PROJECT REPORT

CHAIN REACTION

GROUP MEMBERS

JAYANTH SHANKAR CHENNAREDDY (TEAM LEADER)- 140050039

NARU DIVAKAR REDDY-140050044

VISHAL BABU BHAVANI-140050049

Introduction:

The aim of the project is to design a game which resembles the android app chain reaction. Chain Reaction is a game in which players have to use their brains carefully. It is very difficult to get a strategy to win. One has to continuously review and change it with every move of the opponent which makes the game very exciting and eventually more fun.

Purposes:

The game requires two players, and also permits a single player to compete against a computer opponent. The rules that follow are quite similar to those of the traditional Chain reaction (explained later).The benefit of this program is primarily that it allows someone to play against others using his wits and intuition of a chain reaction.

Game Rules:

Players take it in turns to place their orbs in a cell. Once a cell has reached critical mass the orbs explode into the surrounding cells adding an extra orb and claiming the cell for the player. A player may only place their orbs in a blank cell or a cell that contains orbs of their own colour. As soon as a player loses all their orbs they are out of the game.

Division of Work:

Vishal Babu Bhavani:

Wrote the code for artificial intelligence, contributed significantly towards the algorithm for the 2-player game and much of the code for the graphic user interface.

Jayanth Shankar Chennareddy (team leader):

Wrote the algorithm for the 2-player version of the game, contributed to the graphic user interface by editing images and making the backgrounds for the interface using MS Paint.

Divakar Reddy Naru:

Contributed significantly to the algorithm of the 2-player version of the game, and the inclusion of the undo and save options in the game. Contributed significantly to debugging quite a number of bugs that crept up during the development of the game.

ACKNOWLEDGEMENT:

In the present scenario, where the world is taking breath at a high pace, computers have now become an integral part of our life. We are given with an excellent opportunity to learn computers by our institute under the guidance of one of the best professor in Computer Science of India Dr.D.B.Phatak. He increased the field of our knowledge by indulging us in such a good project which requires a good knowledge of C++ language and graphics. Lessons by sir during the classes proved to be of great help .We learned great qualities like Professionalism, Team work, Self Assessment ,etc which are sure to play an important role in our life. Last but not the least we will also like to thank our very helping TA Ahzaz Hingora who was there always to help us in any difficulty and to clear our doubts.

Overall it was a great journey and again a big thanks to our professor.

Functions Used:

We used the graphics library SFML(Simple and Fast Multimedia Library) to cater to our needs of graphics.

Different libraries used are:

```
#include <SFML/Graphics.hpp>

#include <SFML/System.hpp>

#include <SFML/Audio.hpp>

#include<iostream>

#include<cstdlib>

#include<cstdio>

#include<cstring>
```

Functions that we used:

int criticalmass(int i, int j);

Function to find the maximum no. of allowed orbs in a cell

void arti(int gamespace[8][6],int*x,int*y,int*z);

Function that executes the artificial intelligence part of the game.

int sum(int temp[8][6]);

It is a function which is used by the arti function to return the total no. of the player's orbs in a certain configuration.

int sum2(int temp[8][6]);

It is a function used by arti to find by how may balls the player exceeds the system

bool winmove(int temp[8][6],int*x);

This function is used by the arti function to check whether a winning situation is possible for the player in the next move if the system plays a certain move.

void highlight(int gamespace[8][6],int i,int j,int w);

This function is used to highlight the portion of the grid on which the mouse pointer is moved and also plot the orbs on to the highlighted cell.

void plotter(int gamespace[8][6],float t);

This function plots the orbs onto the grid present on the main gameplay window.

void fission1(int gamespace[8][6],int i, int j, int* x);

This function checks the possibility of fission in the gamespace in the 1st player's turn and undergoes them if possible.

void fission2(int gamespace[8][6],int i, int j);

This function checks the possibility of fission in the gamespace in the 2nd player's turn and undergoes them if possible.

bool terminationcondition(int gamespace[8][6]);

This function is used to check whether the game has terminated at each stage in the gameplay.