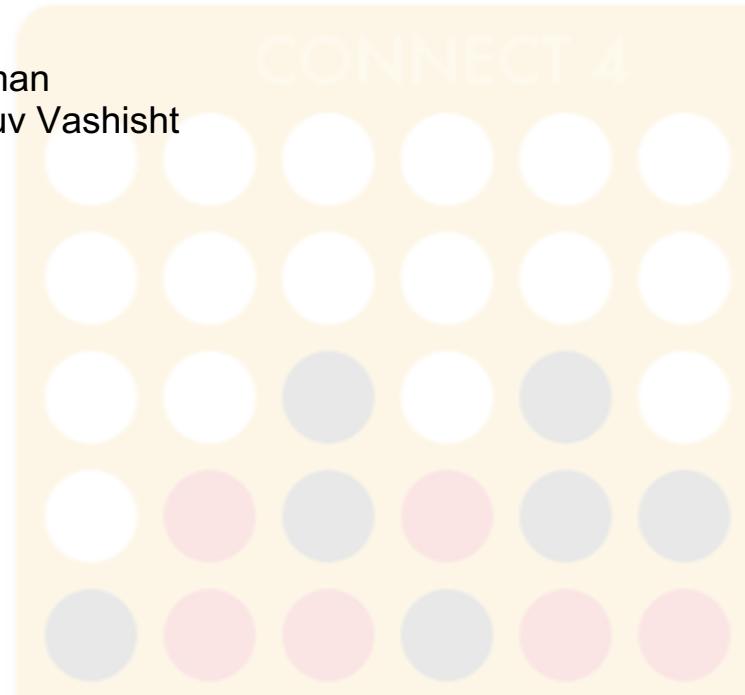


TEAM 32 SPRINT 1

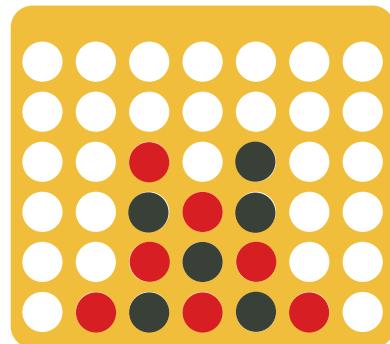
WeConnect4

Product Owner: Kyle Kauffman

Developers: Devika Shanbhag, Dhruv Vashisht



WeConnect4 is a turn-based online multiplayer connect four game, allowing players to **compete through their web-browsers**.



WeConnect4

- 2 players/game
- Turn based
- **Real-time**
- Customizable
- Create/Join game boards
- Play with/Follow friends
- Spectate matches
- Leaderboard

AUTHENTICATION

The screenshot shows a web browser window with the URL `127.0.0.1:8000/connect4/register`. The page has a blue header bar with the text "Connect4". Below the header, there is a "Login" button and a "Register" link. The main form contains the following fields:

Username:	Devika
Password:	*****
Confirm:	*****
E-mail:	dshanbha@andrew.cmi
First Name:	Devika
Last Name:	Shanbhag
primary_hex_color:	Red
secondary_hex_color:	Grey

At the bottom left is a "Register" button.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/connect4/login`. The page has a blue header bar with the text "Connect4". Below the header, there is a "Login" button and a "Register" link. The main form contains the following fields:

Username:	Devika
Password:	*****

At the bottom right is a "Submit" button.

PLAY

[Create New Game Room](#)[Find Opponent](#)

1



mkbhd

VS**JOIN**

2



baconpancakez

VS**JOIN**

3



noobmaster

VS

WAITING..

