

PROJECT REPORT

Team

Slot: 11

Group number: 6

Team members

Name	Roll number
Aakash Praliya (Leader)	140050012
Ashwin Rai	140020097
Ratanjot Singh	140020122
Satyendra Singh Rajpoot	140020092

About Project

This project report is about the game chain reaction. The game is made in C++. In order to provide graphical interface, Simple Direct Media Layer (SDL) library is used.

Game Description:

Chain reaction is a simple game based on reaction of molecules. In this game different players have different colored orbitals. The game consists of a maze having fixed number of slots in which players have to fill in their orbitals.

Orbitals combine to form molecules and when the number of orbitals in molecule is more than three then the molecule explodes into four individual orbitals.

Only same colored orbitals can combine to form molecule. Any orbital can collide with a molecule consisting of 3 orbitals resulting in explosion. The player who eliminates all orbitals except his own type will be declared as winner.

Overview of Program:

- *When the game file is executed then window appears showing the following options:*
 - *Play*
 - *Instructions*
 - *Quit*

- *When Play is selected then it shows another screen to give the number of rows in to be used in the maze of the game. Press any number key to include that number of games.*
- *When number of rows is decided then next window appears asking for the number of columns to be included in the game. Press any number to use that number of columns in the game.*
- *The next screen appears asking for the game mode. It has four options:*
 - *1Player*
 - *2Player*
 - *3Player*
 - *4Player*
- *When 1Player is selected then the user has to play with pc. By default the color of ball of user is red and that of pc is green. Powers are not available for 1Player mode.*
- *When 2Player is 2 Players have to play the game. The color of the balls is red and green.*
- *When 3 Player is chosen then 3 users play the game and the color of the balls is red, green, violet.*
- *When 4 Player is chosen then 4 users play the game and the color of the balls is red, green, violet and yellow.*

- **Players can fill in the slots with their orbitals.**

We have defined a 2d array to keep track of the number of orbitals in a particular slot. When a particular slot is filled then corresponding changes are made in the array corresponding to that particular slot.

Different sets of numbers are defined for different number of orbitals of different colours.

The table given below shows the sets of numbers:

Number of orbitals in slot	Corresponding number in array
Empty space(0)	0
1 red	1
2 red	2
3 red	3
1 green	8
2 green	9
3 green	10
1 violet	13
2 violet	14
3 violet	15
1 yellow	20
2 yellow	21
3 yellow	22

Powers	Corresponding number in array
Paintit	80
Frozen	-20
Maze Destructor	81

Functions of SDL library are used to display different images corresponding to different numbers in the array.

- ***When a particular slot has more than three balls, then the slot will explode into four orbitals.***

We have used chain function to perform chain reaction. The chain function will operate on the 2d array.

Chain function runs on the principle of recursion.

- ***The game will end when orbital of single kind will fill the maze completely.***

Functions used in the Game:

The game consist of following four functions

- **Show function**

Show function is used to show the mouse pointer on the screen. It uses dynamic storage of images to update the screen as the function progresses.

- **Show2 function**

Show2 function is responsible for updating the whole screen,ie, each and every grid as the game progresses. It scans the values of different elements of the main array and updates each grid corresponding to particular values.

- **Chain2 function**

Chain2 function is responsible for performing the chain reaction in the game. It is a recursive function. It uses the check function to check in the midst of chain reaction if a certain player has won. It also calls check function at the end to check that after the chain reaction if a player has won the game.

- **Check function**

Check function is responsible for checking that whether a certain player has won the game. In 2 Player and 3 Player mode it also checks whether a particular player has lost the game.

Some features of the game:

- **Powers**

The game features three powers which appear randomly anywhere in the grid. Powers appear after a certain period of time and stay for a certain period of time. Powers have been coded without using multithreading. The game is coded such that even after the sequential execution of the code, powers appear in the game with a delay and the delay does not hamper the performance of the player.

- **Infinite Loop Solved**

Sometimes in the game situations arise when there are infinite chain reactions. There are infinite recursions of the chain2 function. Therefore in order to solve this problem, check function is used in chain2 function. Check function checks that before starting any chain reaction, any player has already won the game and if so is the case then chain2 function will stop.

- **Variable Maze size**

The game provides the player the freedom to chose any maze size he wants. The maximum size of the maze is 10x10. But player can chose any number of rows less than 10 and any number of columns less than 10.

- ***Undo options***

The game features undo option. It can undo the last move performed.

Work allocation

- 1. Aakash will look for making basic functions used in the game like chain function etc. and integrating various functions into a single product. Aakash is assigned the task of creating chain function which will be used to perform the chain reaction in the game and also a function which will check the winning condition for different players. He is also responsible for including artificial intelligence in the game.*
- 2. Ashwin is responsible for inserting the use of mouse in the game. He is also responsible to show various effects on the maze corresponding to mouse motion.*
- 3. Satyendra will make a function to check the winning as well as losing condition for a player. He is also responsible to make a function to check that in 3 player or 4 player mode the players don't get a chance to play if once their orbs are over in the maze.*
- 4. Ratanjot will look for extending the game into multiplayer system. He is also responsible to make a function to update the game screen as the game progresses. He is responsible to take appropriate images to be included in the game.*

Scope of Project

The project is specifically designed for the use of teenagers and specifically the JIT Bombay students. Chain reaction is basically a fun oriented game which can run on any computer with most basic configuration of hardware and software. The rules of the game are easy to understand. People can play and enjoy this project. We also wanted to add animations to the project. Animations to this project can prove to be very useful. The game could become more interesting.

Future Prospects

After the completion of the product, various errors may be reported by the users. Post-production stage will thus involve maintaining the product for a certain period of times. Fixes to errors in the form of updates will be issued. The mini-game can also be ported as a mobile game and made available on tablets after suitable changes are made for increased adaptability. Also we briefly discussed to include internet features in the game like you can like the game on facebook or tweet about it. Such improvements can increase the scope of this project even further.