



INSTITUTO TECNOLÓGICO DE OAXACA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA:

GRAFICACIÓN

REPORTE DE EVALUACIÓN

INTEGRANTES:

CARRILLO CHAGOYA KEVIN GIOVANNY
HERNANDEZ JIMENEZ ARTURO

DOCENTE:

HERNANDEZ ISLAS ADRIAN

GRUPO:

6SA

No° SEMESTRE:

6

PERIODO:

ENERO/JUNIO 2022



Contenido:

| | |
|---|---|
| Descripción del Juego De La Serpiente | 2 |
| Lenguaje e IDE empleado | 2 |
| Herramientas graficas usadas: | 2 |
| Funciones Implementadas | 3 |
| <input type="checkbox"/> Función main..... | 3 |
| <input type="checkbox"/> Función Display ()..... | 4 |
| <input type="checkbox"/> Función dibujarSnake() | 4 |
| <input type="checkbox"/> Función coordenadasComida()..... | 5 |
| <input type="checkbox"/> Función dibujarComida() | 6 |
| <input type="checkbox"/> Función puntuacionSnake()..... | 6 |
| <input type="checkbox"/> Función movimientoSnake() | 7 |
| Manual De Usuario | 8 |
| <input type="checkbox"/> Interfaz | 8 |
| <input type="checkbox"/> Reglas De Juego | 9 |

Descripción del Juego De La Serpiente

El Snake (a veces también llamado serpiente) es un videojuego lanzado a mediados de la década de 1970 que ha mantenido su popularidad desde entonces, convirtiéndose en un juego clásico y claro está que el objetivo de este proyecto, fue el de implementar dicho juego de la serpiente, en algún lenguaje en específico que nosotros eligiéramos, pero usando una herramienta grafica.

Lenguaje e IDE empleado

Para la implementación del juego de la serpiente decimos utilizar el lenguaje de programación C++ con ayuda del entorno de desarrollo Dev C++.

Herramientas graficas usadas:

Para poder llevar a cabo la implementación de este juego en el lenguaje C++ hemos usado las herramientas graficas que nos brindan las librerías de Open GL y GLUT. Dichas librerías fueron instaladas en nuestro IDE Dev C++ para poder ser utilizadas.

Funciones Implementadas

➤ Función main

En nuestra función main, básicamente se realizó la llamada de unas funciones de la librería OpenGL y Glut, la función glutInit() la utilizamos para inicializar la biblioteca Glut, a continuación tenemos la función glutInitWindowSize() con la cual inicializamos el tamaño de la ventana inicial que en nuestro caso es de 600x600, de ahí con la función glutInitWindowPosition establecimos la posición de la ventana, posteriormente tenemos la cadena de caracteres que sería el título de nuestra ventana el cual posteriormente se lo pasamos como parámetro a la función glutCreateWindow(), luego con la función glutDisplayFunc() determinamos todo lo que se mostrara en nuestra ventana que en este caso se encuentra en la función display(), después tenemos la función de remodelación de ventana que es glutReshapeFunc(), luego para el uso de las teclas utilizamos glutSpecialFunc() y de ahí realizamos la llamada del temporizador con la función glutTimerFunc() a la cual le establecimos una velocidad de 100 ms, finalmente tenemos la constante del tamaño inicial de nuestra serpiente y seguidamente especificamos las coordenadas iniciales del cuerpo de nuestra serpiente para lo cual las vamos guardando en una cola que se estará utilizando.

```
//Funcion Principal
int main(int argc, char** argv){
    glutInit(&argc, argv); //iniciamos OpenGL
    glutInitWindowSize(600, 600); //tamaño de ventana
    glutInitWindowPosition(450, 80);
    //Titulo de la ventana
    const char titulo[] = "Examen Unidad II: Juego de la Serpiente";
    glutCreateWindow(titulo); //pasamos el titulo de nuestra ventana
    glutDisplayFunc(display); //visualización de las funciones para la ventana
    glutReshapeFunc(reshape); //remodelación para la ventana
    glutSpecialFunc(teclasOprimidas); //llamada del teclado para la ventana
    glutTimerFunc(velocidad, moverSnake, 0); //llamada del temporizador que se activará en ms
    //Tamaño inicial de nuestra serpiente
    const int tamanoInicialSnake = 3;
    // Especificamos las coordenadas de cada parte de la serpiente.
    for(int a = 1; a <= tamanoInicialSnake; a++){
        std::deque<double> fila;
        fila.push_back(0.0);
        fila.push_back((longitud + 2.0 + (tamanoInicialSnake * 2)) - (a * 2));
        cuerpoSnake.push_front(fila);
    }
    srand(time(NULL));
    initGL();
    glutMainLoop();
    return 0;
}
```

