PROJECT- SNAKE GAME (ANSHU RAJ)

```
#include <iostream>
#include <windows.h>
#include <cstdlib>
#include <ctime>
#include <stack>
using namespace std;

class Move
{
public:
 int x, y;
 Move(int a, int b)
 {
 x = a;
 y = b;
 }
};
stack<Move> snake;

void printArr(char arr[10][10], int row, int col)
{
 system("cls");
```

```
 cout << endl;

 cout << endl;
 for (int i = 0; i < row; i++)

 {

 for (int j = 0; j < col; j++)

 {

 cout << arr[i][j] << " ";

 }

 cout << endl;

 }

 cout << endl;

 cout << endl;

}


void updateSnake(char arr[10][10], int &snakeX, int &snakeY, int &pX, int &pY)

{


 int x = abs(snakeX - pX);

 int y = abs(snakeY - pY);


 if (x > y)

 {

 if (snakeX > pX)

 {

 if (arr[snakeX - 1][snakeY] != 'O' && arr[snakeX - 1][snakeY] != '#' && arr[snakeX - 1][snakeY] !=
```

```
'$')

{

arr[snakeX][snakeY] = ' ';
snakeX--;

arr[snakeX][snakeY] = '~';

}

else if (arr[snakeX - 1][snakeY] == 'O')

{

arr[snakeX][snakeY] = ' ';

snakeX++;

arr[snakeX][snakeY] = '~';

}

}

else

{

if (arr[snakeX + 1][snakeY] != 'O' && arr[snakeX + 1][snakeY] != '#' && arr[snakeX +  1][snakeY] !=
'$')

{

arr[snakeX][snakeY] = ' ';

snakeX++;

arr[snakeX][snakeY] = '~';

}

else if (arr[snakeX + 1][snakeY] == 'O')

{

arr[snakeX][snakeY] = ' ';

snakeX--;
```

```
arr[snakeX][snakeY] = '~';

}

}
}

else

{

if (snakeY > pY)

{

if (arr[snakeX][snakeY - 1] != 'O' && arr[snakeX][snakeY - 1] != '#' && arr[snakeX][snakeY - 1] != '$')

{

arr[snakeX][snakeY] = ' ';

snakeY--;

arr[snakeX][snakeY] = '~';

}

else if (arr[snakeX][snakeY - 1] == 'O')

{

arr[snakeX][snakeY] = ' ';

snakeY++;

arr[snakeX][snakeY] = '~';

}

}

else

{

if (arr[snakeX][snakeY + 1] != 'O' && arr[snakeX][snakeY + 1] != '#' && arr[snakeX][snakeY  + 1] != '$')

{
```

```
arr[snakeX][snakeY] = ' ';

snakeY++;
arr[snakeX][snakeY] = '~';

}

else if (arr[snakeX][snakeY + 1] == 'O')

{

arr[snakeX][snakeY] = ' ';

snakeY--;

arr[snakeX][snakeY] = '~';

}

}

}

Move m(snakeX, snakeY);

snake.push(m);

}

int checkStatus(char arr[10][10], int snakeX, int snakeY, int pX, int pY, int winX, int winY)

{

if (pX == snakeX && pY == snakeY)

{



return 0;

}

else if (pX == winX && pY == winY)

{

return 1;
```

```
 }

 return -1;
}


int main()

{


 srand(static_cast<unsigned int>(time(nullptr)));  int

row = 10, col = 10;

 char arr[10][10];


 int obstacles = 5;

 cout << "Enter Total obstacles : ";

 cin >> obstacles;

 int pX = rand() % 8 + 1, pY = rand() % 8 + 1;  int

snakeX = 8, snakeY = 1;

 int winX = 8, winY = 5;

 while (1)

 {

 snakeX = rand() % 8 + 1;

 snakeY = rand() % 8 + 1;

 if (snakeX != pX && snakeY != pY)  {

 break;

 }

 }
```

```
while (1)
{

winX = rand() % 8 + 1;

winY = rand() % 8 + 1;

if ((winX != pX && winY != pY) && (winX != snakeX && winY != snakeY))  {

break;

}

}



Move m(snakeX, snakeY);

snake.push(m);



for (int i = 0; i < obstacles; i++)

{

int x = rand() % 8 + 1;

int y = rand() % 8 + 1;

if ((x == snakeX && y == snakeY) || (x == winX && y == winY) || (x == pX && y == pY))  {

i--;

continue;

}

arr[x][y] = 'O';

}



for (int i = 0; i < row; i++)
{
```

```
for (int j = 0; j < col; j++)

{

if (i == 0 || j == col - 1 || i == row - 1 || j == 0)  {

arr[i][j] = '#';

