CS101

PROJECT:*CHAIN REACTION*

SRS DOCUMENT

INTRODUCTION:

Chain reaction is a strategy game for two players. The objective of chain reaction is to take control of the board by eliminating opponent's orbs. Whoever finally takes control over the board wins the game.

DEFINING THE GAME:

Players take it in turns to place their orbs in a cell. Once a cell has reached a specific number of orbs, the cell explodes adding each orb to the adjacent cells and claiming these cells for that player.

Players may only place their orbs in a blank cell or in a cell that contains orbs of his own colour. As soon as a player eliminates all his opponent's orbs wins the game.

FUNCTIONAL SPECIFICATIONS:

Header files:

simplecpp

cmath

Graphics:

- **initCanvas("name",w,h):**

  This creates a window of width 'w', height 'h' with title -name.

- **Rectangle r(cx,cy,LX,Ly):**
  This creates a rectangle centred at (cx,cy) and length Lx, breadth Ly.
- **Line l(x1,y1,x2,y2):**
  This creates a line with extremities (x1,y1),(x2,y2) on the window.
- **Circle c(cx,cy,r):**
  This creates a circle with radius 'r' and centred at (cx,cy).
- **Text t(cx,cy,"message"):**
  This is used to write text -message centred at (cx,cy) on the window.
- **Text t(cx,cy,number):**
  This is used to write -number-centred at (cx,cy) on the screen. The "number" can be in the form of an expression and it's value will be displayed on the screen.
- **s.setScale(factor):**
  This changes the size of shape "s" to desired scale.
- **getClick():**
  This function returns an integer value which is encoded form of co-ordinates of the point where mouse click is given on the window.

  If (x,y) are the integral co-ordinates of a point, then the function returns the value (65536*x+y).

- **s.hide():**
  This is used to hide a shape "s".

USER DEFINED FUNCTIONS:

- **main_program:**
  This is the main function of the program.
- **grid():**
  This function is used to draw an mxn grid.
- **geti(int getClick()):**
  This function returns an integer value "i".
- **getj(int getClick()):**
  This function returns an integer value "j".
- **scan():**
  This function scans the entire grid and checks for explosions.
- **score(int array[][]):**
  This function returns the sum of all elements in the array.

DATA SPECIFICATIONS:

In this program we will be using *simplecpp* graphics library.

Input: User input will be in the form of mouseclicks.

Output: The grid will be displayed on the screen throughout the game with circles shown in each cell as they are filled in.

The basic aim is to display windows, text, grid, circles and to record mouseclicks for input and output.

ALGORITHM:

- Create a window of size 800x600 with title "Game".
- Create text "CHAIN REACTION" at the centre of this window.
- Make the window visible for 5 seconds and close it.
- create another window of size 1000x800 with title "Menu".
- Create two rectangular tabs of size 400x200.
- Write text "PLAY" and "INSTRUCTIONS" in each of these rectangular tabs.
- When a click is given on the "INSRUCTIONS" tab, create a new window displaying the rules of the game.
- When the click is given on the "PLAY" tab, create another window titled "Chain Reaction".

- Now create a 8x8 grid of size 800x800 on this window.
- Declare two -2 dimensional arrays one of which stores the values of cells and the other the number of circles present in all the cells of grid. Also declare a -1 dimensional array to store the co-ordinates of clicked position in encoded form.
- Orbs/circles of radius 15 are put in the clicked cells in such a way that for every 'odd order' click, a 'Green' orb appears and for every 'even order' click, a 'Blue' orb appears in that cell.
- Also orbs of one colour can't be put in a cell with orb(s) of another colour.
- If a green orb is put in a cell, it is adjusted in that cell based on the previous number of green orbs in that cell. Meanwhile increase the value of cell and the number of circles in that cell by one.
- Similar is the case for blue orb except decreasing the cell value by one.
- At any instant, if, in any cell, there are same number of orbs as the number of adjacent cells, then explode that cell and add each of these orbs to the adjacent

cells and claiming opponent's orbs if any in that cell.

- Scan the grid repeatedly until none of the cells explodes.
- After each scan,

  if all the occupied cells contain orbs of same colour, then create a window titled "Game Over".

  Create text on this window declaring the winner orb and the score of that player and then close the window after 8 seconds.

  If orbs of different colour are present in the occupied cells, then continue the game.

-END OF SRS-