

Welcome To:

Treat For  Cobra

A CLASSIC SNAKE GAME

TREATFORCOBRA: A CLASSIC SNAKE GAME

Presented By:

Shahria Sultana Elin

Roll no:1445

BSSE14

Supervised By:

Dr. Naushin Nower

Professor

IIT, DU

Introduction:

- ❖ “ TreatForCobra” is a classic snake game that we all have played on our “Nokia” button phones.



❖ This game will be divided into three levels:

1.Level-1:

In this Level-1 snake game, the objective is to collect food while racing against the clock and competing with a friend.

2.Level-2:

Collision detection condition: It means there will be a condition that will detect if the snake has a collision with the wall.

3.Level-3:

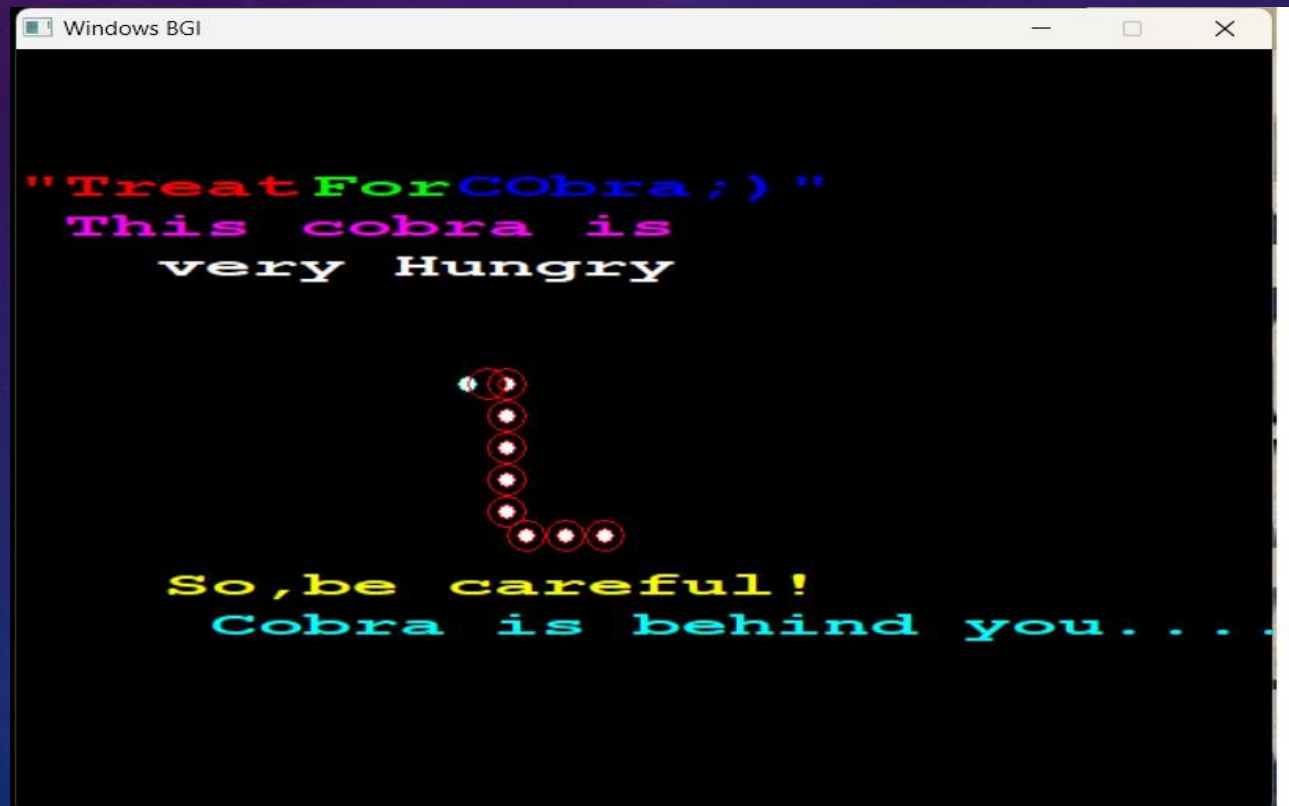
The number of difficulties will be increased in the path of the snake at this level.

