**Program for Chain reaction game:**

import pygame

import sys

import time

from math import \*

# Initialization of Pygame

pygame.init()

# Block = Size of each block

blocks = 55

# Order of the matrix

sqFactor = chnrxnv4.getsize()

noPlayers = int(input("How many players?"))

mode = int(input("Select Mode: Eliminating(1) or non-eliminating(0)?"))

width = blocks \* sqFactor

height = blocks \* sqFactor

display = pygame.display.set\_mode((width, height))

# Sets the title of the game screen.

pygame.display.set\_caption("Chain Reaction %dP Mode" % noPlayers)

# Define the clock.

clock = pygame.time.Clock()

# Define a standard font

font = pygame.font.SysFont("Calibri", 30)

# Predefine all Colors

background = (21, 67, 96)

border = (208, 211, 212)

red = (231, 76, 60)

white = (244, 246, 247)

violet = (255, 0, 255)

yellow = (204, 204, 0)

green = (88, 214, 141)

orange = (255, 153, 0)

darkBlue = (26, 26, 255)

lnm\_green = (102, 255, 51)

blue = (102, 204, 255)

# List storing colours of different players

playerColor = [red, green, violet, yellow, orange, darkBlue, lnm\_green, blue]

# List storing scores of all players.

score = []

for i in range(noPlayers):

# Initialise all scores to 0

score.append(0)

# List storing the colour of players playing the game

players = []

for i in range(noPlayers):

# Assign colours to each player

players.append(playerColor[i])

d = blocks / 2 - 2

# Calculating the number rows and columns based on height and width / block length

cols = int(width / blocks)

rows = int(height / blocks)

# Quit or Close the Game Window

def close():

pygame.quit()

sys.exit()

# Class for Each Spot in Grid

class Spot():

def \_\_init\_\_(self):

self.color = border

# List of neighbours

self.neighbors = []

# List of atoms

self.noAtoms = 0

def addNeighbors(self, i, j):

if i > 0:

self.neighbors.append(grid[i - 1][j])

if i < rows - 1:

self.neighbors.append(grid[i + 1][j])

if j < cols - 1:

self.neighbors.append(grid[i][j + 1])

if j > 0:

self.neighbors.append(grid[i][j - 1])

# Initializing the Grid with "Empty or 0"

def initializeGrid():

global grid, score, players

grid = [[] for \_ in range(cols)]

for i in range(cols):

for j in range(rows):

newObj = Spot()

grid[i].append(newObj)

for i in range(cols):

for j in range(rows):

grid[i][j].addNeighbors(i, j)

# Draw the Grid in Pygame Window

def drawGrid(currentIndex):

r = 0

c = 0

for i in range(int(width / blocks)):

r += blocks

c += blocks

# Line fuction parameters are: window,color,start\_cords,end\_cords

pygame.draw.line(display, players[currentIndex], (c, 0), (c, height))

pygame.draw.line(display, players[currentIndex], (0, r), (width, r))

# Draw the Present Situation of Grid

def showPresentGrid(vibrate=1):

r = -blocks # xcoord

c = -blocks # ycoord

padding = 2

for i in range(cols):

r += blocks

c = -blocks

for j in range(rows):

c += blocks

if grid[i][j].noAtoms == 0:

grid[i][j].color = border

elif grid[i][j].noAtoms == 1:

pygame.draw.ellipse(display, grid[i][j].color,

(r + blocks / 2 - d / 2 + vibrate, c + blocks / 2 - d / 2, d, d))

elif grid[i][j].noAtoms == 2:

pygame.draw.ellipse(display, grid[i][j].color, (r + 5, c + blocks / 2 - d / 2 - vibrate, d, d))

pygame.draw.ellipse(display, grid[i][j].color,

(r + d / 2 + blocks / 2 - d / 2 + vibrate, c + blocks / 2 - d / 2, d, d))

elif grid[i][j].noAtoms == 3:

