****

**THE SNAKE GAME**

**A documentation presented to Ms. Ashlyn Kim Balangcod**

**In Partial Fulfillment  
of the requirements in Introduction to Computer Sciences**

Lim, Lyka Raquel C.

**November 2019**

**DOCUMENTATION**

1. **INTRODUCTION** The snake game is a very popular game which involves the player controlling the snake. The concept originated in the 1976 arcade game Blockade, developed and published by Gremlin. This version of the game is made in C language and is very easy to play. The main objective of this game is to simulate the famous snake game. This is the classic version wherein the aim is to let the snake eat all the foods present in the boarder, every consume of a food particle, its length increases. When a food is eaten, the snake grows a body segment and lengthen its size. After picking up all the foods in the boarder, a magic portal will appear and if the snake goes there, the player wins the game, if the snake bumps into the boarders/additional blocks or accidentally eats himself, game is over.

The player is free to select the difficulty level, there are two levels - the easy and difficult level. In easy level, only the snake and the randomly placed food will be inside the boarder. In difficult level, there would be additional blocks inside the boarder in which if the snake collides with it, player losses the game.

1. **DESCRIPTION OF DATA STRUCTURES**
   1. **Libraries used:**

* *#include<graphics.h>* **:** it gives the program an access to simple graphics library.
* *#include<stdio.h> :* Standard library functions for file input and output.
* *#include<time.h> :* contains definitions of functions to get and manipulate data and time information.
* *#include<stdlib.h> :* Standard library functions of C which includes memory allocation, process control, conversions, and others.
* *#include<conio.h> :* Header in C that contains functions for console input/output.
  1. **Variables:**
* *Int gd :* gd stands for graphdriver. It is an integer that specifies which graphic driver is to be used. In this program, we used DETECT.
* *Int r :* Integer used to initialize and represent the grid’s row.
* *Int c :* Integer used to initialize and represent the grid’s column.
* *Int food :* Integer used to represent the snake’s food.
* *Int p :* Integer used to represent the magic portal that will appear at the last part of the game for the player to win.
* *Int length :* Integer used to represent the snake’s length.
* *Int exits :* integer used to count the snake’s length outside the board.
* Int main : main function
* Int upd : integer for update
* Char m : Character used to represent the snake’s movement.
  1. **Functions:**
* *Void display ( ) : a function used to display the board, food, and blocks.*
* *Void movement ( ) : a function used to make the snake move.*
* *Void initFB ( ) : a function used to initialize the snake’s food and the blocks.*
* *Void update ( ) : a function used to update the snake’s movement.*
* *Void end ( ) : a function used to terminate the program.*
* *Void win ( ) : a function used to display that the player wins the game.*
* *Void resetValues ( ) : a function used to reset the values of all the variables after a game*
* *Void portal ( ) : a function used to make the magic portal appear.*
* *Void menu ( ): a function used to show the game’s menu.*
  1. **Conditional Statements**:
* If statements : a conditional statement that if “if” is true, the program will perform the function or display the information.
* Else If statements : a conditional statement used to specify a new condition to test in a sequential form if the first “if” statement is false.
* Else statements : a conditional statement that if “if” is false, the program will perform the function or display the information.
  1. **Randomize**
* Srand(time(NULL)) : is used to generate random numbers. It is also used in randomizing foods, blocks, and the magic portal.
  1. **Graphics.h**
* Setbkcolor : Used to set the background color in graphics.
* Settextstyle : Used to set the text style of the words to be displayed in the game.
* Outtextxy: Used to print the words and instructions in the game.
* Getch ( ) : Used to get characters being input by the player.
* Cleardevice ( ) : Used to clears the screen in graphics
* Setfillstyle( ) : Sets the current fill pattern and fill color.
* Bar ( ): used to draw a rectangular filled in bar.
* Setcolor ( ) : Used to set the colors in graphics.
* Rectangle ( ) : Used to display rectangles in graphics.
  1. **Loops**
* While loop: Loop used to randomize the foods so that no foods in the same array row and column will be placed twice.
* For loop: loop used for counting the snake’s length, the foods, the blocks and the portal.
  1. **Arrays**
* *Char arr[440][440] : a grid made up of array u*sed to hold every game object, even parts of the snake.
* *Int xbody[1250], ybody [1250] :* array integer used to store the coordinates of the snake’s body.

1. **ALGORITHM (PSEUDOCODE)**
2. **Initialize the variables and functions**

Int gd, gm, xbody[50], ybody[50], r=65, c=65, food, p=o, length = -1, exits = 0;

Char arr[440][440];

void display(); void movement(); void initFB(int diff); void update(); void end(); void win(); void resetValues(); void portal(); void menu(); int main()

1. **Display menu**
2. Start Game, (2) Instructions, (3) Exit

If player chooses 1, go to 3

If player chooses 2, go to 4

If player chooses 3, go to 5

Else, go to 6.

