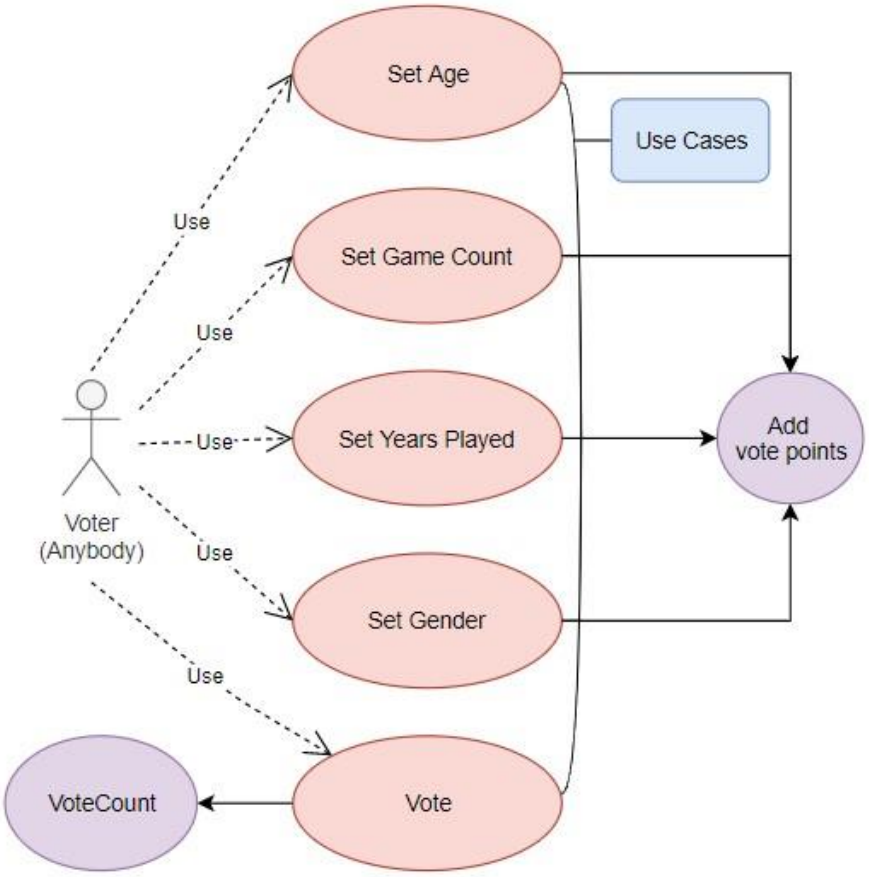
1. The user is at least 21
2. The user owns at least 3 games in 2019
3. The user has played games for over 4 years
4. The user is a female (Population balance)

However, the user may not vote if:

1. The user is younger than 16
2. The user has not owned a game in 2019
3. The user has not played games for at least 2 years
4. Invalid inputs for any criteria (strings or unfollowed instructions)

Additionally, the user will then vote for their favorite game based on a list that will be displayed through Dapp. This will input as an integer value from 0-9.

Use Case Diagram:



Any user can vote, depending on the requirements of each criteria, the user may be unable to vote or their vote will be weighted more/less. Depending on these requirements, their vote may or may not be counted once inputted in the end.

Contract Diagram:

<CONTRACT DIAGRAM>

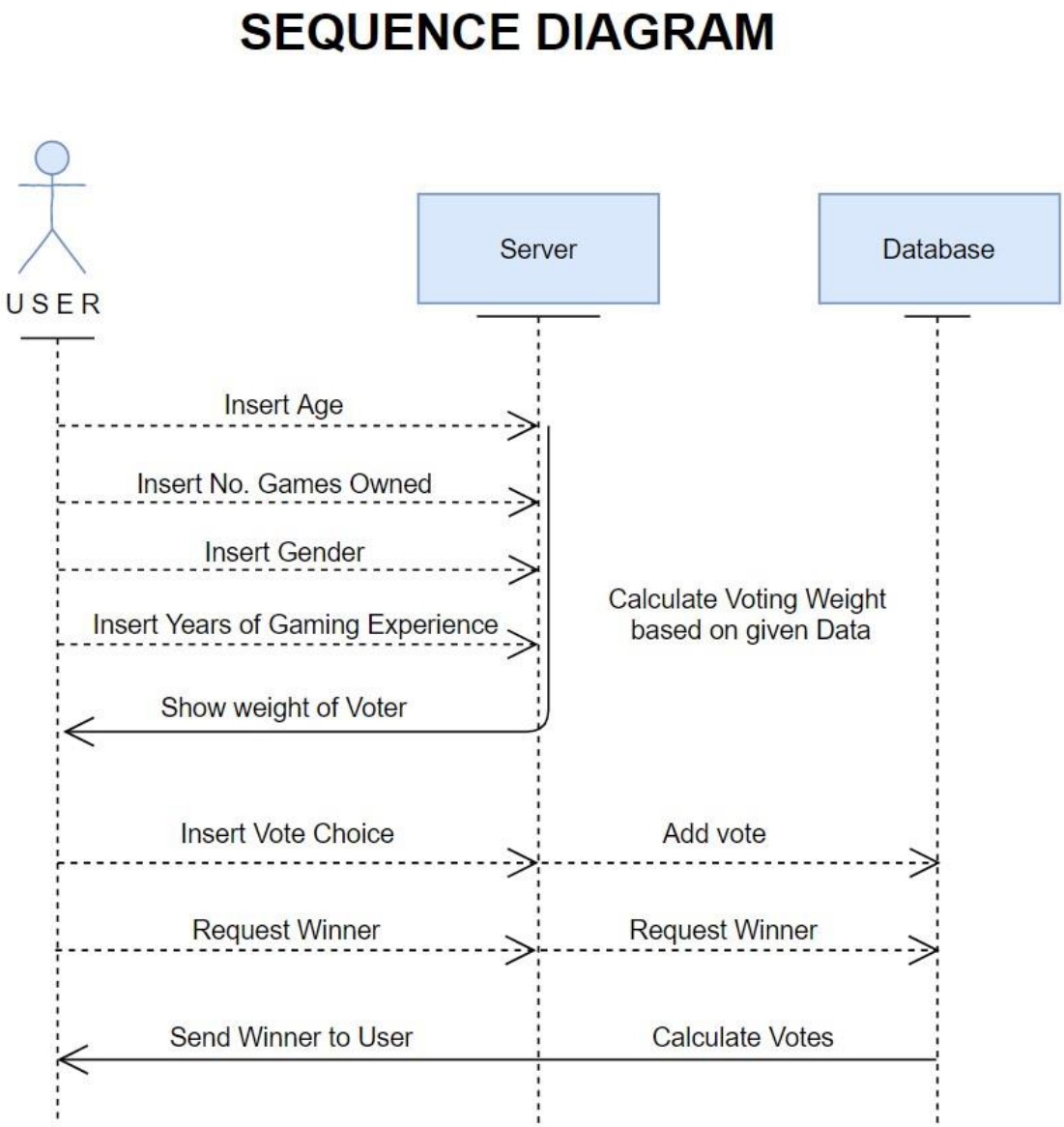
BGBallot

```
struct Voter { }
struct Proposal { }

Proposal [ ]  games;

address user;
mapping(address => Voter) voters;
```

```
 modifier checkAgeReq (uint n)
    { require(n >= 16);
       _;
    }
modifier checkGenderReq (uint n){
    { require(n < 2);
       _;
    }
modifier checkYearCount (uint n){
    { require(n >= 2);
       _;
    }
modifier checkGameCount (uint n){
    { require(n >= 1 );
       _;
    }
```

```
constructor (uint8 numGames) public

setAge (uint _age) checkAgeReq (_age) public { }

setGender (uint _gender) checkGengerReq (_gender) public { }

setYearCount (uint _yearsplayed) checkYearCount (_yearsPlayed ) public { }

setGameCount (uint _gamesOwned) checkGameCount (_gamesOwned) public { }

Vote (uint8 _choice) public { }

reqWinner ( ) public view returns (uint8 winningProposal) { }

getPointsB4Vote ( ) view public returns (uint) { }
```

Sequence Diagram:



Architecture Diagram