}

else if (arr[i][j] != 'O')

{


if (i == pX && j == pY)

{

arr[i][j] = '>';

}

else if (i == snakeX && j == snakeY)  {

arr[i][j] = '~';

}

else if (i == winX && j == winY)  {

arr[i][j] = '$';

}


else

{
arr[i][j] = ' ';  }

}

}

}
```

```
int flag = 1;


int undoCount = 0;

stack<char> s;

while (flag)

{


printArr(arr, row, col);

char m;

cout << "W A S D to play" << endl;  cout <<

"E to exit" << endl;  if (!s.empty())

{


cout << "U to Undo" << endl;  }

cout << "Enter :";

cin >> m;
if (m == 'E' || m == 'e')  {

break;

}


if (m == 'U' || m == 'u')  {

if (s.empty())

{

continue;
```

```
}

char ch = s.top();  s.pop();

if (undoCount == 0)  {

snake.pop();  undoCount++;  }


if (ch == 'W' || ch == 'w')  {

arr[pX][pY] = ' ';  pX = pX + 1;


arr[pX][pY] = '^';
}

else if (ch == 'A' || ch == 'a')  {

arr[pX][pY] = ' ';  pY = pY + 1;


arr[pX][pY] = '<';  }

else if (ch == 'S' || ch == 's')  {

arr[pX][pY] = ' ';  pX = pX - 1;


arr[pX][pY] = 'v';  }

else if (ch == 'D' || ch == 'd')  {

arr[pX][pY] = ' ';  pY = pY - 1;


arr[pX][pY] = '>';  }

Move m = snake.top();
```

```
arr[snakeX][snakeY] = ' ';  snakeX = m.x,
snakeY = m.y;
 arr[snakeX][snakeY] = '~';
 snake.pop();
 continue;
 }

 if (m == 'D' || m == 'd')
 {

 if (pY + 1 >= col - 1 || arr[pX][pY + 1] == 'O')
 {

 updateSnake(arr, snakeX, snakeY, pX, pY);
 int status = checkStatus(arr, snakeX, snakeY, pX, pY, winX, winY);  if (status == 0)
 {
 cout << "You loose";
 break;
 }
 else if (status == 1)
 {
 cout << "You win!";
 break;
 }
 else
```

```
{
continue;

}

}

arr[pX][pY] = ' ';

pY = pY + 1;


arr[pX][pY] = '>';

s.push(m);

}


else if (m == 'w' || m == 'W')

{


if (pX - 1 <= 0 || arr[pX - 1][pY] == 'O')

{

updateSnake(arr, snakeX, snakeY, pX, pY);

int status = checkStatus(arr, snakeX, snakeY, pX, pY, winX, winY);  if (status == 0)

{

cout << "You loose";

break;

}

else if (status == 1)

{

cout << "You win!";
```

```
break;
}
else
{
continue;
}
}
arr[pX][pY] = ' ';
pX = pX - 1;

arr[pX][pY] = '^';
s.push(m);
}
else if (m == 's' || m == 'S')
{

if (pX + 1 >= row - 1 || arr[pX + 1][pY] == 'O')
{
updateSnake(arr, snakeX, snakeY, pX, pY);
int status = checkStatus(arr, snakeX, snakeY, pX, pY, winX, winY);  if (status == 0)
{
cout << "You loose";
break;
}
else if (status == 1)
```

```
{
cout << "You win!";
break;
}
else
{
continue;
}
}
arr[pX][pY] = ' ';
pX = pX + 1;

arr[pX][pY] = 'v';
s.push(m);
}
else if (m == 'a' || m == 'A')
{

if (pY - 1 <= 0 || arr[pX][pY - 1] == 'O')
{
updateSnake(arr, snakeX, snakeY, pX, pY);
int status = checkStatus(arr, snakeX, snakeY, pX, pY, winX, winY);  if (status == 0)
{
cout << "You loose";
break;
```

```
}
else if (status == 1)
{
cout << "You win!";
break;
}
else
{
continue;
}
}
arr[pX][pY] = ' ';
pY = pY - 1;

arr[pX][pY] = '<';
s.push(m);
}
int status1 = checkStatus(arr, snakeX, snakeY, pX, pY, winX, winY);  if
(status1 == 0)
{
cout << "You loose!!";
break;
}
else if (status1 == 1)
{
```

```
 cout << "You win!";
 break;
 }
 updateSnake(arr, snakeX, snakeY, pX, pY);
 int status = checkStatus(arr, snakeX, snakeY, pX, pY, winX, winY);  if (status
== 0)
 {
 cout << "You loose!!";
 break;
 }
 else if (status == 1)
 {
 cout << "You win!";
 break;
 }
 }
}
```