angle = 90

x = r + (d / 2) \* cos(radians(angle)) + blocks / 2 - d / 2

y = c + (d / 2) \* sin(radians(angle)) + blocks / 2 - d / 2

pygame.draw.ellipse(display, grid[i][j].color, (x - vibrate, y, d, d))

x = r + (d / 2) \* cos(radians(angle + 90)) + blocks / 2 - d / 2

y = c + (d / 2) \* sin(radians(angle + 90)) + 5

pygame.draw.ellipse(display, grid[i][j].color, (x + vibrate, y, d, d))

x = r + (d / 2) \* cos(radians(angle - 90)) + blocks / 2 - d / 2

y = c + (d / 2) \* sin(radians(angle - 90)) + 5

pygame.draw.ellipse(display, grid[i][j].color, (x - vibrate, y, d, d))

pygame.display.update()

# Increase the Atom when Clicked

def addAtom(i, j, color):

grid[i][j].noAtoms += 1

grid[i][j].color = color

def checkbomb():

global grid

global cols

flag = 0

for i in range(cols):

for j in range(rows):

if grid[i][j].noAtoms >= len(grid[i][j].neighbors):

flag = 1

color = grid[i][j].color

break

if flag == 0:

return 0

else:

return 1

def cutrecursion(colour):

global grid

global cols

flag = 1

for i in range(cols):

for j in range(rows):

if grid[i][j].color != colour:

flag = 0

break

if flag == 0:

return 0

else:

return 1

def updategrid(color, vibrate):

global grid

while checkbomb() == 1 and cutrecursion(color) == 0:

for i in range(cols):

for j in range(rows):

if grid[i][j].noAtoms >= len(grid[i][j].neighbors):

grid[i][j].noAtoms = grid[i][j].noAtoms % len(grid[i][j].neighbors)

for m in range(len(grid[i][j].neighbors)):

grid[i][j].neighbors[m].noAtoms += 1

grid[i][j].neighbors[m].color = grid[i][j].color

# Checking if Any Player has WON!

def isPlayerInGame():

global score

playerScore = []

for i in range(noPlayers):

playerScore.append(0)

for i in range(cols):

for j in range(rows):

for k in range(noPlayers):

if grid[i][j].color == players[k]:

playerScore[k] += grid[i][j].noAtoms

score = playerScore[:]

def scorecalc():

global score

global noPlayers

playerScore = []

for i in range(noPlayers):

playerScore.append(0)

for i in range(cols):

for j in range(rows):

for k in range(noPlayers):

if grid[i][j].color == players[k]:

playerScore[k] += grid[i][j].noAtoms

score = playerScore[:]

# GAME OVER

def gameOver(playerIndex):

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

close()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_q:

close()

if event.key == pygame.K\_r:

gameLoop()

text = font.render("Player %d Won!" % (playerIndex + 1), True, white)

text2 = font.render("Press \'r\' to Reset!", True, white)

display.blit(text, (width / 3, height / 3))

display.blit(text2, (width / 3, height / 2))

pygame.display.update()

clock.tick(60)

def checkWon():

num = 0

for i in range(noPlayers):

if score[i] == 0:

num += 1

if num == noPlayers - 1:

for i in range(noPlayers):

if score[i]:

return i

return 9999

# Main Loop

def gameLoop():

global noPlayers

global score

global mode

while chnrxnv4.getsignal==0:

pass

initializeGrid()

loop = True

turns = 0

currentPlayer = 0

vibrate = 2.0

while loop:

for event in pygame.event.get():

if event.type == pygame.QUIT:

close()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_q:

close()

if event.type == pygame.MOUSEBUTTONDOWN:

x, y = pygame.mouse.get\_pos()

i = int(x / blocks)

j = int(y / blocks)

if grid[i][j].color == players[currentPlayer] or grid[i][j].color == border:

turns += 1

addAtom(i, j, players[currentPlayer])

updategrid(players[currentPlayer], vibrate)

currentPlayer += 1

if currentPlayer == noPlayers:

currentPlayer = 0

if turns > noPlayers:

scorecalc()

while score[currentPlayer] == 0 and mode == 1:

currentPlayer += 1

if currentPlayer == noPlayers:

currentPlayer = 0

if turns >= noPlayers:

isPlayerInGame()

display.fill(background)

