MiniProject Phase II Overview :

Name : Aakash Shankar Prasad
Roll : 2201CS01
Course : CS210
Course Instructor : Dr . Jimson Mathew

The following is the link to the G-drive for the same video demonstration submitted:
miniProjectPhaseII.mp4

➢ Story Line

- The storyline is simple , two people are bored and play a game of tic tac toe , the winner has to give a treat , one of them loses two gams in a row ! Now he thinks that, it is unfair and hence , there must still be a way for him to win , for this game to be fair

➢ Rules of the game

- Two players play three rounds of Tic-Tac-Toe with a “bet” on each game
- There are three levels of bet:
    - Normal → normal play with +1/-1/0 for a win/loss/draw
    - Bet → bet for +3/-2/-1 for a win/loss/draw
    - Blind → higher bet for +5/-3/-2 for a win/loss/draw

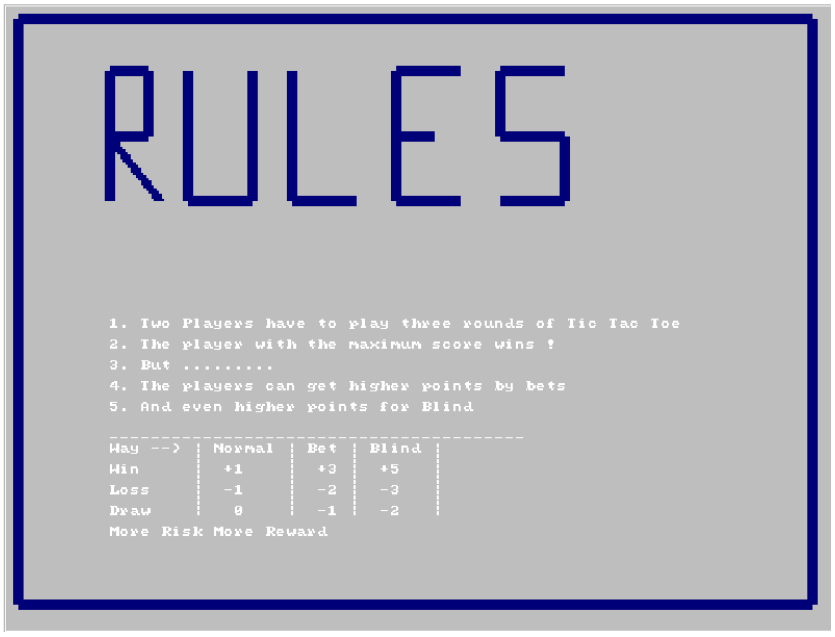
- *Motto of the game* - **High Risks High Rewards**

➢ Flow of the code

- The game opens with the Welcome Page with ASCII ART on it

- The following buttons control the flow for the player
  - Push Button → to move the objects around
  - Switches → to decide the selection of object and , in other cases decide the direction of flow of the object

- The welcome page is followed by the rules page which has the rules and the score table updated

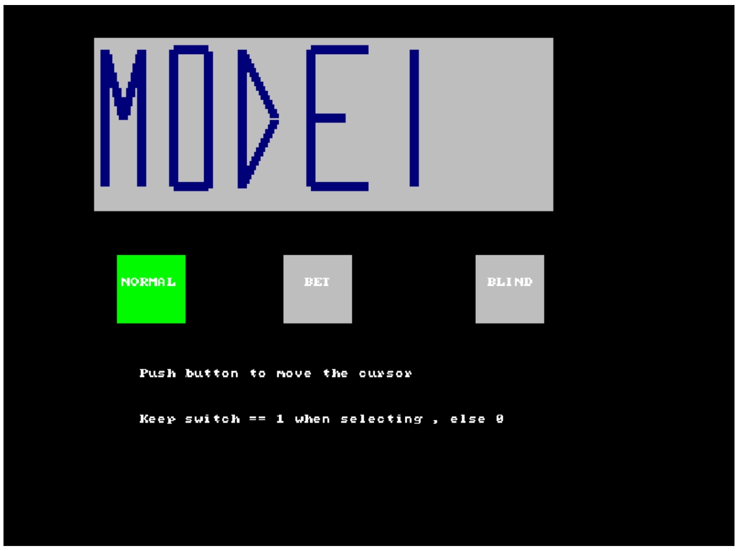


RULES

```
1. Two Players have to play three rounds of Tic Tac Toe
2. The player with the maximum score wins !
3. But .........
4. The players can get higher points by bets
5. And even higher points for Blind

---------------------------------------------
Way -->  | Normal | Bet | Blind |
Win      |  +1    |  +3 |  +5   |
Loss     |  -1    |  -2 |  -3   |
Draw     |   0    |  -1 |  -2   |
More Risk More Reward
```

