```
from tkinter import *
from tkinter import messagebox
import random


class Board:
    bg_color={

        '2': '#eee4da',
        '4': '#ede0c8',
        '8': '#edc850',
        '16': '#edc53f',
        '32': '#f67c5f',
        '64': '#f65e3b',
        '128': '#edcf72',
        '256': '#edcc61',
        '512': '#f2b179',
        '1024': '#f59563',
        '2048': '#edc22e',
    }
    color={
         '2': '#776e65',
        '4': '#f9f6f2',
        '8': '#f9f6f2',
        '16': '#f9f6f2',
        '32': '#f9f6f2',
        '64': '#f9f6f2',
        '128': '#f9f6f2',
        '256': '#f9f6f2',
        '512': '#776e65',
        '1024': '#f9f6f2',
        '2048': '#f9f6f2',
```

```
}


def __init__(self):
    self.n=4
    self.window=Tk()
    self.window.title('ProjectGurukul 2048 Game')
    self.gameArea=Frame(self.window,bg= 'azure3')
    self.board=[]
    self.gridCell=[[0]*4 for i in range(4)]
    self.compress=False
    self.merge=False
    self.moved=False
    self.score=0

    for i in range(4):
        rows=[]
        for j in range(4):
            l=Label(self.gameArea,text='',bg='azure4',
            font=('arial',22,'bold'),width=4,height=2)
            l.grid(row=i,column=j,padx=7,pady=7)


            rows.append(l);
        self.board.append(rows)
    self.gameArea.grid()


def reverse(self):
    for ind in range(4):
        i=0
        j=3
        while(i<j):
            self.gridCell[ind][i],self.gridCell[ind][j]=self.gridCell[ind][j],self.gridCell[ind][i]
```

```
            i+=1
            j-=1

    def transpose(self):
        self.gridCell=[list(t)for t in zip(*self.gridCell)]

    def compressGrid(self):
        self.compress=False
        temp=[[0] *4 for i in range(4)]
        for i in range(4):
            cnt=0
            for j in range(4):
                if self.gridCell[i][j]!=0:
                    temp[i][cnt]=self.gridCell[i][j]
                    if cnt!=j:
                        self.compress=True
                    cnt+=1
        self.gridCell=temp

    def mergeGrid(self):
        self.merge=False
        for i in range(4):
            for j in range(4 - 1):
                if self.gridCell[i][j] == self.gridCell[i][j + 1] and self.gridCell[i][j] != 0:
                    self.gridCell[i][j] *= 2
                    self.gridCell[i][j + 1] = 0
                    self.score += self.gridCell[i][j]
                    self.merge = True

    def random_cell(self):
        cells=[]
```

```
    for i in range(4):
        for j in range(4):
            if self.gridCell[i][j] == 0:
                cells.append((i, j))
    curr=random.choice(cells)
    i=curr[0]
    j=curr[1]
    self.gridCell[i][j]=2

def can_merge(self):
    for i in range(4):
        for j in range(3):
            if self.gridCell[i][j] == self.gridCell[i][j+1]:
                return True

    for i in range(3):
        for j in range(4):
            if self.gridCell[i+1][j] == self.gridCell[i][j]:
                return True
    return False

def paintGrid(self):
    for i in range(4):
        for j in range(4):
            if self.gridCell[i][j]==0:
                self.board[i][j].config(text='',bg='azure4')
            else:
                self.board[i][j].config(text=str(self.gridCell[i][j]),
                bg=self.bg_color.get(str(self.gridCell[i][j])),
                fg=self.color.get(str(self.gridCell[i][j])))
```

```
class Game:
    def __init__(self,gamepanel):
        self.gamepanel=gamepanel
        self.end=False
        self.won=False

    def start(self):
        self.gamepanel.random_cell()
        self.gamepanel.random_cell()
        self.gamepanel.paintGrid()
        self.gamepanel.window.bind('<Key>', self.link_keys)
        self.gamepanel.window.mainloop()

    def link_keys(self,event):
        if self.end or self.won:
            return

        self.gamepanel.compress = False
        self.gamepanel.merge = False
        self.gamepanel.moved = False

        presed_key=event.keysym

        if presed_key=='Up':
            self.gamepanel.transpose()
            self.gamepanel.compressGrid()
            self.gamepanel.mergeGrid()
            self.gamepanel.moved = self.gamepanel.compress or self.gamepanel.merge
            self.gamepanel.compressGrid()
            self.gamepanel.transpose()
```

```
elif presed_key=='Down':
    self.gamepanel.transpose()
    self.gamepanel.reverse()
    self.gamepanel.compressGrid()
    self.gamepanel.mergeGrid()
    self.gamepanel.moved = self.gamepanel.compress or self.gamepanel.merge
    self.gamepanel.compressGrid()
    self.gamepanel.reverse()
    self.gamepanel.transpose()

elif presed_key=='Left':
    self.gamepanel.compressGrid()
    self.gamepanel.mergeGrid()
    self.gamepanel.moved = self.gamepanel.compress or self.gamepanel.merge
    self.gamepanel.compressGrid()

elif presed_key=='Right':
    self.gamepanel.reverse()
    self.gamepanel.compressGrid()
    self.gamepanel.mergeGrid()
    self.gamepanel.moved = self.gamepanel.compress or self.gamepanel.merge
    self.gamepanel.compressGrid()
    self.gamepanel.reverse()
else:
    pass

self.gamepanel.paintGrid()
print(self.gamepanel.score)

flag=0
for i in range(4):
```

```
            for j in range(4):
                if(self.gamepanel.gridCell[i][j]==2048):
                    flag=1
                    break

        if(flag==1): #found 2048
            self.won=True
            messagebox.showinfo('2048', message='You Wonnn!!')
            print("won")
            return

        for i in range(4):
            for j in range(4):
                if self.gamepanel.gridCell[i][j]==0:
                    flag=1
                    break

        if not (flag or self.gamepanel.can_merge()):
            self.end=True
            messagebox.showinfo('2048','Game Over!!!')
            print("Over")

        if self.gamepanel.moved:
            self.gamepanel.random_cell()

        self.gamepanel.paintGrid()



gamepanel =Board()
game2048 = Game( gamepanel)
game2048.start()
```

Program Output