# Vibrate the Atoms in their Cells

vibrate \*= -1

drawGrid(currentPlayer)

showPresentGrid(vibrate)

# pygame.display.update()

res = checkWon()

if res < 9999:

gameOver(res)

clock.tick(25)

**Program for GUI:**

from PyQt5 import QtCore, QtGui, QtWidgets

import pygame

import sys

import time

from math import \*

class Ui\_mainWindow(object):

size=5

pcount=2

mode=1

signal=0

def setupUi(self, mainWindow):

mainWindow.setObjectName("mainWindow")

mainWindow.resize(1920, 1200) # 800,599

font = QtGui.QFont()

font.setFamily("Segoe Script")

font.setPointSize(10)

font.setBold(True)

font.setWeight(75)

mainWindow.setFont(font)

self.centralwidget = QtWidgets.QWidget(mainWindow)

self.centralwidget.setObjectName("centralwidget")

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setGeometry(QtCore.QRect(0, 0, 1920, 1200)) # -8-8 831,611

font = QtGui.QFont()

font.setPointSize(20)

self.label.setFont(font)

self.label.setText("")

self.label.setPixmap(QtGui.QPixmap("photos/t1.jpg"))

self.label.setObjectName("label")

self.label\_2 = QtWidgets.QLabel(self.centralwidget)

self.label\_2.setGeometry(QtCore.QRect(100, 50, 900, 120)) # 611,91

palette = QtGui.QPalette()

brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)

brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Text, brush)

brush = QtGui.QBrush(QtGui.QColor(120, 120, 120))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)

self.label\_2.setPalette(palette)

font = QtGui.QFont()

font.setFamily("Segoe Print")

font.setPointSize(48)

font.setBold(False)

font.setWeight(50)

self.label\_2.setFont(font)

self.label\_2.setObjectName("label\_2")

self.label\_3 = QtWidgets.QLabel(self.centralwidget)

self.label\_3.setGeometry(QtCore.QRect(50, 350, 370, 71))

self.label\_3.setObjectName("label\_3")

self.label\_4 = QtWidgets.QLabel(self.centralwidget)

self.label\_4.setGeometry(QtCore.QRect(500, 300, 175, 50))

self.label\_4.setObjectName("label\_4")

self.label\_5 = QtWidgets.QLabel(self.centralwidget)

self.label\_5.setGeometry(QtCore.QRect(500, 400, 175, 50))

self.label\_5.setObjectName("label\_5")

self.label\_6 = QtWidgets.QLabel(self.centralwidget)

self.label\_6.setGeometry(QtCore.QRect(500, 500, 175, 50))

self.label\_6.setObjectName("label\_6")

self.label\_7 = QtWidgets.QLabel(self.centralwidget)

self.label\_7.setGeometry(QtCore.QRect(500, 600, 165, 50))

self.label\_7.setObjectName("label\_7")

self.label\_8 = QtWidgets.QLabel(self.centralwidget)

self.label\_8.setGeometry(QtCore.QRect(500, 700, 165, 50))

self.label\_8.setObjectName("label\_8")

self.label\_9 = QtWidgets.QLabel(self.centralwidget)

self.label\_9.setGeometry(QtCore.QRect(500, 800, 165, 50))

self.label\_9.setObjectName("label\_9")

self.label\_10 = QtWidgets.QLabel(self.centralwidget)

self.label\_10.setGeometry(QtCore.QRect(500, 890, 175, 50))

self.label\_10.setObjectName("label\_10")

self.label\_11 = QtWidgets.QLabel(self.centralwidget)

self.label\_11.setGeometry(QtCore.QRect(50, 900, 400, 200)) # GAME MODE

self.label\_11.setObjectName("label\_11")

self.label\_12 = QtWidgets.QLabel(self.centralwidget)