4



dhruvv

VS

devikas

SPECTATE

5



vmedward

VS

barry

SPECTATE



Kyle

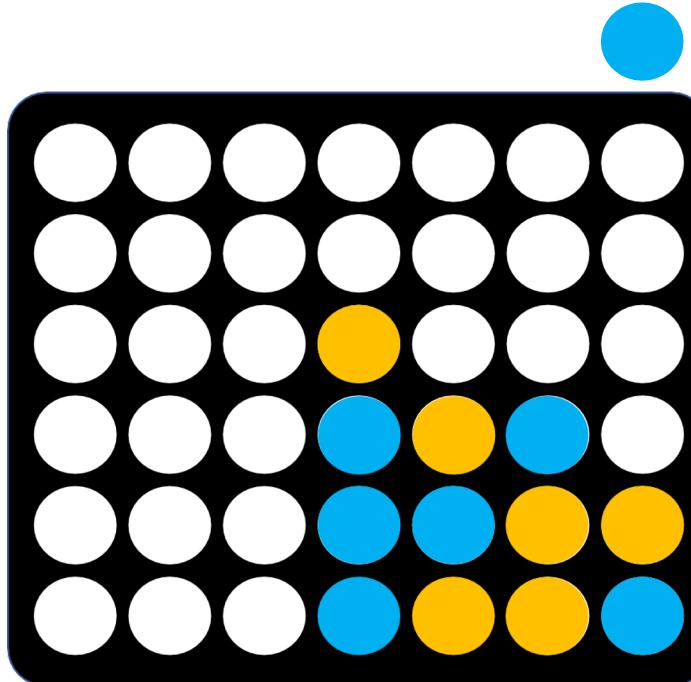
Player Information

TIMER



Devika

Player Information



CHAT

Dhruv: Go Kyle!

Dhruv: Good save, Devika!

Jeff: This is so intense!

Jeff:

Sujata: Watch out!

Kyle: I got this!

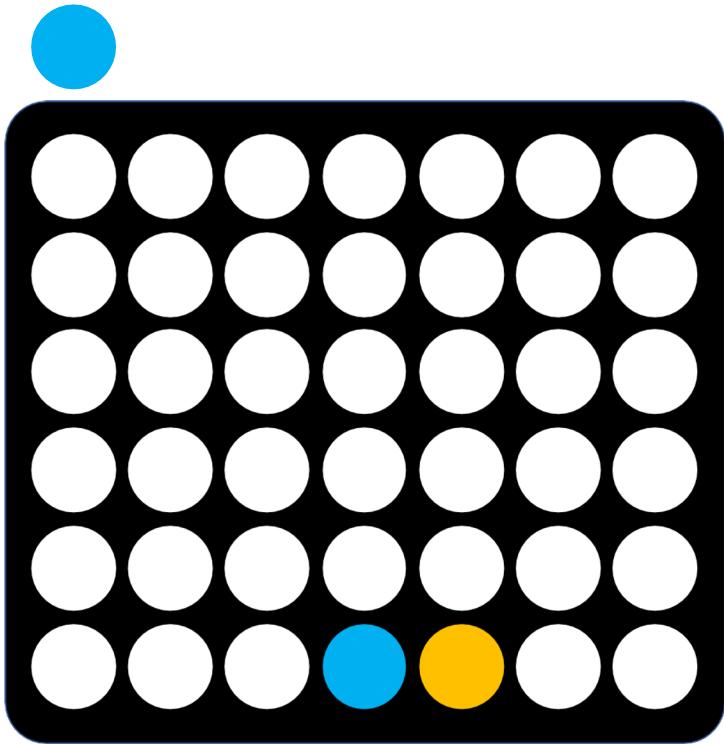
Enter text here





Devika

Info



TIMER

Enter text here

Send



Kyle

Info



DHRUV

@dhruvv

EDIT

Choose Photo

LegoBat.png

Choose Color

STATS

Reset

PLAYED: 26

WON: 15

LOST: 11

LAST 10 GAMES



[Home](#)
[Profile](#)
[Leaderboard](#)
[Logout](#)



LEADERBOARD



1		<i>noobmaster</i>	0.98
2		<i>baconpancakez</i>	0.97
3		<i>dhruvv</i>	0.94
4		<i>kermit_da_balrog</i>	0.89
5		<i>darude_sandstorm</i>	0.82

DATA MODELS

```
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
```

DATA MODELS

```
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)
```

PRODUCT BACKLOG

- Implement User Login and Registration
- Determine the Data models and control flow
- Create Game Play / Spectate Page.
- Implement logic for game full / join
- Create Profile Page
- Design GUI using Javascript and HTML/CSS
- Implement Game engine logic
 - Updating game state
 - Logic for disk placement
 - Logic for winning
 - Logic for ties
 - Handling invalid situations and edge cases
 - Properly handle game object deletion after game is over
- Implement Leaderboard functionality

PRODUCT BACKLOG

- Support viewing online players
 - Ability to challenge online players to be opponents
- Incorporate OAuth
- Allow customization of disk color based on users primary and secondary color preferences
- Deploy application on the cloud
- Testing
 - Integration
 - Security
 - Load
 - User

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.)
 - This needs to be modified for OAuth
- ✓ Create Profile Page
 - Include color picker (primary and secondary)
- ✓ Create Game Page
 - Basic view of the game and opponents

NEXT SPRINT..

- Create Game Start / View Page.
- Implement logic for game full / join
- Create Profile Page
- Design GUI using Javascript and HTML/CSS (simple version)
- Begin implementing game logic

Next Sprint:

Product Owner: Devika Shanbhag (dshanbha)

CURRENT FUNCTIONALITY

QUESTIONS?