Snake Fruit Game In C

Atish kumar sahu
Software Developer

01-Aug-2023

—

C Programming Language

—

Atish kumar sahu

Content:

About:

Greetings! I am Atish Kumar Sahu, a dynamic and dedicated individual hailing from the enchanting city of Berhampur, Odisha, India. With a relentless passion for technology and innovation, I have made significant strides in the world of software development. During my tenure as a Junior Application Developer at Pantheon Inc, from June 2022 to October 2022, I immersed myself in the realm of application development, honing my skills and contributing to th e creation of cutting edge solutions. My pursuit of knowledge led me to acquire a B.Tech degree in Computer Science Engineering (CSE) from Parala Maharaja Engineering College, graduating in the year 2022. Throughout my academic journey, I demonstrated a keen aptitude for learning and consistently showcased an exemplary work ethic. As a professional, I pride myself on my adeptness in team management, fostering collaboration, and driving projects to successful completion. My unwavering focus and determination enable me to tackle challenges head on, delivering results that exceed expectations. In terms of technical expertise, I possess proficiency in an array of programming languages, including C, Java, and MySQL, and my knowledge extends to the realm of web development. Additionally, I am well versed in utilizing tools such as MS Office and Google Suite to streamline operations and boost productivity. In summary, I am an enthusiastic and adaptable individual, committed to delivering exceptional outcomes in the realm of software development. With a solid foundation in technology and a penchant for hard work, I eagerly embrace opportunities to contribute meaningfully to projects and organizations. Thank you for considering my profile, and I look forward to making a valuable impact wherever I embark on my professional journey.

Introduction:

Snake Fruit is a classic and addictive arcade game that has been enjoyed by generations of gamers around the world. The game concept is simple yet captivating, as players control a snake to collect "fruit" while avoiding collisions with the walls and the snake's own tail. The game's challenge lies in increasing the snake's length as it devours more fruit, making maneuvering progressively more difficult. In this document, we will explore the implementation of the Snake Fruit game using the C programming language. We will delve into the key functions that bring this game to life, including `void snake()`, `void print()`, `void resetscreen()`, `void random()`, `void getchar()`, `void movement()`, `void tailmove()`, and `int gameover()`.

By understanding the structure and functionalities of these functions, we can grasp the logic behind the game and how it unfolds during gameplay. The main function of the game, `void snake()`, lays the foundation for the game's mechanics. It initializes the game, sets up the playfield, places the snake and fruit at their initial positions, and prepares the screen for display.

This function acts as the entry point for the game, orchestrating all other functions and interactions between elements. By understanding the functions involved in the game's development, aspiring programmers can gain insights into game development fundamentals, such as handling user input, updating screen displays, and managing game logic. Whether you're a beginner or an experienced coder, exploring the Snake Fruit game in C presents an exciting opportunity to hone your programming skills and build captivating interactive applications.

C Programming Language:

C is a powerful and widely used general purpose programming language created by Dennis Ritchie in the early 1970s at Bell Labs. Known for its efficiency and portability, C has become the foundation for numerous operating systems, software applications, and embedded systems. It is a procedural language, following a structured approach to programming. Its syntax is minimalistic yet expressive, providing low level control over the hardware while maintaining a relatively simple and elegant design.

C's popularity stems from its ability to directly interact with the computer's memory, making it suitable for system level programming tasks and optimizing performance critical applications. Its standard library offers a rich set of functions to manipulate strings, arrays, and other data structures efficiently. As a widely taught programming language, C serves as an excellent starting point for learning computer programming concepts and understanding the underlying principles of how computers work. Despite its age, C remains relevant and widely used in various domains, such as operating systems, compilers, embedded systems, and game development, making it a fundamental language for programmers to master.

Header File:

A header file is a file with extension (.h). a general practice in c or c++ program is that we keep all the constants, macros, system wide global variable, and function prototypes in header files and include that header file whatever it is required. Both user and system header files are included using the preprocessing directive #include.

It has following two forms. If a header file happens to be twice, the compiler will process its content twice and my result an error. the standard way to prevent this is to enclose the entire real contents of the header file in a conditional as follows.

Required Header Files In This Project:

#include<stdio.h>

In c programming #include<stdio.h> is a preprocessor directive that tells the compiler to include the standard input-output library in the program. the library is essential for input and output operation like reading from the keyboard and displaying on the screen.

It provide functions like "printf()" for formatted output and "scanf()" for formatted input. Including "stdio.h" at the beginning of a c program allows you to utilize these input-output functions perform essential I/O operations with your code.

#include<windows.h>

The "windows.h" library provides functions for creating and managing and windows, handling messages working with resources, and interacting with the windows environment. It also includes constants and data types necessary for windows programming.

#include<stdlib.h>

The #include <stdlib.h> is a preprocessor directive in the C programming language. It is used to include the standard library header file stdlib.h in your C program. This header file provides several standard functions for memory allocation, random number generation, and other essential utilities.

#Include<conio.h>

The header file provides function like "getch()" and "clrscr()" which facilitate character input without displaying it on the screen and clearing the screen, respectively. However it's essential to note that the "conio.h" library is not portable and may not work on all systems or modern compilers. As a result, it is generally adviced to use standard input-output functions from "stdio.h" for portability and compatibility across different platforms.