self.label\_12.setGeometry(QtCore.QRect(500, 950, 300, 100))

self.label\_12.setObjectName("label\_12")

self.label\_13 = QtWidgets.QLabel(self.centralwidget)

self.label\_13.setGeometry(QtCore.QRect(500, 1050, 300, 100))

self.label\_13.setObjectName("label\_13")

self.label\_14 = QtWidgets.QLabel(self.centralwidget)

self.label\_14.setGeometry(QtCore.QRect(1100, 750, 300, 100))

self.label\_14.setObjectName("label\_14")

self.label\_15 = QtWidgets.QLabel(self.centralwidget)

self.label\_15.setGeometry(QtCore.QRect(1450, 720, 400, 300))

self.label\_15.setObjectName("label\_15")

self.label\_16 = QtWidgets.QLabel(self.centralwidget)

self.label\_16.setGeometry(QtCore.QRect(1450, 820, 400, 300))

self.label\_16.setObjectName("label\_16")

self.label\_17 = QtWidgets.QLabel(self.centralwidget)

self.label\_17.setGeometry(QtCore.QRect(1450, 920, 400, 300))

self.label\_17.setObjectName("label\_17")

self.label\_18 = QtWidgets.QLabel(self.centralwidget)

self.label\_18.setGeometry(QtCore.QRect(70, 210, 1400, 30))

self.label\_18.setObjectName("label\_18")

self.label\_19 = QtWidgets.QLabel(self.centralwidget)

self.label\_19.setGeometry(QtCore.QRect(1380, 330, 250, 278))

self.label\_19.setText("")

self.label\_19.setPixmap(QtGui.QPixmap("photos/play2.png"))

self.label\_19.setObjectName("label\_19")

self.label\_20 = QtWidgets.QLabel(self.centralwidget)

self.label\_20.setGeometry(QtCore.QRect(1300, 650, 400, 60))

self.label\_20.setObjectName("label\_20")

self.widget = QtWidgets.QWidget(self.centralwidget)

self.widget.setGeometry(QtCore.QRect(460, 280, 71, 700)) # 230 190 21 271

self.widget.setObjectName("widget")

self.verticalLayout\_4 = QtWidgets.QVBoxLayout(self.widget)

self.verticalLayout\_4.setContentsMargins(10, 0, 0, 0)

self.verticalLayout\_4.setObjectName("verticalLayout\_4")

self.radioButton = QtWidgets.QRadioButton(self.widget)

palette = QtGui.QPalette()

brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)

brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)

brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Text, brush)

brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ButtonText, brush)

brush = QtGui.QBrush(QtGui.QColor(120, 120, 120))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)

brush = QtGui.QBrush(QtGui.QColor(120, 120, 120))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ButtonText, brush)

self.radioButton.setPalette(palette)

font = QtGui.QFont()

font.setPointSize(20)

self.radioButton.setFont(font)

self.radioButton.setText("")

self.radioButton.setObjectName("radioButton")

self.verticalLayout\_4.addWidget(self.radioButton)

self.radioButton\_3 = QtWidgets.QRadioButton(self.widget)

self.radioButton\_3.setEnabled(True)

self.radioButton\_3.setText("")

self.radioButton\_3.setObjectName("radioButton\_3")

self.verticalLayout\_4.addWidget(self.radioButton\_3)

self.radioButton\_4 = QtWidgets.QRadioButton(self.widget)

self.radioButton\_4.setText("")

self.radioButton\_4.setObjectName("radioButton\_4")

self.verticalLayout\_4.addWidget(self.radioButton\_4)

self.radioButton\_2 = QtWidgets.QRadioButton(self.widget)

self.radioButton\_2.setObjectName("radioButton\_2")

self.verticalLayout\_4.addWidget(self.radioButton\_2)

self.radioButton\_5 = QtWidgets.QRadioButton(self.widget)

self.radioButton\_5.setObjectName("radioButton\_5")

self.verticalLayout\_4.addWidget(self.radioButton\_5)

