**SIES Graduate School of Technology**

**Department of Computer Engineering**

**OSTL Mini Project Report on**

**‘2048 GAME’**

**(AY 2019-20)**

**GROUP MEMBERS:**

118A1058- NIKHITHA JAYAPRAKASH

118A1068- RAO BHAGYASHREE VENKATESHKUMAR

118A1062- PAVITHRA RAMACHANDRAN

**INTRODUCTION:**

2048 is a single-player sliding block puzzle game, designed by the Italian web developer Gabriele Cirulli. 2048 was originally written in [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). The game's objective is to slide numbered tiles on a grid to combine them to create a tile with the number 2048.

**PROJECT GOALS:**

The goal is to develop the 2048 original game.

**Features:**

* Played on a 4×4 grid, with numbered tiles that slide smoothly when a player moves them using the 4 arrow keys.
* When the user makes a tile of 2048, “YOU WON!” is displayed.
* When no new tiles can be added (Grid is completely filled) or no more tiles can be merged on collosion, then “GAME OVER” is displayed.

**GAME LOGIC:**

* Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4.
* Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid.
* If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided.
* The resulting tile cannot merge with another tile again in the same move.
* If a move causes three consecutive tiles of the same value to slide together, only the two tiles farthest along the direction of motion will combine.
* If all four spaces in a row or column are filled with tiles of the same value, a move parallel to that row/column will combine the first two and last two.
* A scoreboard on the top of the window keeps track of the user's score. The user's score starts at zero, and is increased whenever two tiles combine, by the value of the new tile.

**GAME CODE:**

import tkinter as tk

import random

class Game(tk.Tk):

board = []

new\_tile= [2,2,2,2,2,2,4]

score = 0

highscore = 0

scorestring = 0

highscorestring = 0

def \_\_init\_\_(self, \*args, \*\*kwargs):

tk.Tk.\_\_init\_\_(self, \*args, \*\*kwargs)

self.scorestring = tk.StringVar(self)

self.scorestring.set("0")

self.highscorestring = tk.StringVar(self)

self.highscorestring.set("0")

self.create\_widgets()

self.canvas = tk.Canvas(self, width=410, height=410, borderwidth=5)

self.canvas.pack(fill="both", expand="false")

self.new\_game()

def addNewTile(self):

index = random.randint(0,6)

x = -1

y = -1

while self.isFull() == False:

x = random.randint(0,3)

y = random.randint(0,3)

if (self.board[x][y] == 0):

self.board[x][y] = self.new\_tile[index]

x1 = y\*105

y1 = x\*105

x2 = x1 + 105 - 5

y2 = y1 + 105 - 5

num = self.board[x][y]

if num == 2:

self.square[x,y] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#81ECC2", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="#f78a8a", text="2")

elif num == 4:

self.square[x,y] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#5EECB5", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="#f78a8a", text="4")

break

def isFull(self):

for i in range(0,4):

for j in range(0,4):

if (self.board[i][j] == 0):

return False

return True

def printboard(self):

wide=105

high= 105

self.square = {}

for col in range(4):

for row in range(4):

x1 = col\*wide

y1 = row\*high

x2 = x1 + wide - 5

y2 = y1 + high - 5

num = self.board[row][col]

if num == 0:

self.print0(row, col, x1, y1, x2, y2)

elif num == 2 or num==4 or num==8 or num==16 or num==32 or num==64 or num==128 or num==256 or num==512 or num==1024 or num==2048:

self.printnum(row, col, x1, y1, x2, y2,num)

def print0(self, row, col, x1, y1, x2, y2):

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#f5f5f5", outline="")

def printnum(self, row, col, x1, y1, x2, y2,num):

if num==2:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#81ECC2", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="#494949", text="2")

elif num==4:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#5EECB5", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="#494949", text="4")

elif num==8:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#56D9A6", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="8")

elif num==16:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#42C894", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="16")

elif num==32:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#3BBD8B", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="32")

elif num==64:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#35AC7D", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="64")

elif num==128:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#299E70", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="128")

elif num==256:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#269368", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="256")

elif num==512:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#22805B", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="512")

elif num==1024:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#1C7150", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="1024")

elif num==2048:

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#185E43", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36), fill="white", text="2048")

def create\_widgets(self):

self.buttonframe = tk.Frame(self)

self.buttonframe.grid(row=2, column=0, columnspan=4)

tk.Button(self.buttonframe, text = "New Game",command=self.new\_game).grid(row=0, column=0)

tk.Label(self.buttonframe, text = "Score:").grid(row=0, column=1)

tk.Label(self.buttonframe, textvariable=self.scorestring).grid(row=0, column=2)

tk.Label(self.buttonframe, text = "Record:").grid(row=0, column=3)

tk.Label(self.buttonframe, textvariable=self.highscorestring).grid(row=0, column=4)

self.buttonframe.pack()

def keyPressed(self,event):

shift = 0

if event.keysym == 'Down':

for j in range(0,4):

shift = 0

for i in range(3,-1,-1):

if self.board[i][j] == 0:

shift += 1

else:

if i - 1 >= 0 and self.board[i-1][j] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i-1][j] = 0

elif i - 2 >= 0 and self.board[i-1][j] == 0 and self.board[i-2][j] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i-2][j] = 0

elif i == 3 and self.board[2][j] + self.board[1][j] == 0 and self.board[0][j] == self.board[3][j]:

self.board[3][j] \*= 2

self.score += self.board[3][j]

self.board[0][j] = 0

if shift > 0:

self.board[i+shift][j] = self.board[i][j]

