

2048

REQUIREMENTS:

- It requires the MS-DOS version 2.0 or greater

HEADER FILES USED:

❖ #include<graphics.h>

- ◆ `initgraph(int*,int*,char*)` – It initializes graphics window.
- ◆ `settextstyle(int,int,int)` – It sets the text font,direction in which text to be displayed and size of the text in graphics window.
- ◆ `settextjustify()` – It justifies the text in the graphics window.
- ◆ `outtextxy(int,int,char*)` – It prints the given text in given co-ordinates in graphics window.
- ◆ `getmaxx()` – It returns the maximum abscissa.
- ◆ `getmaxy()` – It returns the maximum ordinate.
- ◆ `setcolor(int)` – Make the succeeding outputs in the given colour.
- ◆ `setfillstyle(int,int)` – It sets the current fill colour and fill-pattern.
- ◆ `pieslice(int,int,int,int,int)` – It draws and fills a pie slice with the given radius.

- ◆ `cleardevice()` – It erases the entire graphics screen and moves the current position to (0,0).
- ◆ `closegraph()` – It closes the graphics window.

❖ `#include<conio.h>`

- ◆ `getch ()` – Reads a character directly from the console without buffer, and without echo.
- ◆ `getche ()` – Reads a character directly from the console without buffer, but with echo.
- ◆ `kbhit()` – It returns 1 if any key is pressed else returns 0.
- ◆ `clrscr ()` – Clears the screen.
- ◆ `gotoxy()` – Set the current position ,in default c++ output screen, to the given co-ordinates.

❖ `#include<dos.h>`

- ◆ `delay(int)` – It is used to suspend execution for a given time.
- ◆ `sound(int)` – It play a sound in the given frequency.
- ◆ `nosound()` – It stops the sound previously played by sound function.

❖ `#include<string.h>`

- ◆ `strcpy(char*,char*)` – It copy the second string to the second string.

❖ #include<stdlib.h>

- ◆ randomize() – It seeds rand function with the system time.
- ◆ exit(int) – It terminates the program..

❖ #include<stdio.h>

- ◆ sprintf(char*,...) – It writes the given string in the given buffer.

CLASS and OBJECT:

Class Element:

This is class represents a single block in the 2046 game.

Class object: Base is an two dimensional array of pointers of Element.Each element of Base points to a dynamic Element object.The object is used to invoke the member functions of the class Element to double its current value, move its position etc.

This class implements encapsulation.This allows to integrate the properties of single block and use them as a single entity.

- **Private Members:**

- int value
- int row
- int col
- int colour
- int size

- **Public Members:**

- static int trow
- static int tcol
- static int boardLeft
- static int boardTop;
- static Board *base
- int getValue()
- void Double()
- int fallUp()
- int fallDown()
- int fallRight()
- int fallLeft()
- void disp()

Class Board:

This is class represents the board in the 2046 game.

Class object: Game is an object of Board. The object is used to invoke the member functions of the class Board to display,moveup,moveright etc.

This class implements encapsulation.This allows to integrate the properties of the whole board and use them as a single entity.

- **Private Members:**

- int trow
- int tcol
- int size
- int colour

- int right
- int left
- int top
- int bottom
- Element ***base
- int row

- Public Members:

- void drawBoard()
- void showBoard()
- int getValue(int,int)
- void merge(int,int,int,int)
- int isFree(int,int)
- int createElement()
- void swipeRight()
- void swipeLeft()
- void swipeUp()
- void swipeDown()
- void move(int,int,int,int)
- void play()

FUNCTIONS USED:

- Member functions:

Element class:

- int getValue()

It returns the value of the Element(single block in the board).

➤ void Double()

It doubles the value of the Element.

➤ int fallUp()

It changes the co-ordinates of the Element appropriately.

➤ int fallDown()

It changes the co-ordinates of the Element appropriately.

➤ int fallRight()

It changes the co-ordinates of the Element appropriately.

➤ int fallLeft()

It changes the co-ordinates of the Element appropriately.

➤ void disp()

It displays the block in graphic window with appropriate settings.

Board class:

➤ void drawBoard()

It draws the border for the game board.

➤ void showBoard()

It calls drawBoard() and displays the individual blocks.

➤ int getValue(int,int)

It returns the value of the element at the given co-ordinate.

➤ void merge(int,int,int,int)

It merges the blocks in the given co-ordinates.

➤ int isFree(int,int)

It returns 1 if there is a block in the given co-ordinate else returns 0.

➤ int createElement()

It creates a new block with random value.

➤ void swipeRight()

It moves all the blocks to the right.

➤ void swipeLeft()

It moves all the blocks to the left.

➤ void swipeUp()

It moves all the blocks in the upward direction.

➤ void swipeDown()

It moves all the block in the downward direction.

➤ void move(int,int,int,int)

It moves the element specified from its current position to the new specified position.