self.radioButton\_6 = QtWidgets.QRadioButton(self.widget)

self.radioButton\_6.setText("")

self.radioButton\_6.setObjectName("radioButton\_6")

self.verticalLayout\_4.addWidget(self.radioButton\_6)

self.radioButton\_7 = QtWidgets.QRadioButton(self.widget)

self.radioButton\_7.setText("")

self.radioButton\_7.setObjectName("radioButton\_7")

self.verticalLayout\_4.addWidget(self.radioButton\_7)

self.widget1 = QtWidgets.QWidget(self.centralwidget)

self.widget1.setGeometry(QtCore.QRect(460, 950, 21, 200)) # 230 480 21 53

self.widget1.setObjectName("widget1")

self.verticalLayout\_5 = QtWidgets.QVBoxLayout(self.widget1)

self.verticalLayout\_5.setContentsMargins(0, 0, 0, 0)

self.verticalLayout\_5.setObjectName("verticalLayout\_5")

self.radioButton\_8 = QtWidgets.QRadioButton(self.widget1)

self.radioButton\_8.setObjectName("radioButton\_8")

self.verticalLayout\_5.addWidget(self.radioButton\_8)

self.radioButton\_9 = QtWidgets.QRadioButton(self.widget1)

self.radioButton\_9.setObjectName("radioButton\_9")

self.verticalLayout\_5.addWidget(self.radioButton\_9)

self.widget2 = QtWidgets.QWidget(self.centralwidget)

self.widget2.setGeometry(QtCore.QRect(1400, 750, 20, 400))

self.widget2.setObjectName("widget2")

self.verticalLayout\_6 = QtWidgets.QVBoxLayout(self.widget2)

self.verticalLayout\_6.setContentsMargins(0, 0, 0, 0)

self.verticalLayout\_6.setObjectName("verticalLayout\_6")

self.radioButton\_10 = QtWidgets.QRadioButton(self.widget2)

self.radioButton\_10.setText("")

self.radioButton\_10.setObjectName("radioButton\_10")

self.verticalLayout\_6.addWidget(self.radioButton\_10)

self.radioButton\_11 = QtWidgets.QRadioButton(self.widget2)

self.radioButton\_11.setText("")

self.radioButton\_11.setObjectName("radioButton\_11")

self.verticalLayout\_6.addWidget(self.radioButton\_11)

self.radioButton\_12 = QtWidgets.QRadioButton(self.widget2)

self.radioButton\_12.setText("")

self.radioButton\_12.setObjectName("radioButton\_12")

self.verticalLayout\_6.addWidget(self.radioButton\_12)

mainWindow.setCentralWidget(self.centralwidget)

self.pushButton = QtWidgets.QPushButton(self.centralwidget)

self.pushButton.setGeometry(QtCore.QRect(1485, 445, 25, 25))

self.pushButton.setObjectName("pushButton")

self.retranslateUi(mainWindow)

self.pushButton.clicked.connect(self.controltrans)

QtCore.QMetaObject.connectSlotsByName(mainWindow)

def controltrans(self):

count=0

if self.radioButton.isChecked()==1:

count+=1

self.pcount=2

if self.radioButton\_2.isChecked()==1:

count+=1

self.pcount=5

if self.radioButton\_3.isChecked()==1:

count+=1

self.pcount=3

if self.radioButton\_4.isChecked()==1:

count+=1

self.pcount=4

if self.radioButton\_5.isChecked()==1:

count+=1

self.pcount=5

if self.radioButton\_6.isChecked()==1:

count+=1

self.pcount=7

if self.radioButton\_7.isChecked()==1:

count+=1

self.pcount=8

if self.radioButton\_8.isChecked()==1:

count+=1

self.mode=0

if self.radioButton\_9.isChecked()==1:

count+=1

self.mode=1

if self.radioButton\_10.isChecked()==1:

count+=1

self.size=6

if self.radioButton\_11.isChecked()==1:

count+=1

self.size=8

if self.radioButton\_12.isChecked()==1:

count+=1

self.size=12

if count==3:

signal=1

sys.exit(app.exec\_())

def getplayers(self):

return self.pcount

def getmode(self):

return self.mode

def getsize(self):

return self.size

def getsignal(self):

return self.signal

def retranslateUi(self, mainWindow):