  - This is followed by three rounds of Tic Tac Toe with the player choosing the mode of play before every player and deciding the flow of the play



Push button to move the cursor

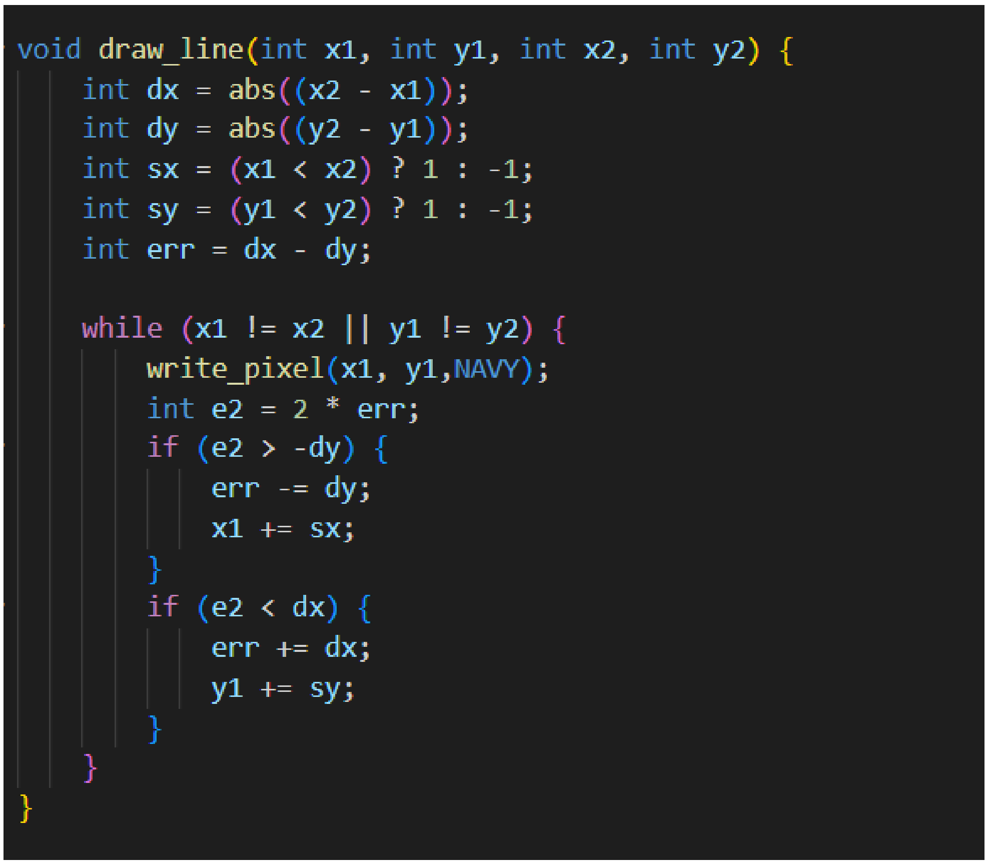
Keep switch == 1 when selecting , else 0

- The player with the highest score at the end of all the three rounds is the winner !

➢ Algorithms Used:

- Draw Line
- Drawing Thick lines
- Drawing circles
- Waiting for Input (BusyWait?!)

