

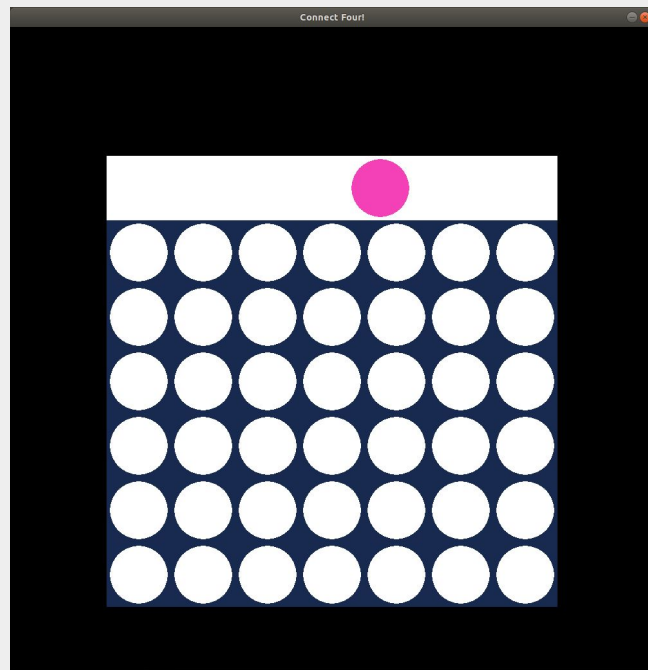
Final Project KB: **CONNECT FOUR**

<https://github.com/arommal/kb-connect-four>

Rosa Valentine Lammora
Afia Hana Yusriya

05111840000035
05111840000011

Board Design



Board 6 x 7

Design Assets

- Logo
- Buttons
- Background

Libraries

Library yang digunakan

1. **Pygame** : Untuk operasi pembuatan game
2. **Math** : Untuk memudahkan operasi matematika
3. **Numpy** : Untuk operasi vektor dan matriks
4. **Sys** : Untuk mengakses konfigurasi interpreter pada saat runtime dan berinteraksi dengan environment sistem operasi.

Implementation: Environment Setup

```
ROW = 6
COLUMN = 7
SQUARE_PX = 100
RAD = int(SQUARE_PX / 4) + 20

h = 200
w = 150

height = 200 + SQUARE_PX * (ROW + 1)
width = 150 + SQUARE_PX * COLUMN

surface = pygame.display.set_mode((1000, 1000))

pygame.display.set_caption("Connect Four!")

# color palette
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
C_BOARD = (25, 42, 81)
C_BG = (239, 235, 241)
C_BALLA = (245, 65, 183)
C_BALLB = (157, 134, 222)
```

- Baris pada board berjumlah 6 dan kolom berjumlah 7
- Baris dan kolom diproyeksikan ke board dengan dikalikan SQUARE_PX
- Height dan weight adalah ukuran board
- Surface (app window) berukuran 1000 x 1000
- h dan w adalah konstanta yang ditambahkan ke tiap pemanggilan/pengaksesan baris atau kolom agar board terletak di tengah-tengah surface

Implementation: Board Initialization

```
def createboard():  
    board = np.full((ROW, COLUMN), -1)  
    return board
```

```
def drawboard(board):  
    for c in range(COLUMN):  
        for r in range(ROW):  
            # Rect(left, top, width, height)  
            pygame.draw.rect(surface, C_BOARD, pygame.Rect(w + c * SQUARE_PX, h + r * SQUARE_PX + SQUARE_PX, SQUARE_PX, SQUARE_PX))  
            pygame.draw.circle(surface, WHITE, (int(w + c * SQUARE_PX + SQUARE_PX/2), int(h + r * SQUARE_PX + SQUARE_PX + SQUARE_PX/2)), RAD)  
  
    for c in range(COLUMN):  
        for r in range(ROW):  
            if board[r][c] == 0:  
                pygame.draw.circle(surface, C_BALLA, (int(w + c * SQUARE_PX + SQUARE_PX/2), height - int(h + r * SQUARE_PX + SQUARE_PX/2)), RAD)  
            elif board[r][c] == 1:  
                pygame.draw.circle(surface, C_BALLB, (int(w + c * SQUARE_PX + SQUARE_PX/2), height - int(h + r * SQUARE_PX + SQUARE_PX/2)), RAD)  
  
    pygame.display.update()
```

```
def printboard(board):  
    print(np.flip(board, 0))
```