❖ WORK TILL MID

LEVEL-1

FEATURES

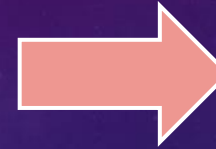
WELCOMING LOGO:



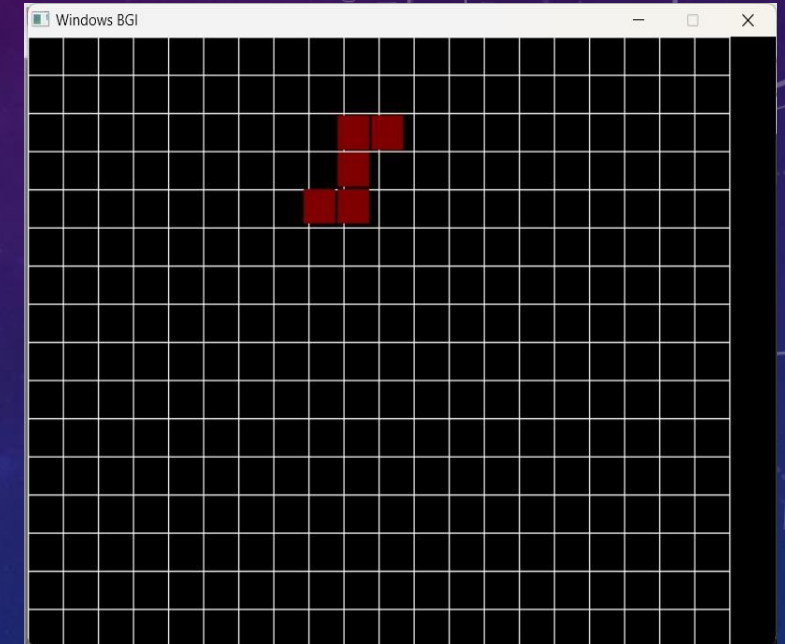
❖ Move The Snake :

Function for moving the snake:

```
182 //Move the Snake
183 void moveSnake() {
184
185     for (int i = length - 1; i > 0; i--) {
186         snake[i] = snake[i - 1];
187     }
188     if (direction == 'r') {
189         snake[0].x += CELL_SIZE-1;
190     } else if (direction == 'l') {
191         snake[0].x -= CELL_SIZE-1;
192     } else if (direction == 'u') {
193         snake[0].y -= CELL_SIZE-1;
194     } else if (direction == 'd') {
195         snake[0].y += CELL_SIZE-1;
196     }
197 }
```



Output:



Main

Function:

```
260
261 if (direction != 'f') {
262     char newDirection = getch();
263     if (newDirection == 'r' && direction != 'l') {
264         direction = 'r';
265     } else if (newDirection == 'l' && direction != 'r') {
266         direction = 'l';
267     } else if (newDirection == 'u' && direction != 'd') {
268         direction = 'u';
269     } else if (newDirection == 'd' && direction != 'u') {
270         direction = 'd';
271     }
272
273
274
275
```

❖ Placing the Food:

In the snake game, making sure the food appears in different places each time is important. To do this, I have used a tool called a random number generator. Once the Snake eats the food (meaning it moves over the food), we increase the Snake's size and score. After that, we need to choose a new random spot on the grid to place the next piece of food.

