

Software Requirements specifications

FOR CS101 PROJECT

CHAIN REACTION

By team members

Ranvijay Singh(140020064)

Tanany Banerjee(140020019)

Lokesh Soni(140020069)

Index

1. Purpose
2. Problem statement
3. Functional specifications
4. Data Specifications
5. Softwares Required
6. Appendices

PURPOSE:

One of the purposes of this project is to prepare a strategy game, which is interesting for the user to play. The main purpose is that we wanted to choose a project wherein we get to use maximum number of concepts of c++ like arrays, pointers, functions and so on. We have also used this as an opportunity to learn graphics.

Problem Statement

To create a replica of the Chain reaction game provided by various websites and Google play store. We are using the knowledge bestowed onto us in the CS101 course, Youtube videos and also with the help of some of the websites.

Functional Specifications

Functions of the Computer:

The basic tasks performed by the system may be broadly classified as follows:

In this GUI interface responding to user events.

The main task performed by the computer is registering the mouse click of user and input typed using keyboard and responding accordingly, as per where the mouse is clicked or which key is pressed.

The system will also provide the game area according to players' choice. The system also validates the moves and executes the players moves in a specific pattern.

Libraries used

For graphics in the game we have used SDL library .

The various functions used throughout the program and a brief description:

int displayfirstscreen()

{ Displays a screen for the player to choose between 'play', 'instruction' and 'quit' options'.

- 'play' option continues the game
- 'instruction' option displays instructions regarding the game
- 'quit' option exits the game

The function returns the value 0 , 1 or -1 if 'play', 'instruction' or 'quit' option is selected respectively.}

int displaysecondscreen()

{Displays a screen to select the number of players to play the game. There can be 2, 3, 4 players. According to player specification the function returns values = 2, 3, or 4 depending upon input received.}

int displaythirdscreen()

{Displays a screen for the user to choose the size of the grid.

The grid can be 5x5, 8x8, 10x10 according to user requirement. The function returns value = 5, 8, or 10 depending upon the size of the grid chosen.}

int keyboardormouse()

{Displays a screen for the user to choose between mouse or keyboard gameplay within the game. The function returns value = 3 or 2 if keyboard or mouse is chosen respectively.

int validatemove()

{This function checks whether a requested move is applicable or not with respect to certain conditions.If the move is valid it returns 0 otherwise it returns -1. It is called from within gridkey function.}

void executemove()

{This function is called from within gridkey function. It executes the blast done by a player after it has been validated by the validatemove function. }

void applybutton()

{This function uses sdl library functions to apply a required specified image at a certain specified coordinate passed to the function (the images to be applied are initially loaded before calling the function) }

void displayinstructionscreenscreen()

{This function opens a window which displays the instruction the players about various aspects of the game.It returns to the main function when 'a' is pressed.}

void gameoverscreen()

{This function displays the screen after the game is over. It displays the index of the player who won the game}

void gridkey()

{It is the main function where the game takes place.It opens a window of size according to the gridsize specified by the players. The game continues till the moment atoms of only one colour remain in the grid. In such a such it ends the game and a gameover screen is displayed according to the colour of the atoms remaining in the grid.}

void applyimage()

{This function makes use of certain SDL functions. It takes an image to clip it on the screen at given coordinates and makes the background of the image transparent so as to display the background colour of the screen.}

void exec()

{This is a function called within the executemove function. It checks the value of the Grid[][] array and makes certain changes as required for the game.}

Other SDL library specific functions are used in specific functions of the program.

Data specifications

This game uses mouse and keyboard for input and output is displayed on the screen. Most of the data used in the program is generated by the system in runtime. This is handled at using variables, and arrays among other data structures.

Specifications Required:

- Open source C++ compiler which includes sdl library.
- Requires 20MB of disk space.
- A decent processor to support(intel core i3 or higher is suggested)

- The game works completely on Windows OS.(this program has not been tested in Ubuntu or other linux based OS')

Unresolved Problems faced in execution of the program:

- Game runs perfectly when in mouse part but gameoverscreen is not called in keyboard part of gridkey function.
- We could not include the time part as time.h library did not sync properly with SDL library.

Appendices:

Credits

CS101 Course Prof. Deepak B Phatak and
Prof. Supratik Chakraborty
LabBatch TA : Prakhar Gupta

Batch members

Ranvijay Singh (Team leader)
Tanay Banerjee

Lokesh Soni