Implementation: Winning Conditions

```
def is_winning(board, playerid):  
    for c in range(COLUMN - 3):  
        for r in range(ROW):  
            if board[r][c] == playerid and board[r][c+1] == playerid and board[r][c+2] == playerid and board[r][c+3] == playerid:  
                return True  
  
    for c in range(COLUMN):  
        for r in range(ROW - 3):  
            if board[r][c] == playerid and board[r+1][c] == playerid and board[r+2][c] == playerid and board[r+3][c] == playerid:  
                return True  
  
    for c in range(COLUMN - 3):  
        for r in range(ROW - 3):  
            if board[r][c] == playerid and board[r+1][c+1] == playerid and board[r+2][c+2] == playerid and board[r+3][c+3] == playerid:  
                return True  
  
    for c in range(COLUMN - 3):  
        for r in range(3, ROW):  
            if board[r][c] == playerid and board[r-1][c+1] == playerid and board[r-2][c+2] == playerid and board[r-3][c+3] == playerid:  
                return True  
  
    return False
```

4 horizontal

4 vertikal

4 diagonal
positif

4 diagonal
negatif

Player akan dinyatakan menang jika berhasil membuat 4 bola sejajar secara vertikal atau horizontal atau diagonal

Implementation: Game Play

```
while not gameover:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()

        # mouse movement listener
        if event.type == pygame.MOUSEMOTION and event.pos[0] < (1000-w-RAD) and event.pos[0] > w+RAD:
            pygame.draw.rect(surface, WHITE, pygame.Rect(w, h, width - w, SQUARE_PX))
            posix = event.pos[0]
            print(posix)

            if turn == 0:
                pygame.draw.circle(surface, C_BALLA, (posix, h + int(SQUARE_PX/2)), RAD)
            else:
                pygame.draw.circle(surface, C_BALLB, (posix, h + int(SQUARE_PX/2)), RAD)

    pygame.display.update()
```

Selama belum gameover maka

1. Menggunakan **event** untuk menerima semua input dari user
2. Jika mendeteksi bahwa player meng-klik tombol exit, maka window game akan tertutup
3. Menggunakan **.MOUSEMOTION** untuk mendeteksi pergerakan dari mouse (mendapat input, posisi, dan posisi-relatif)
4. Menggambar circle (untuk bola) yang warnanya berubah sesuai giliran pemain

Implementation: Game Play

```
if event.type == pygame.MOUSEBUTTONDOWN:
    pygame.draw.rect(surface, BLACK, (w, h, width, SQUARE_PX))

    # player 1
    if turn == 0:
        posix = event.pos[0]
        column = int(math.floor((posix - w) / (SQUARE_PX)))

        if check_location(board, column):
            row = next_valid_row(board, column)
            drop_ball(board, row, column, 0)

            if is_winning(board, 0):
                label = font.render("Player 1 Wins", 1, BLACK)
                surface.blit(label, (40, 10))
                gameover = True

    # player 2
    else:
        posix = event.pos[0]
        column = int(math.floor((posix - w) / SQUARE_PX))

        if check_location(board, column):
            row = next_valid_row(board, column)
            drop_ball(board, row, column, 1)

            if is_winning(board, 1):
                label = font.render("Player 2 Wins", 1, BLACK)
                surface.blit(label, (40, 10))
                gameover = True
```

1. Menggunakan **.MOUSEBUTTONDOWN** untuk mengetahui di kolom mana bola diletakkan (posisi mouse click)
2. Ketika giliran player 1, maka setelah mendapat di mana kolom bola ingin diletakkan, program akan mengecek apakah kolom tersebut valid (karena ada kemungkinan kolom tersebut sudah penuh)
3. Jika kolom valid, maka bola akan diletakkan di baris paling bawah yang masih kosong dari kolom tersebut
4. Setelah itu, cek apakah player 1 sudah menang atau belum. Jika menang, maka gameover akan di-set menjadi true dan game akan selesai
5. Jika player 1 belum menang, maka variabel turn akan di-update sehingga sekarang menjadi giliran player 2
6. Saat giliran player 2, prosesnya akan sama seperti player 1

Further Developments

1. Fitur **Main Menu** untuk memilih jenis board
 - a. Board 6 x 7
 - b. Board 5 x 6
2. **AI opponent** yang membuat move berdasarkan metode Alpha-Beta Pruning