1. **Start game**
2. Scan for the level of difficulty: (1) Easy, (2) Difficult
3. Store the selected level of difficulty
4. If option is not equal to 1 or 2, Go to A.
5. Ask user to input number of foods (0 to 50 only).
6. Store number of foods
7. If option is less than 0 or greater than 50, ask the user again to input a valid option. Go to D.
8. Randomize the placement of foods in the board.
9. If option in A is 2, Randomize blocks
10. Display board, foods, blocks(if difficult), snake’s head.
11. Movement
12. Ask input for snake’s head movement with the options (W or w) to go up, (D or d) to go to the right, (S or s) to go down, (A or a) to go to the left, and (Q or q) to quit the program.
13. If option is Q or q, go to 5.
14. If option is not equal to (W or w) or (D or d), or (S or s), or (A or a), or (Q or q). Go to i.
15. If snake’s head collides with a food, its length will increase by 1 block
    1. the increased block will follow the previous location of the snake’s head. If the length is more than 2, the rest of the blocks will follow the preceding block’s previous locations.
16. If the snakes head collides with a block, boarder, or its body. Display “Game over”.
17. Ask the user if he/she wants to play again: (1) Yes, (2) No. If player presses 1, go to 2. if player presses 2, go to 5.
18. If the food equals 0, randomize a location for the portal (exit)
19. If the whole snake enters the portal. Display “Congratulations you win the game.” The player needs to press any key, then to go to 2.
20. **Instructions. Go to 2.**
21. **Terminate the program**
22. **Ask the user again to input a valid option. Go to 2.**
23. **ERROR HANDLING**

* When pressing random keys during game proper, the program will disregard the key being pressed.
* When pressing random keys that are not in the options, the player will be asked to put a valid option.
* When the player entered foods less than 10 or greater than 50, the player will be asked again to input a valid number of foods.

1. **APPLICATION MANUAL**
2. **Set up CODEBLOCKS**

* Open your browser.
* Type “codeblocks.org” in URL section.
* In the upper left corner of the screen, click the “download section”.
* Download the binary release.
* Find for codeblocks-17.12-setup.exe.
* Download the FossHUB or Sourceforge.net
* File will start to download. Click “yes” to Set codeblocks on your computer.
* For installing CodeBlock on your computer, click “next” then “I agree”.
* Click “next”, and click “install” to install the compiler on your computer.
* Click “yes” to run the codeblocks, then click “next” to open the codeblocks.
* Click “ok” to start using codeblocks.

1. **Add “GRAPHICS.H” header in Codeblocks.**

* Download “WinBGIm” from <http://winbgim.codecutter.org/>
* Extract the three files : (a) graphics.h , (b) winbgim.h , (c) libbgi.a
* Copy and paste the downloaded files “graphics.h” , and “winbgim.h” files into the include folder of code::blocks
* Copy and paste libbgi.a to the lib folder of compiler directory
* Open codeblocks
* Go to settings, then compiler and linker settings
* Under the “Link libraries”. click the add button and browse. Select the libbgi.a file copied to the lib folder.
* In “other linker options”, paste the following commands -Ibgi -lgdi32 -lcomdlg32 -luuid -loleaut32 -lole32 -lwinmm
* Click ok
* Open graphics.h file with notepad ++
* Go to line 302 and replace that line with this line : int left = 0, int top = 0, int right = INT\_MAX,
* Save the file.

1. **Open Code::Blocks**

* Run the program

1. **Start the game**

* The program will start with a menu, the player will be given 3 options. (1) Start game, (2) Instructions, and (3) Exit.
* If the player presses the option 1, the player will be given two choices, either to play in an (1) Easy Mode or (2) Difficult mode. Easy mode will have no blocks while difficult mode will include blocks on the board.
* After choosing the level of difficulty, the program will ask for the number of foods the snake must eat.
* The foods then will be distributed across the board then the program will proceed to the game proper.
* If player losses, game is over. If player wins, the program will congratulate you. Then the program will go back to menu.
* If the player presses the option 2, the instruction on how to play the game will flash on the screen, this includes the game mechanics and the keyboard controls the player needed to know in order to play the game. Written in here is the player’s guide to win the game.
* The program will ask if you want to proceed to the game or not.
* If player chooses (1) Yes, the program will go back to the menu. If player chooses (2) No, the game will end and the program will terminate.
* If the player presses the option 3, the program will terminate.