

## Project Specification for WeConnect4 (Team 32)

### Product Backlog:

- Implement User Login and Registration
- Create Game Start / View Page.
  - Implement logic for game full / join
- Create Profile Page
- Design GUI using Javascript and HTML/CSS
- Implement Game engine logic
  - Logic for tile placement
  - Logic for winning
  - Logic for ties
  - Properly handle game object deletion after game is over
- Implement Leaderboard functionality
- Implement games with online players view
  - Display games where opponent is online
- Incorporate OAuth

### Sprint Backlog:

- Create mockups for each page
- Describe functionality and necessary functions
- Create all models
- Implement Basic User Login and Registration
  - Include color picker (primary and secondary)
- Create Profile Page
  - Include color picker (primary and secondary)

### Product Owner (Sprint 1):

- Kyle Kauffman (kkauffma)

### Data Models: [Link to file](#)

```
from django.db import models
from django.contrib.auth.models import User

class Profile(models.Model):
    user = models.OneToOneField(User, on_delete=models.PROTECT)
    image = models.FileField(
        default='connect4/static/connect4/profile.jpg',
        upload_to="connect4/static/connect4/images")
    content_type = models.CharField(max_length=50, default='image/jpeg')
    primary_color = models.CharField(max_length=7)
    secondary_color = models.CharField(max_length=7)
    total_wins = models.IntegerField(default=0)
    total_losses = models.IntegerField(default=0)
    total_ties = models.IntegerField(default=0)
```

```

def __str__(self):
    return self.user.first_name + " " + self.user.last_name

class GameObject(models.Model):
    # Board should be a JSON field which is a 2d list
    board = models.JSONField()
    # whoever starts the game will be player1
    player1 = models.ForeignKey(
        Profile, on_delete=models.PROTECT, related_name="player1")
    player2 = models.ForeignKey(
        Profile, on_delete=models.PROTECT, null=True, related_name="player2")
    # Colors will be a color value
    https://stackoverflow.com/questions/8863810/python-find-similar-colors-best-way
    # will need to find if colors are too similar
    player1_color = models.CharField(max_length=7)
    player2_color = models.CharField(max_length=7)
    turn = models.ForeignKey(
        Profile, on_delete=models.PROTECT, related_name="turn")
    outcome = models.ForeignKey(
        Profile, on_delete=models.PROTECT, related_name="outcome")
    # Null when not started, False when in progress, True when over
    game_over = models.BooleanField(null=True)
    moves_played = models.IntegerField(default=0)

```

## Action Descriptions:

### Authentication (Register + Login)

A user can input relevant form fields to register themselves on the website. This would help us uniquely identify players. No two users can have the same username. On registration, the user is automatically logged in and redirected to the PlayArea. The same credentials can be used to login at any point. Login is required to view all other pages on the website.

### Profile

User info is visible on this page. Users can upload a profile picture, view user's statistics (win/loss/draw record, games played, points, etc.) and can customize their game disks with colors. The player's disk would be either a primary or secondary color of their choice.

### Create or join a game (PlayArea)

Users can join ongoing game rooms or create a new game room. They can find opponents by challenging their friends or other people that are online at that time to play a game. A game begins when exactly 2 players have joined a room after which no other users can join the room as players. Users can enter ongoing game rooms as spectators.

#### Play a game (Game room - Play mode)

Users can see the latest state of their Connect4 grid and game information of self and opponent. When it is their turn to play, they can click on a column to drop their disk in that column. All rules of the game need to be followed and for any invalid move or attempt, an appropriate message is displayed to inform the user. When a user connects 4 disks in a row, their win is displayed. If the grid is full and/or no player could connect 4 disks, the game is declared a draw. All users in the room can comment in the chat for that room. Users can leave a room at any point or shake hands to draw a game mutually.

#### Spectate a game (Game room - Spectate mode)

Users can join a room as spectators. They have the same view of the game with its current state but cannot modify the grid. Spectators can watch, react and comment.

#### Leaderboard

Users can see the rankings of all users based on the best win ratio. Their ranking is highlighted when they visit the leaderboard.

#### Navigation

Users can navigate between pages using the nav bar. Users can join multiple games through different tabs if they wish, but they need to ensure that they play their turn in time. Users can see a list of active game rooms in which they are playing and can join back if they exited the room due to any reason.