➤ Función Display ()

En esta función únicamente se manda a llamar otras funciones que serán mostradas en toda en la ventana y que son parte de la lógica de nuestro juego como la de dibujar nuestra serpiente en ventana, la de dar coordenadas random a la comida y dibujarla y finalmente la de mostrar la puntuación.

```
//Funcion para mostrar todo en la ventana
void display(){
    //Limpiamos la superficie de dibujo
    glClear(GL_COLOR_BUFFER_BIT);
    //Matriz de cámara
    glMatrixMode(GL_MODELVIEW);
    //llamada a la funcion de dibujar serpiente
    dibujarSnake();
    //llamada a la funcion de generar y dibujar comida
    coordenadasComida();
    dibujarComida();
    //llamada a la funcion de mostrar puntuaciones
    puntuacionSnake();
    //intercambiamos los búferes de la ventana
    glutSwapBuffers();
}
```

➤ Función dibujarSnake()

En esta función básicamente lo que se hace es el obtener las coordenadas del cuerpo de nuestra serpiente almacenadas en una cola y una vez obtenidas ir graficando en forma de polígono.

```
//Funcion de dibujar el cuerpo de la Serpiente
void dibujarSnake(){
    // pintamos las coordenadas del cuerpo de la serpiente
    glLoadIdentity();
    glTranslated(cuerpoSnake[0][0], cuerpoSnake[0][1], -40.0);
    glColor3f(1, 1, 0); //color verde
    //Damos forma de cuadro
    glBegin(GL_POLYGON);
    glVertex2d( 1, 1);
    glVertex2d( 1, -1);
    glVertex2d(-1, -1);
    glVertex2d(-1, 1);
    glEnd();
    for(int a = 1; a < cuerpoSnake.size(); a++){
        glLoadIdentity();
        glTranslated(cuerpoSnake[a][0], cuerpoSnake[a][1], -40.0);
        glColor3f(0, 1, 1); //color verde
        glBegin(GL_POLYGON);
        glVertex2d( 1, 1);
        glVertex2d( 1, -1);
        glVertex2d(-1, -1);
        glVertex2d(-1, 1);
        glEnd();
    }
}
```

➤ Función coordenadasComida()

En esta función lo que hicimos fue generar las coordenadas de la comida de la serpiente para lo cual se tomó en cuenta que si las coordenadas generadas para la comida ya están siendo ocupadas por el cuerpo de la serpiente se vuelva a repetir la generación de las coordenadas y finalmente las almacenamos en nuestro arreglo llamado coordComida.

```
//Funcion de dibujar comida
void coordenadasComida(){
    //Si hay comida dibujada no dibuja
    if(hayComida){
        return;
    }
    //arreglo de coordenadas temporales de la comida
    int coord_temp_comida[2];
    //variable choque
    bool choqueCoordenadas;
    //do while
    do {
        choqueCoordenadas = false;
        //Generamos nuestras coordenadas aleatorias temporales
        coord_temp_comida[0] = 2 * (rand() % ((int)(longitud + 1))) - (int)longitud;
        coord_temp_comida[1] = 2 * (rand() % ((int)(longitud + 1))) - (int)longitud;
        //coord_temp_comida[1] = 2 * (rand() % ((int)(longitud - 8))) - (int)longitud;

        // si las coordenadas de la comida generada es en un punto del cuerpo de la serpiente
        for(int a = 0; a < cuerpoSnake.size(); a++){
            //comparacion de coordenadas del cuerpo de la serpiente con la de la comida
            if(coord_temp_comida[0] == cuerpoSnake[a][0] &&
               coord_temp_comida[1] == cuerpoSnake[a][1]){
                choqueCoordenadas = true;
            }
        }
    } while(choqueCoordenadas);

    //guardamos las coordenadas temporales generadas en arreglo de coordenadas comida
    coordComida[0] = coord_temp_comida[0];
    coordComida[1] = coord_temp_comida[1];

    hayComida = true;
}
```