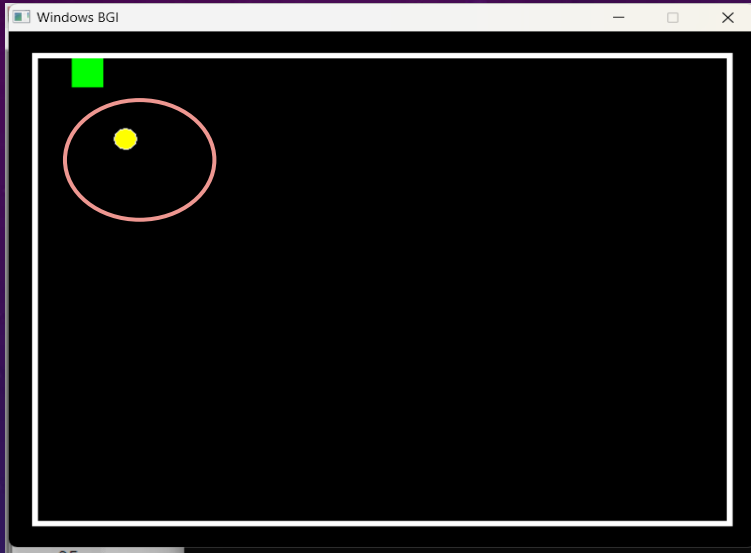
Function for placing the food:

```
205 void drawFood(){
206     srand(123); // Use a fixed seed for consistent results
207
208     int maxX = getmaxx();
209     int maxY = getmaxy();
210
211
212
213     setfillstyle(SOLID_FILL, YELLOW);
214     circle(food.x, food.y, 10);
215     floodfill(food.x, food.y, WHITE); // Fill the circle with the current fill style
216
217
218
219
220
221 }
222 void foodrand(){
223
224     srand(123);
225
226     int maxX = getmaxx();
227     int maxY = getmaxy();
228
229     //int radius = 50;
230     food.x = rand() % (maxX - 2 * 10) + 10;
231     food.y = rand() % (maxY - 2 * 10) + 10;
232
233
234 }
```

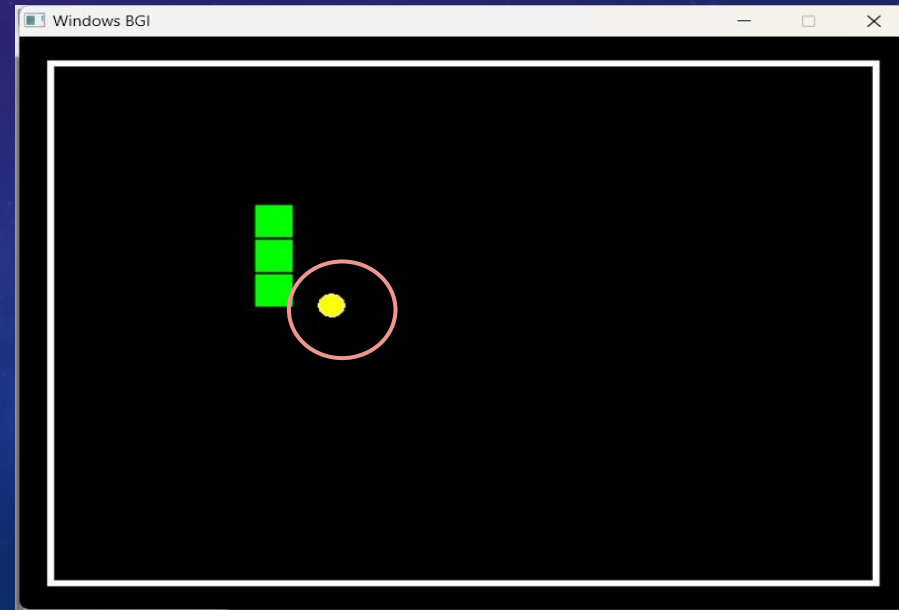
```
174 void drawSnake() {
175     for (int i = 0; i < length; i++) {
176         setfillstyle(SOLID_FILL, RED);
177         bar(snake[i].x, snake[i].y, snake[i].x + (CELL_SIZE-3), snake[i].y + (CELL_SIZE-3));
178
179         if((snake[i].x+snake[i].x + (CELL_SIZE-3))/2>=(food.x)&&(snake[i].y+snake[i].y+ (CELL_SIZE-3))/2>=(food.y+50)){
180             length++;
181             records[numofplayer].score+=100;
182             foodrand();
183         }
184     }
185 }
```



Before consuming food :



After Consuming food:



Before consuming food:

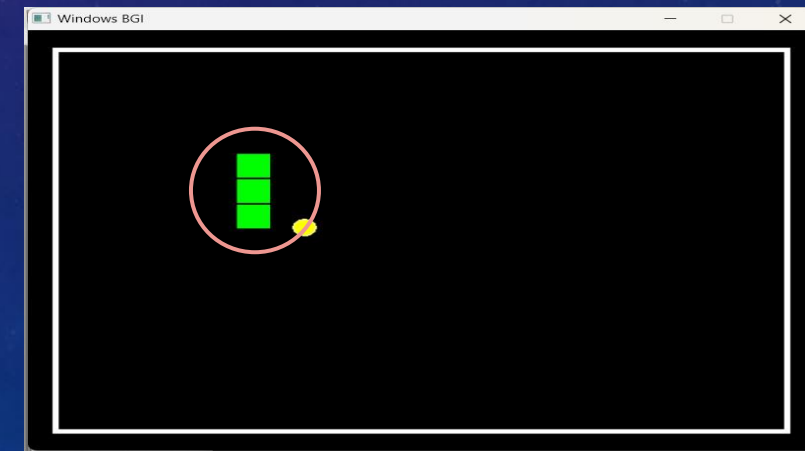


❖ INCREASING THE LENGTH OF THE SNAKE AFTER CONSUMING THE FOOD:

Function:

```
154
155 //Snake
156 void drawSnake() {
157
158     for (int i = 0; i < length; i++) {
159         int color=COLOR(0,255,0);
160         setfillstyle(SOLID_FILL, color);
161         bar(snake[i].x, snake[i].y, snake[i].x + (CELL_SIZE-3), snake[i].y + (CELL_SIZE-3));
162         double len=(snake[i].x+snake[i].x + (CELL_SIZE-3))/2;
163         double width=(snake[i].y+snake[i].y+ (CELL_SIZE-3))/2;
164
165
166
167
168
169         if((len>=food.x)&&(width>=food.y)){
170             length++;
171             food.x+=50;
172             food.y+=50;
173             records[numofplayer].score+=100;
174
175         }
176
177
178
179
180     }
181 }
```

After consuming food:



❖ TAKING USER NAME & RECORDING SCORE:

User :

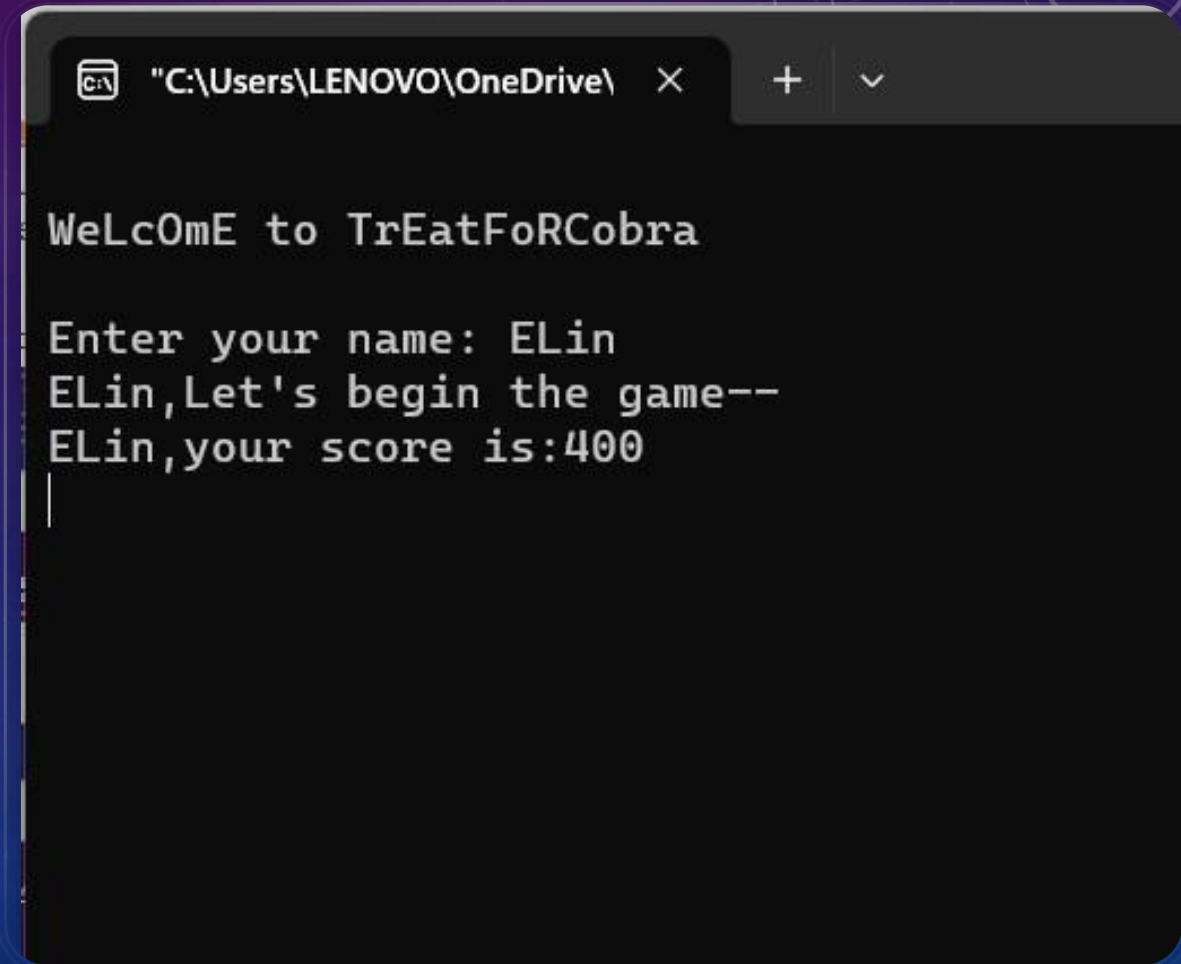
```
127
128 //User
129 void user(){
130     numofplayer++;
131     printf(" \nWeLcOmE to TrEatFoRCobra\n") ;
132     // printf("Enter Time Limit(In seconds): ");
133     //scanf("%d",&time_limit);
134     printf("\nEnter your name: ");
135     scanf("%s", records[numofplayer].playerName);
136 }
137
```