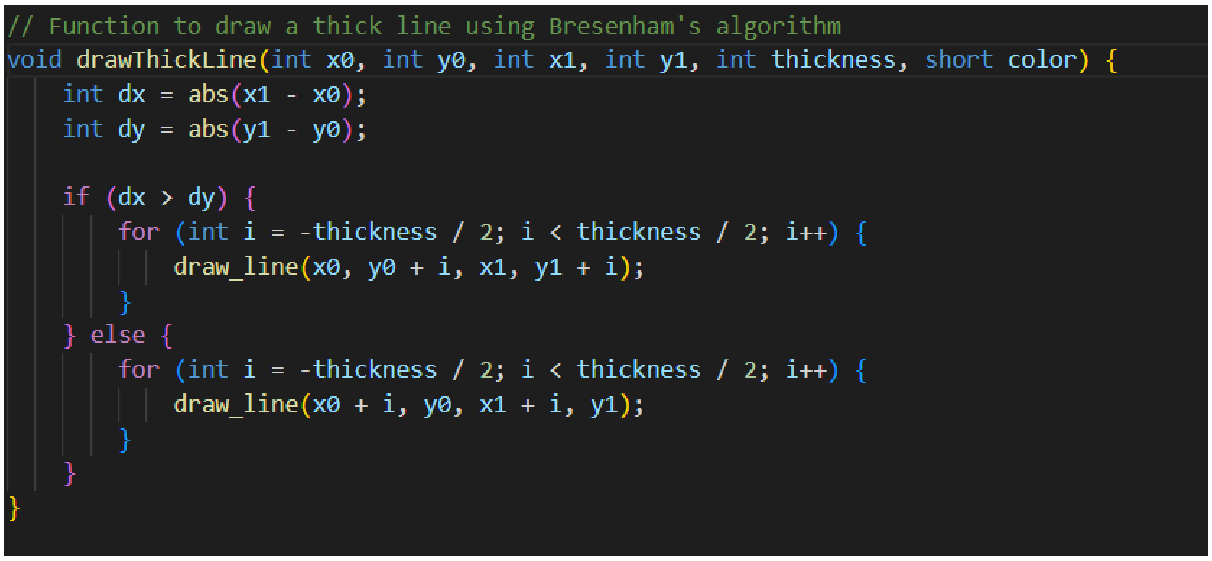
1. Bresman's Algorithm for drawing a line

a. 

```
void draw_line(int x1, int y1, int x2, int y2) {
    int dx = abs((x2 - x1));
    int dy = abs((y2 - y1));
    int sx = (x1 < x2) ? 1 : -1;
    int sy = (y1 < y2) ? 1 : -1;
    int err = dx - dy;

    while (x1 != x2 || y1 != y2) {
        write_pixel(x1, y1,NAVY);
        int e2 = 2 * err;
        if (e2 > -dy) {
            err -= dy;
            x1 += sx;
        }
        if (e2 < dx) {
            err += dx;
            y1 += sy;
        }
    }
}
```

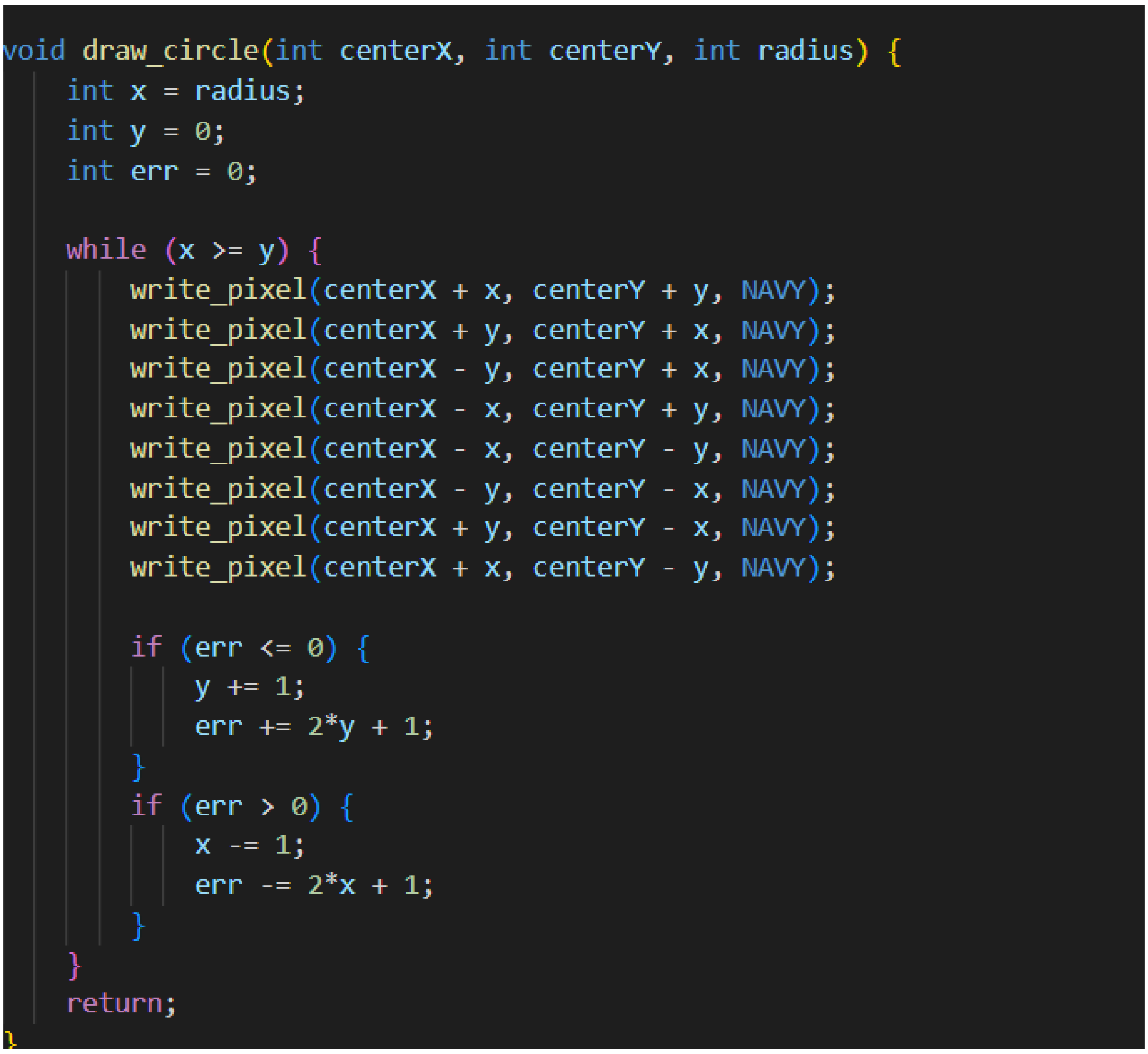
2. Bresman's Algorithm for drawing a thick Line