➤ Función dibujarComida()

En esta función lo que realizamos fue únicamente ir recuperando las coordenadas generadas para la comida y a estas les aplicamos traslación y finalmente la pintamos de color azul y la dibujamos en forma de polígono.

```
//Funcion para dibujar la comida
void dibujarComida(){
    //funcion de traslacion
    glLoadIdentity();
    glTranslatef(coordComida[0], coordComida[1], -40.0);
    traslacion(coordComida[0], coordComida[1]);
    //traslacion(coordComida[0], coordComida[1]);
    glColor3f(0, 0, 1); //pintaremos en color rojo
    //generamos nuestra comida en forma de poligono para que aparezca rellena
    glBegin(GL_POLYGON);
        glVertex2d( 1, 1);
        glVertex2d( 1, -1);
        glVertex2d(-1, -1);
        glVertex2d(-1, 1);
    glEnd();
}
```

➤ Función puntuacionSnake()

En esta función mostramos la puntuaciones en nuestra ventana con ayuda de la función drawBitmapText().

```
//Funcion para la puntuacion
void puntuacionSnake(){
    const char* msgPuntuacion = (char*) "Puntuacion del Jugador 1: ";
    const char* msgLongitud = (char*) "Longitud de la serpiente: ";
    const char* nombres = (char*) "Kevin Chagoya y Arturo Hernandez";
    //tratamos los string como un stream
    std::stringstream nombres_string;
    std::stringstream puntuacion_string;
    std::stringstream longitud_string;
    nombres_string << nombres;
    puntuacion_string << msgPuntuacion << puntuacion;
    longitud_string << msgLongitud << cuerpoSnake.size();
    //Mostramos los mensajes en la ventana
    drawBitmapText((char*) nombres_string.str().c_str(), -longitud + 15.0f, longitud - 2.0f, -39.5f);
    drawBitmapText((char*) puntuacion_string.str().c_str(), -longitud + 15.0f, longitud - 4.0f, -39.5f);
    drawBitmapText((char*) longitud_string.str().c_str(), -longitud + 15.0f, longitud - 6.0f, -39.5f);
}
```

➤ Función movimientoSnake()