➤ void play()

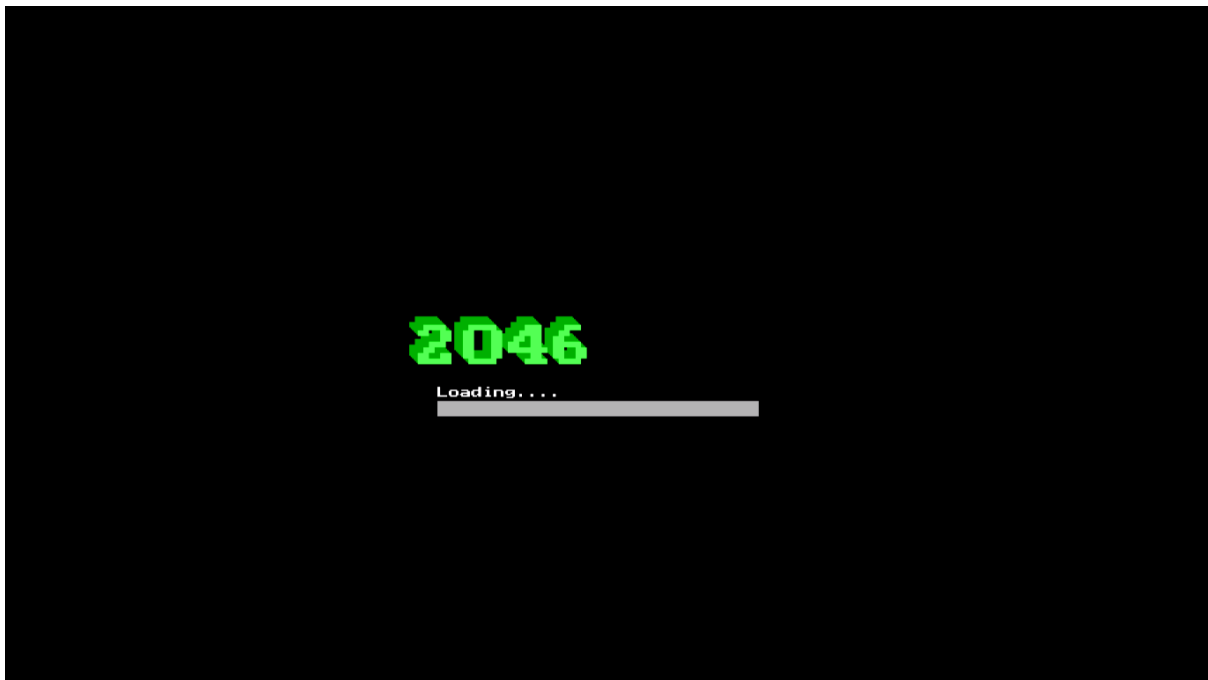
It opens the graphics windows, gets input and calls display function.

- **Global Functions:**

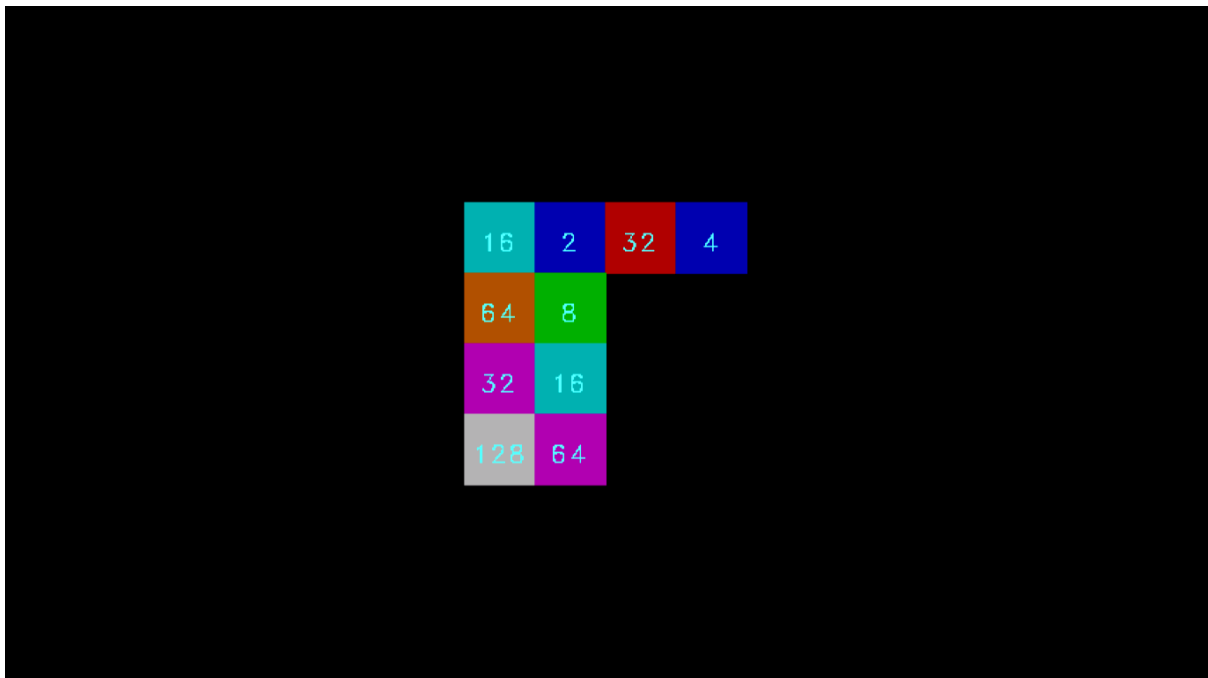
- `void text3D(int,int,int,int,int,int,char):`

- Display the text given to the function in the given location.

OUTPUT:



2048 game opening screen.



2048 gameplay.

SHORTCOMINGS:

- In 2046, the transition screen is not smooth.