a. 

```
// Function to draw a thick line using Bresenham's algorithm
void drawThickLine(int x0, int y0, int x1, int y1, int thickness, short color) {
    int dx = abs(x1 - x0);
    int dy = abs(y1 - y0);

    if (dx > dy) {
        for (int i = -thickness / 2; i < thickness / 2; i++) {
            draw_line(x0, y0 + i, x1, y1 + i);
        }
    } else {
        for (int i = -thickness / 2; i < thickness / 2; i++) {
            draw_line(x0 + i, y0, x1 + i, y1);
        }
    }
}
```

3. Besman's Algorithm to draw a circle

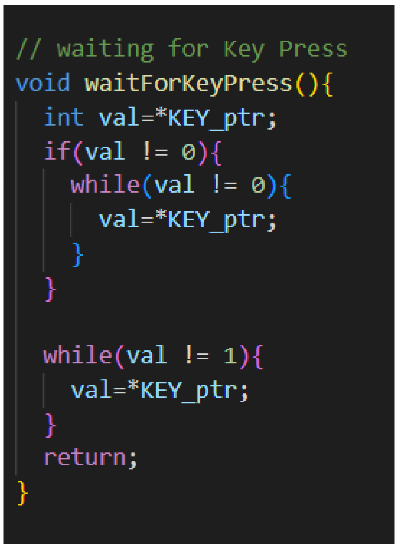
a. 

```
void draw_circle(int centerX, int centerY, int radius) {
    int x = radius;
    int y = 0;
    int err = 0;

    while (x >= y) {
        write_pixel(centerX + x, centerY + y, NAVY);
        write_pixel(centerX + y, centerY + x, NAVY);
        write_pixel(centerX - y, centerY + x, NAVY);
        write_pixel(centerX - x, centerY + y, NAVY);
        write_pixel(centerX - x, centerY - y, NAVY);
        write_pixel(centerX - y, centerY - x, NAVY);
        write_pixel(centerX + y, centerY - x, NAVY);
        write_pixel(centerX + x, centerY - y, NAVY);

        if (err <= 0) {
            y += 1;
            err += 2*y + 1;
        }
        if (err > 0) {
            x -= 1;
            err -= 2*x + 1;
        }
    }
    return;
}
```

4. Waiting for the desired input on the push buttons

a. 

```
// waiting for Key Press
void waitForKeyPress(){
    int val=*KEY_ptr;
    if(val != 0){
        while(val != 0){
            val=*KEY_ptr;
        }
    }

    while(val != 1){
        val=*KEY_ptr;
    }
    return;
}
```