En esta función lo que hicimos es saber que tecla fue oprimida (arriba, abajo, izquierda o derecha) y mediante un switch dependiendo de los casos hacemos las rotaciones correspondientes a la serpiente, así mismo en esta función implementamos el código para saber en qué momento la serpiente ingiere la comida o bien en qué momento la serpiente choca con la pared o bien con su propio cuerpo.

```
//Funcion para los movimientos de la serpiente
void movimientoSnake(int d){
    direccion = d;
    int parte_siguiente = cuerpoSnake.size() - 1;
    std::deque<double> cabezaSnake = cuerpoSnake[parte_siguiente];

    double dx = 0.0;
    double dy = 0.0;
    double borde = 0.0;
    int ejeSnake = 0;
    //Saber que direccion ir
    switch (direccion){
        case ARRIBA:{
            dy = 2.0;
            borde = longitud; //restar 10 no encimar
            ejeSnake = 1;
            break;
        }

        case ABAJO:{
            dy = -2.0;
            borde = -longitud;
            ejeSnake = 1;
            break;
        }

        case DERECHA:{
            dx = 2.0;
            borde = longitud;
            ejeSnake = 0;
            break;
        }

        case IZQUIERDA:{
            dx = -2.0;
            borde = -longitud;
            ejeSnake = 0;
            break;
        }
    }

    for(int a = 0; a < cuerpoSnake.size(); a++){
        //comparamos coordenadas para saber si son iguales
        if(cuerpoSnake[0][0] + dx == cuerpoSnake[a][0] &&
           cuerpoSnake[0][1] + dy == cuerpoSnake[a][1]){
            MessageBox(0, "GAME OVER", "Perdiste", MB_OK);
            exit(0); //salimos
        }
    }

    // si chocamos con la pared del area de juego
    if(cuerpoSnake[0][ejeSnake] == borde){
        MessageBox(0, "GAME OVER", "Perdiste", MB_OK);
        exit(0);
    }

    // Si la serpiente ingiere la comida la puntuacion y la longitud de nuestros contadores incrementara
    if(cuerpoSnake[0][0] + dx == coordComida[0] &&
       cuerpoSnake[0][1] + dy == coordComida[1]){
        puntuacion += 10; //aumentamos el contador de puntuacion en 10
        inicioCrecimiento++; //aumentamos el contador de longitud en 1
        hayComida = false; //cambiamos el estado de la comida de que no hay
    }

    cabezaSnake[0] = cuerpoSnake[0][0] + dx;
    cabezaSnake[1] = cuerpoSnake[0][1] + dy;

    cuerpoSnake.push_front(cabezaSnake);

    if(!inicioCrecimiento){
        cuerpoSnake.pop_back();
    } else if(inicioCrecimiento == crecimiento){
        inicioCrecimiento = 0;
    } else {
        inicioCrecimiento++;
    }

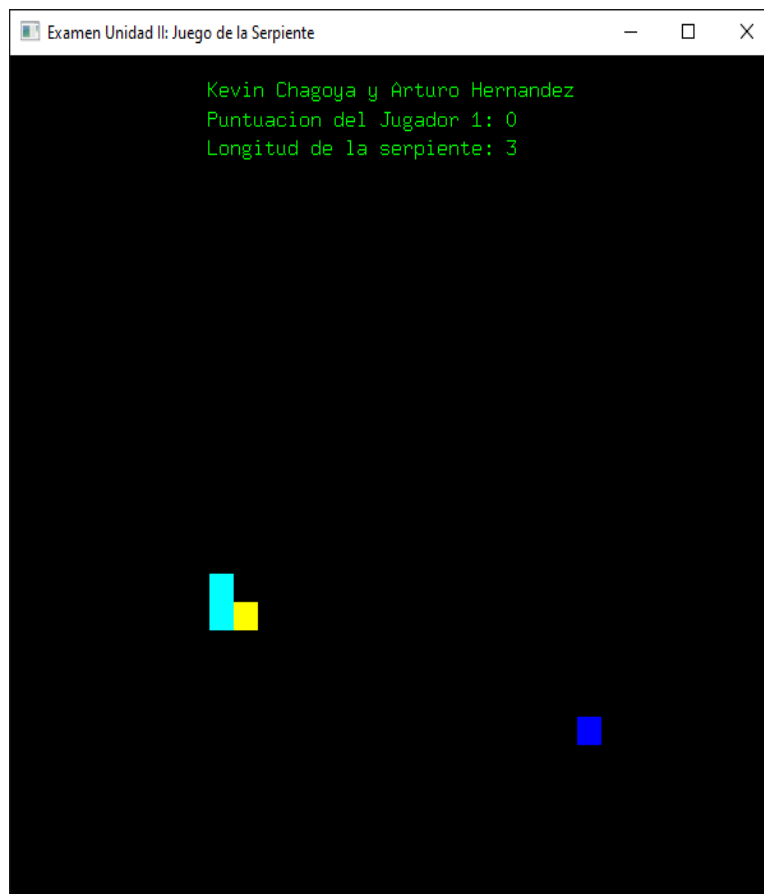
    glutPostRedisplay();
}
```


Manual De Usuario

➤ Interfaz

La ventana de la interfaz se inicializa en una dimensión de 600x600 pixeles, el cual aparece centrada en la pantalla del ordenador.

