Amar Trivedi, Subhan Mirpour, Christopher Gugelmeier, Jake Holmes

Data Structures

- JSON files will be used for data storage.
- Username, experience, wins, losses, and other stats are objects that will be stored.
- Most fields will be private with accessor functions.
- The JSON file will be read in from the games folder location and be iterated through to get all of the available users or if a new one needs to be created. When the players finish each game their stats will be updated and saved to the file once they are done playing.
- Constructor that takes a JSON file as an argument, and creates the player object from that information.
- The data will be stored in classes such as a card class or tower class.

In game stats
- Elixir spent
- Cards placed
- Towers destroyed
- Enemies defeated
- Favourite card type
- Favorite card

After game stats
- Games played
- Games won
- Time played

  Calculated
  - Games lost
  - Win percentage

Data Flow

- Once there are 2 player objects, start the game.
- Shuffle cards
- Display 4 usable cards for each player
- The 2-minute timer begins.
- Start each player with 5 elixirs (max 10), add 1 elixir per second for each player.
- Players place cards, consuming their elixir.
- The next card in rotation will show when a card is played. 4 cards must always be visible to the player.
- Cards attack each other or towers, until one player has destroyed the others main tower.

- When 2-minute timer runs out, if number of towers alive are equal on both sides, add 1 more minute (2x elixir gained speed)
- After game ends, user stats are updated with win/loss increment.

System Architecture and Assets

- Using the Unity 2D Game Engine.