Snake *Returns*

Documentation

Lab Batch 33 CS101 Project

# Index

1. Introduction
2. Requirements
3. Game Basics
4. Working
5. Member Variables
6. Member Functions
7. Credits

# Introduction

Snake *Returns* brings back the era of Snakes gaming but with a twist. Now you use a mouse to control your snake through the maze to his fruit. Now you can enjoy bonus surprises. Get addicted.

# Requirements

1. Ubuntu (preferably 10.04)

*Other requirements yet to be determined.*

# 3. Game Basics

Snake *Returns* is nothing but a revamped version of the same old addicting game with new additions and features.

There are 10 levels in all. And each level has 12 fruits (plus some bonus) to offer. To complete each level the user has to eat all the 12 fruits of the level.

## Snake Movement

The snake moves on its previous path until the user clicks to change the direction. When the user clicks, the snake turns in the direction of the click. The snake cannot pass through walls nor can it eat its own body. Any of the above event results in Game Over immediately. Hence the user has to manoeuvre the snake to its food.

## Fruits

Every time the snake eats up the fruit, a new fruit is plotted on a new co-ordinate, which is determined using a random function. Also a different fruit is generated each time. Eating twelve fruits in a level clears the level.

## Levels / Maps

The game kicks off from an easy level and the difficulty increases as the user ascends. Levels will have mazes, different theme and some twists on the higher levels.

## Scoring

The scoring is time based. Based on the time taken by the user to reach the fruit, he will be awarded the points. The level score and the total score alongwith the highscore is always displayed on the top.

# 4. Working

Snake *Returns* is based on EzWindows API, which comes for C++ programming language. The following paragraphs explain how the game actually works.

As soon the game starts, first all the environment/global variables are initialised. Then all the bitmaps used in the game are loaded followed by loading the level 0 (that is the menu level).

The menu level is a standard menu with options for New Game, High Scores and Exit.

The snake has three parts: head, body and tail. All parts are different blocks represented with different bitmaps.

The Map and Maze Algorithm: Each map has two part: an actual bitmap and a two dimensional array. The array stores values corresponding to the co-ordinates which contain the walls or fruit. This array is loaded from an external file along with the level bitmap image.

Snake Movement Algorithm: The program is basically designed to control the movement of snake’s head. The rest of the body part simply follows the previous body part. Also the direction of the snake’s movement is controlled by a global variable. Hence whenever the snake is to be turned the globally set direction variable is changed and the snake is turned.

Scoring Algorithm: Yet to be finalised.

Check Algorithm: After every snake movement check for snake’s head co-ordinates has to be done to assert whether it has eaten the fruit or struck a wall or eaten itself. This is simply done by checking the value in map array at co-ordinates of snake’s head. If that’s 1 it means it has struck a wall and if it is -1 it has eaten a fruit! A separate check is made under an iteration to check that its not eating its own body.

As the user eats up fruits the length of the snake increases. This is done by increasing no. of blocks used up to build the snake’s body.

# 5. Member Variables

int snake[][] : *This is a 15 X 2 matrix which is used to store the co- ordinates of the snake’s body.*

int map[][] : *This is a 40 X 28 matrix which is used to store the co-ordinate system of the whole map.*

int level :  *This represents the level of the game on which user plays.*

int subLevel :  *This represents the number of fruits eaten by snake and hence, determines the length of the snake.*

int delay : *Delay is a time interval which is used to call the function “move” repeatedly after that time interval.*

int xsnake, ysnake : *”xsnake” and ”ysnake” are the x-coordinate and y-coordinate of the snake respectively.*

char dir : *“dir” represents the direction of the movement of snake (initially right “R”).*

int fruitcord[][] : *This is a 2 X 2 matrix used to store the co-ordinates of the fruit.*

int fruit\_type : *This variable is used to determine which fruit is to be plotted.*

Position center : *This is to locate the centre of the window.*

BitMap head : *This is the bitmap image of the head of the snake.*

BitMap body : *This is the bitmap image of the body of the snake.*

BitMap tail : *This is the bitmap image of the tail of the snake.*

# 6. Member Functions

int ApiMain() : *The main function which loads images of the head, body and tail of the snake and calls for gameplay.*

int prepLevel() : *This function is used to plot the designed level on the map.*

int playGame() : *This function is used for the basic gameplay of the game.*

int move() : *This function is used to move the snake such that the snake’s body follows its head.*

int randomCood(int l) : *This function is used to generate a pair of random co-ordinates in the map.*

int loadMap() : *This function loads the map of the level and assigns a particular value at a point where wall, fruit or snake is present.*

int plotFruit() : *This function plots the fruit on the random co-ordinates generated and takes care that it does not lie on the snake or a wall.*

int mouseClick(const Position &p) : *This function is called when mouse click occurs. It turns the snake according to the position of the mouse click.*

int timerClick() : *This function is called repeatedly after a particular time interval. It is used to keep the snake moving and check for head crash at each step.*

int check() : *This function checks whether the snake collides with the wall and whether it eats the fruit.*

int check\_fruit() : *This function checks that the fruit does not lie on the wall or the snake.*

# 7. Credits

team 1

Kumar Pallav

Aman Mangal

Akshay Godara

Palli Ashish

team 2

Pulkit Maheshwari

Prateek Agarwal

Kartik Chaudhary

Nitesh Meena

TEAM 3

Vishnu Vardhan

Ashok Kumar

Gautam Sumu

Aman Bansal

Himani Jain

TA

Kalpit Dixit