



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en Software y tecnologías emergentes

**Materia:** Paradigmas de la programación

**Alumno:** Vazquez Guzman Jorge Antonio

**Matrícula:** 372504

**Maestro:** Carlos Gallegos

**Tema - Unidad :** Practica 1

**Fecha de Entrega :** 30 de Mayo del 2024

**Ensenada Baja California a 18 de Mayo del 2024**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Introduccion

Se pidió crear un snake con ciertos criterios, los cuales se trataron de seguir a medida de lo posible.

### Desarrollo

Algunos de los fragmentos del código son los siguientes:

```
#include "raylib.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#ifdef PLATFORM_WEB
#include <emscripten/emscripten.h>
#endif

//-----
// Algunas definiciones
//-----
#define SNAKE_LENGTH 256
#define SQUARE_SIZE 29

//-----
// Tipos y definiciones de estructuras
//-----

// Estructura para almacenar los datos de la serpiente
typedef struct Snake
{
    Vector2 position;
    Vector2 size;
    Vector2 speed;
    Color color;
} Snake;
```



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

```
if (!pause)
{
    // Control del jugador
    if (IsKeyPressed(KEY_RIGHT) && (snake[0].speed.x == 0) && allowMove)
    {
        snake[0].speed = (Vector2){SQUARE_SIZE, 0};
        allowMove = false;
    }
    if (IsKeyPressed(KEY_LEFT) && (snake[0].speed.x == 0) && allowMove)
    {
        snake[0].speed = (Vector2){-SQUARE_SIZE, 0};
        allowMove = false;
    }
    if (IsKeyPressed(KEY_UP) && (snake[0].speed.y == 0) && allowMove)
    {
        snake[0].speed = (Vector2){0, -SQUARE_SIZE};
        allowMove = false;
    }
    if (IsKeyPressed(KEY_DOWN) && (snake[0].speed.y == 0) && allowMove)
    {
        snake[0].speed = (Vector2){0, SQUARE_SIZE};
        allowMove = false;
    }
}
```



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

```
// Colisión
if ((snake[0].position.x == fruit.position.x) && (snake[0].position.y == fruit.position.y))
{
    // InitAudioDevice();
    PlaySound(eaten);

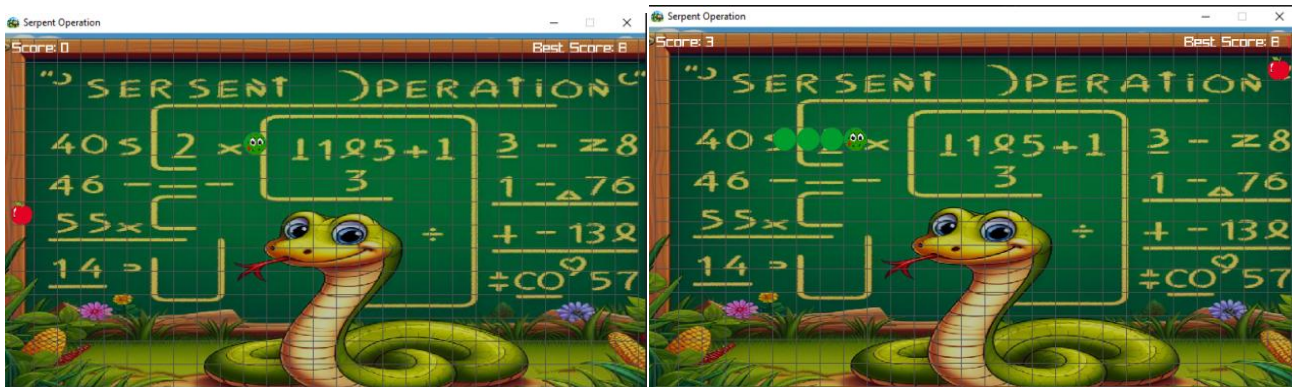
    snake[counterTail].position = snakePosition[counterTail - 1];
    counterTail += 1;
    fruit.active = false;
    fruitCollected = true;
    fruitsEaten += 1;
    score += 1;

    if (fruitsEaten == 3)
    {
        pause = true;
        gameState = PAUSED; // Cambia el estado del juego a pausado
    }

    // Restablece el contador después de tres frutas comidas
    if (fruitsEaten == 3)
    {
        fruitsEaten = 0;
    }
}
framesCounter++;
```

Esos son algunos de los métodos que se utilizan para controlar ciertas cosas del juego.

El juego se mira de la siguiente forma.





# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

El juego fue desarrollado en clases pasadas, por lo que opte por una de las buenas practicas de programación. CREAR CODIGO REUTILIZABLE, el código se encuentra en un repositorio distinto al de la materia, ademas que integra otras opciones como operaciones matemáticas.

El link al repositorio es el siguiente: <https://github.com/JVPresti/Juego>

En caso del repositorio encontrarse en privado, se podría cambiar momentáneamente.