self.board[i][j] = 0

self.printboard()

self.addNewTile()

self.isOver()

elif event.keysym == 'Right':

for i in range(0,4):

shift = 0

for j in range(3,-1,-1):

if self.board[i][j] == 0:

shift += 1

else:

if j - 1 >= 0 and self.board[i][j-1] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i][j-1] = 0

elif j - 2 >= 0 and self.board[i][j-1] == 0 and self.board[i][j-2] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i][j-2] = 0

elif j == 3 and self.board[i][2] + self.board[i][1] == 0 and self.board[0][j] == self.board[3][j]:

self.board[i][3] \*= 2

self.score += self.board[i][3]

self.board[i][0] = 0

if shift > 0:

self.board[i][j+shift] = self.board[i][j]

self.board[i][j] = 0

self.printboard()

self.addNewTile()

self.isOver()

elif event.keysym == 'Left':

for i in range(0,4):

shift = 0

for j in range(0,4):

if self.board[i][j] == 0:

shift += 1

else:

if j + 1 < 4 and self.board[i][j+1] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i][j+1] = 0

elif j + 2 < 4 and self.board[i][j+1] == 0 and self.board[i][j+2] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i][j+2] = 0

elif j == 0 and self.board[i][1] + self.board[i][2] == 0 and self.board[i][3] == self.board[i][0]:

self.board[i][0] \*= 2

self.score += self.board[i][0]

self.board[i][3] = 0

if shift > 0:

self.board[i][j-shift] = self.board[i][j]

self.board[i][j] = 0

self.printboard()

self.addNewTile()

self.isOver()

elif event.keysym == 'Up':

for j in range(0,4):

shift = 0

for i in range(0,4):

if self.board[i][j] == 0:

shift += 1

else:

if i + 1 < 4 and self.board[i+1][j] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i+1][j] = 0

elif i + 2 < 4 and self.board[i+1][j] == 0 and self.board[i+2][j] == self.board[i][j]:

self.board[i][j] \*= 2

self.score += self.board[i][j]

self.board[i+2][j] = 0

elif i == 0 and self.board[1][j] + self.board[2][j] == 0 and self.board[3][j] == self.board[0][j]:

self.board[0][j] \*= 2

self.score += self.board[0][j]

self.board[3][j] = 0

if shift > 0:

self.board[i-shift][j] = self.board[i][j]

self.board[i][j] = 0

self.printboard()

self.addNewTile()

self.isOver()

self.scorestring.set(str(self.score))

if self.score > self.highscore:

self.highscore = self.score

self.highscorestring.set(str(self.highscore))

def new\_game(self):

self.score = 0

self.scorestring.set("0")

self.board = []

self.board.append([0,0,0,0])

self.board.append([0,0,0,0])

self.board.append([0,0,0,0])

self.board.append([0,0,0,0])

while True:

x = random.randint(0,3)

y = random.randint(0,3)

if (self.board[x][y] == 0):

self.board[x][y] = 2

break

index = random.randint(0,6)

while self.isFull() == False:

x = random.randint(0,3)

y = random.randint(0,3)

if (self.board[x][y] == 0):

self.board[x][y] = self.new\_tile[index]

break

self.printboard()

def isOver(self):

for i in range(0,4):

for j in range(0,4):

if (self.board[i][j] == 2048):

game = [["Y", "O", "U", "",],["W", "O", "N", "!"], ["", "", "", ""], ["", "", "", ""]]

self.size(game)

for i in range(0,4):

for j in range(0,4):

if (self.board[i][j] == 0):

return False

for i in range(0,4):

for j in range(0,3):

if (self.board[i][j] == self.board[i][j+1]):

return False

for j in range(0,4):

for i in range(0,3):

if self.board[i][j] == self.board[i+1][j]:

return False

game = [["G", "A", "M", "E",],["O", "V", "E", "R"], ["", "", "", ""], ["", "", "", ""]]

self.size(game)

return True

def size(self,game):

wide = 105

high = 105

self.square = {}

for col in range(4):

for row in range(4):

x1 = col\*wide

y1 = row\*high

x2 = x1 + wide - 5

y2 = y1 + high - 5

self.square[row,col] = self.canvas.create\_rectangle(x1,y1,x2,y2, fill="#81ECC2", outline="")

self.canvas.create\_text((x1 + x2)/2, (y1+y2)/2,font=("Arial", 36),fill="#494949", text=game[row][col])

if \_\_name\_\_ == "\_\_main\_\_":

app = Game()

app.bind\_all('<Key>', app.keyPressed)

app.wm\_title("2048")

app.minsize(420,450)

app.maxsize(420,450)

app.mainloop()

**CODE EXPLANATION:**

a) We have used the tkinter module in python. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

b) \*args and \*\*kwargs are special syntax that are used in functions to pass a variable number of arguments to a function.

We have used widgets such as buttons, canvas, label and text.

c) Initially we have a 4\*4 grid with 2 tiles at random position. When we press any of the 4 arrow keys, the tiles move in the chosen direction. We use the keypress event of Tkinter module to accept the arrow keys. Tkinter also allows you to create bindings on the class and application level; here we use **bind\_all** method to create binding for the keypress event.

d) On pressing any key:

i) If board is not full(checked using isFull function)new tile is added to the board (addNewTile function handles this).

ii) The new tile will be either with value 2 or 4, generated randomly on any random position of the grid, with the font colour as pink.

iii) For the tiles to move in the chosen direction, we shift each tile either by shifting or by merging and shifting.

**CONCLUSION:**

Thus we have successfully implemented the 2048 game in python using Tkinter module.