\_translate = QtCore.QCoreApplication.translate

mainWindow.setWindowTitle(\_translate("mainWindow", "Chain Reaction"))

self.label\_2.setText(\_translate("mainWindow",

"<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"

"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"

"p, li { white-space: pre-wrap; }\n"

"</style></head><body style=\" font-family:\'Segoe Print\'; font-size:48pt; font-weight:400; font-style:normal;\">\n"

"<p align=\"center\" style=\" margin-top:12px; margin-bottom:12px; margin-left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-size:36pt; font-weight:600; color:#ffffff;\">Chain Reaction V 4.1</span></p></body></html>"))

self.label\_3.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:20pt; color:#ffffff;\">No. Of Players:</span></p></body></html>"))

self.label\_4.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">2 Players</span></p></body></html>"))

self.label\_5.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">3 Players</span></p></body></html>"))

self.label\_6.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">4 Players</span></p></body></html>"))

self.label\_7.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">5 Players</span></p></body></html>"))

self.label\_8.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">6 Players</span></p></body></html>"))

self.label\_9.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">7 Players</span></p></body></html>"))

self.label\_10.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">8 Players</span></p></body></html>"))

self.label\_11.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:24pt; color:#ffffff;\">Game Mode:</span></p></body></html>"))

self.label\_12.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; font-weight:400; color:#ffffff;\">Rebirth</span></p></body></html>"))

self.label\_13.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:16pt; color:#ffffff;\">Battle Royale</span></p></body></html>"))

self.label\_14.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:22pt; color:#ffffff;\">Grid Size:</span></p></body></html>"))

self.label\_15.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:14pt; color:#ffffff;\">6 x 6</span></p><p><span style=\" font-size:14pt; color:#ffffff; vertical-align:super;\">(Suitable for 2-3 Players)</span></p></body></html>"))

self.label\_16.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:12pt; color:#ffffff;\">8 X 8</span></p><p><span style=\" font-size:12pt; color:#ffffff;\">(Suitable for 4-6 Players)</span></p></body></html>"))

self.label\_17.setText(\_translate("mainWindow",

"<html><head/><body><p><span style=\" font-size:12pt; color:#ffffff;\">12 x 12</span></p><p><span style=\" font-size:12pt; color:#ffffff;\">(Suitable for 7-8 Players)</span></p></body></html>"))

self.label\_18.setText(\_translate("mainWindow",

"<html><head/><body><p align=\"center\"><span style=\" font-size:20pt; font-weight:400; color:#ffffff; vertical-align:super;\">BY: Girish Salunke - 110 Anjali Save - 111 Bhumika Punjabi- 96 SEIT- 2 (S22)</span></p></body></html>"))

self.label\_20.setText(\_translate("mainWindow",

"<html><head/><body><p align=\"center\"><span style=\" font-size:20pt; color:#ffffff;\"> Click 'S' to Play</span></p></body></html>"))

self.pushButton.setText(\_translate("mainWindow", "S"))

self.radioButton.setToolTip(\_translate("mainWindow", "<html><head/><body><p>2 players</p></body></html>"))

self.radioButton\_2.setText(\_translate("mainWindow", "RadioButton"))

self.radioButton\_5.setText(\_translate("mainWindow", "RadioButton"))

self.radioButton\_8.setText(\_translate("mainWindow", "RadioButton"))

self.radioButton\_9.setText(\_translate("mainWindow", "RadioButton"))

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

import sys

app = QtWidgets.QApplication(sys.argv)

mainWindow = QtWidgets.QMainWindow()

ui = Ui\_mainWindow()

ui.setupUi(mainWindow)

mainWindow.show()

sys.exit(app.exec\_()

**Screenshots:**