- Dicha ventana tiene por nombre “**Examen Unidad II: Juego de la Serpiente**”.
- Cuenta con un color de fondo totalmente **negro**.
- En la parte superior se encuentra el puntaje del jugador, así como la longitud de la serpiente, de color **verde**.
- La figura de la serpiente se representa por pixeles coloreados de color **azul claro** y **amarillo**, **azul** para el cuerpo y los pixeles agregados y **amarillo** para la cabeza de la serpiente.
- La comida que estará comiendo la serpiente se representa por un pixel pintado de color **azul fuerte**.



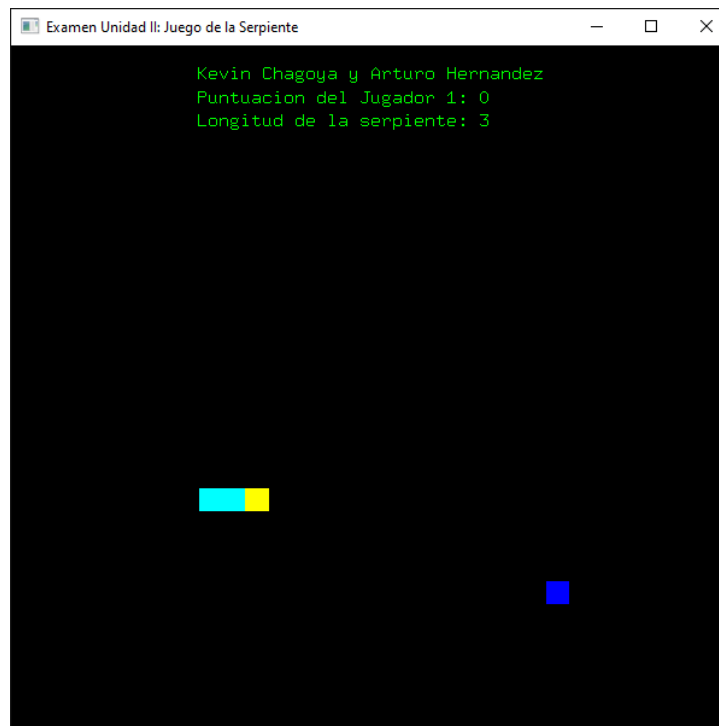
➤ Reglas De Juego

Para poder jugar este juego se necesita saber antes las reglas para llevarse una mejor experiencia. Comenzando por la forma de ganar.

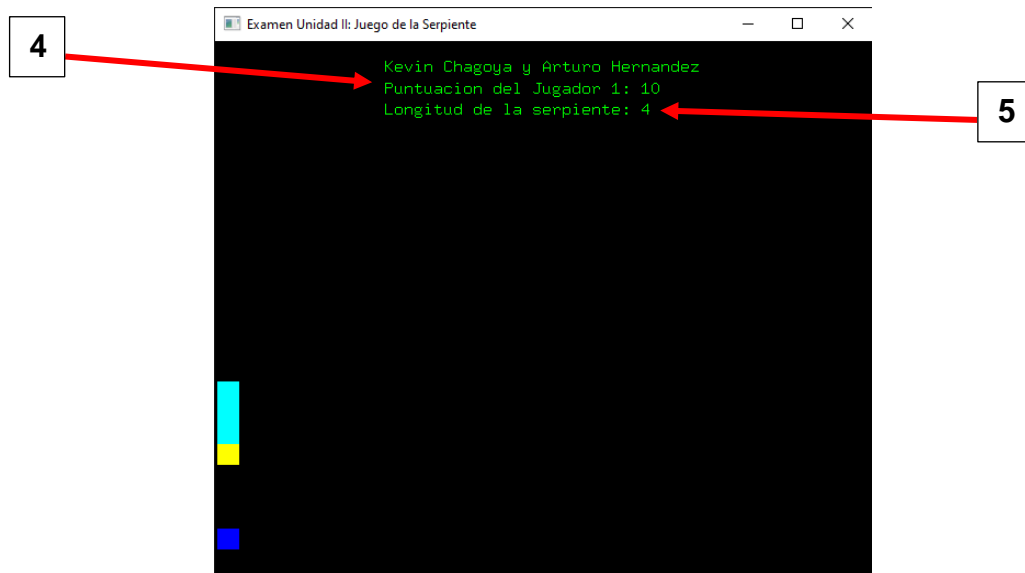
1. **Nota: Teclas a utilizar:** El jugador debe de pulsar las teclas de dirección de su teclado para direccionar el desplazamiento de la serpiente (Arriba, abajo, izquierda, derecha).



2. **Inicio:** Al iniciar el juego la serpiente empezará a desplazarse por la ventana en una dirección aleatoria con una longitud inicial de 3, así como la comida también aparecerá en alguna posición aleatoria dentro de la ventana.

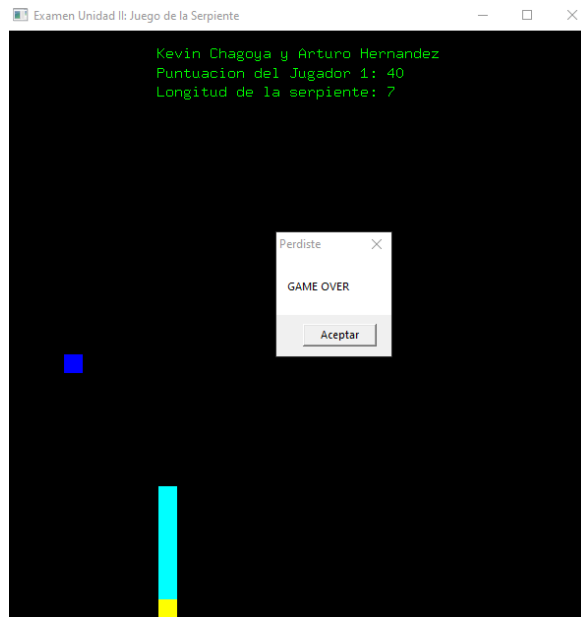


3. **Objetivo:** El objetivo del juego es que la serpiente se desplace hacia donde se encuentre la posición de la comida y pasar sobre ella para ganar puntuación, la dirección se le dará el jugador con las teclas dirección.
4. **Puntuación:** El puntaje se elevará 10 puntos por cada comida que la serpiente coma.
5. **Longitud Serpiente:** Por cada comida que la serpiente coma, la longitud de la misma se ira elevando en 1.
6. **Forma de ganar:** Para lograr ganar en el juego de la serpiente esta deberá haber comido una cantidad de 25 comidas, pero cabe resaltar que cada que esta come la velocidad aumenta lo cual incrementa un poco mas la dificultad.



7. Formas de perder:

- **Choque con el borde:** Podrás perder si la serpiente choca con algún borde de la ventana, con esto se finaliza el juego y aparecerá un cuadro de dialogo con el texto “GAME OVER”.



- **Choque con su mismo cuerpo:** Otra forma en que se puede perder es que la serpiente choque con su mismo cuerpo. Al irse este alargando la probabilidad de que choque con su mismo cuerpo es más grande, esto lo vuelve más complicado e interesante. De igual forma si esto sucede, aparecerá un cuadro de dialogo con el texto “GAME OVER” y se finalizará el juego.



8. Para terminar con la ronda perdida solo se debe de dar clic en el botón “Aceptar” de nuestra ventana emergente y a continuación el juego finalizara por completo.