#include<time.h>

The #include <time.h> is a preprocessor directive in the C programming language. It is used to include the standard library header file time.h in your C program. This header file provides functions and data types related to time and date manipulation, making it an essential tool for time-based operations and applications.

By including time.h in your C program, you gain access to various functions that allow you to retrieve and manipulate time-related information. One of the most common functions is time(), which returns the current time in the form of the number of seconds since the epoch (January 1, 1970, 00:00:00 UTC). This function is valuable when you need to record timestamps, measure the execution time of code, or create time-based algorithms.

Pre-Processing & Macro Expansion In C:

The C preprocessor is a micro-processor that is used by compiler to transform your code before compilation. It is called micro preprocessor because it allows us to add macros. It is a program that processes our source program before it is passed to the compiler. Preprocessor commands (often known as directives) form what can almost be considered a language within C language. The c preprocessor offers following operators to help in creating macros, macros continuation(\), stringize(#), token pasting(##). one of the powerful functions of the C preprocessor is the ability to simulate functions using parameterized macros. The preprocessor defined operator is used in constant expression to determine if an identifier is defined using #define if the specified identifier is defined, the value is true (non-zero). if the symbol is not defined the value is false(zero).

A macro is a segment of code which is replaced by the value of macro. Macro is defined by "#define" directive. There are two types of macros. a macro is a text substitution definition which mean whenever a macros is called the respective text for which macro is defined gets expanded at the line of call.

Function In C:

In every programming language function plays an important role. Basically function is a block of code which has some name of identification. Even in the C programming the smallest program there is at least one function. All function name must be unique. In C language the main() function is the mandatory function, because compiler will start its execution from main() function. So, that there is one function name, just be main() function. you can define function is any sequence, no keyboard is a function. Operating System calls main() function to begin the execution of every C programming source code.

Type Of Function:

Generally there are two types of function are there in every programming language. In C programming language also there are two types of function is there.

1. Pre-Defined Function:

These are the types of function which is already created for C programming language to simplify the source code. Basically these functions actual code was stored in library flies or you can say the header files which are the (.h) extension files. Because of that in every source code we need to add the header file to simplify our source code. Example: printf(), scanf(), exit(), pow(), etc.

2. User-Defined Function:

These are the types of function which are created by user/programmer who wants to simplify the code according to its requirement. Function helps a user from so many ends. User defined function must be declared by programmer or user. The programmer has to write the code for such function and test them properly before using them. Example: void print(), void resetscreen(), void movement(), void random(), etc.

Return Type Function:

A return type function is a function that explicitly returns a value after performing its operations. It uses a data type to specify the type of value it will return and the return keyword is used to send the result back to the calling code.

Non-Return(void) type Function:

This type of function does not return any value. instead it performs a set of operations or tasks without providing a result to the caller. It declared using the "void" keyword indicating that no value will be returned from the function.

Require Functions In This Project:

int main():

In this project in the main function we call all other following user-defined function like void snake(), void print(), void resetscreen(), void random(), int getchar(), void movement(), void tailremove(), int gameover(). Inside the main function using a while loop we called void print(), void resetscreen(), void random(), int getchar(), void movement(), void tailremove(), int gameover() except void snake(), because these functions will operate every time till the score of the player will increase by 10. The void snake() function is not in iteration mode because when we start the game the snake starts from the middle of the border and if we iterate the snake function every time then the snake will not move forward.

void snake():

In this function we will initialize both snake and fruit by using loop and array concepts.

void print():

In this function first we create a boundary of the game. After the creation of the boundary we use conditions to create the snake and fruit. The fruit will generate only inside the boundary of the game.

void resetscreen():

In this function we reset the screen every time whenever the snake moves and after the game over.

void random():

The above function is used for to create the fruit inside the boundary but in random order.

int getchar():

The above function is used for to take the keyboard key hit value.

void movement():

The above function is used for to move the entire snake only inside the boundary and we took key "a" for move left, key "d" for move right, key "s" for move bottom, and key "w" for move up. If in case the head of snake touch the boundary the snake will appear at the opposite of the boundary like for left to right and up to bottom.

void tailremove():

This function is used for to move the tail of snake with the head and increase the tail size of the snake.

int gameover():

In this function if incase the snake touch its own body then the game will over and after that in the bottom of the game it will show the score of your game and score value will saved in a file if the score is the highest score. The file will created in your "E" drive. And after the game over it will ask for to paly again or exit the game.

Decision Control Statement:

Decision control in programming allows the execution of different code blocks based on specified conditions. It uses conditional statements like if-else and switch-case to determine which set of instructions to run, enabling programs to adapt and make choices based on input or variable values.

Iterative Control Instruction:

Iterative control instructions in programming, like loops (e.g., for, while, do-while), enable the repeated execution of a block of code until a certain condition is met. They are used to automate repetitive tasks, iterate over data structures, and create efficient algorithms for handling large datasets.

Conclusion:

In conclusion, the Snake Fruit game implemented in C programming language demonstrates the successful integration of essential functions, including void snake(), void print(), void resetscreen(), void random(), void getchar(), void movement(), void tailmove(), and int gameover(). This game provides an engaging and interactive experience, showcasing the power of programming in creating entertaining applications. Through the clever utilization of these functions, players can enjoy navigating the snake to collect fruit while avoiding collisions. The project serves as an excellent example of leveraging C's capabilities to build a classic arcade game that entertains and challenges users.

----- THE END -----