Score:

```
//Snake
void drawSnake() {
    for (int i = 0; i < length; i++) {
        int color=COLOR(0,255,0);
        setfillstyle(SOLID_FILL, color);
        bar(snake[i].x, snake[i].y, snake[i].x + (CELL_SIZE-3), snake[i].y + (CELL_SIZE-3));
        double len=(snake[i].x+snake[i].x + (CELL_SIZE-3))/2;
        double width=(snake[i].y+snake[i].y+ (CELL_SIZE-3))/2;

        if((len>=food.x)&&(width>=food.y)){
            length++;
            food.x+=50;
            food.y+=50;
            records[numofplayer].score+=100;
        }
    }
}
```

Output:




```
"C:\Users\LENOVO\OneDrive\
WeLcOmE to TrEatFoRCobra
Enter your name: Elin
Elin,Let's begin the game--
Elin,your score is:400
```

❖ TimeLimit:

Logic:

```
280
281
282     }
283     //Time
284     time(&currentTime);
285     double Time = difftime(currentTime, startTime);
286
287     if (Time >= time_limit) {
288
289         break;
290     }
291
292 }
```

Output:




```
Time is up!
Congratulations!
```

❖ Winning Condition:

Logic:

```
384
385     printf("\n%s, your score is: %d\n", records[numofplayer].playerName, records[numofplayer].score);
386
387
388
389     if(records[1].score > records[2].score){
390         printf("Winner: %s", records[1].playerName);
391     }
392     else if(records[1].score == records[2].score) {
393         printf("Draw");
394     }
395     else printf("Winner: %s", records[2].playerName);
396
397
398     return 0;
399 }
400
401
```



```
"C:\Users\LENOVO\OneDrive\ x + v

WeLcOmE to TrEatFoRCobra

Enter your name: Elin
Elin,Let's begin the game--
Elin,your score is:0

PLAYER-2

WeLcOmE to TrEatFoRCobra

Enter your name: Shahria
Shahria,Let's begin the game--
Shahria,your score is:0
Draw
Process returned 0 (0x0)   execution time : 33.131 s
Press any key to continue.
|
```

Remaining Works:

Level-2:

Collision detection condition: It means there will be a condition that will detect if the snake has a collision with the wall.

Level-3:

The number of difficulties will be increased in the path of the snake at this level.

My GitHub Link:

<https://github.com/SultanaElin/SPL-1-MID>

A small simulation of level-1:



