# A Project Report on

**HANGMAN GAME**

Submitted by

**Pranali Waghmare(18IT1011)**

**Purva Tol(18IT2017)**

**Ritvika Sanap(18IT1040)**

Under the Guidance of

**Mr. Madhav Vyas**

**Department of Information Technology**

**Ramrao Adik Institute of Technology**

**Nerul ,Navi Mumbai**

**Certificate**

This is to certify that the project entitled **Hangman Game** is being submitted to the Department of Information Technology, Ramrao Adik Institute of Technology, Navi Mumbai.

Project Guide External

Examiner(Mr. Madhav Vyas) ( )

# Acknowledgement

We owe our gratitude to many people who have supported us throughout this journey.

We would, firstly like to express our heartfelt gratitude towards our respected Principal Dr. M.D. Patil and our Head of Department Dr. Ashish Jadhav for providing us immense facilities, guidance and never-ending support.

The completion of any inter-disciplinary project depends upon cooperation and combined efforts of several sources of knowledge. We take this opportunity to express our profound gratitude and deep regards to our guide Mr. Madhav Vyas for his exemplary guidance, monitoring and constant encouragement throughout the course of this project.

Lastly, we thank our parents, family, friends and well-wishers who always looked for the chance to help us in whatever means came forth and for their constant encouragement without which the project would not be a distant reality.

Contents

1. AIM 6
2. Introduction 7
3. Abstract 8
4. Working of the system 9
5. CODE 10
6. Result 22
7. Conclusion 25

**AIM**

The purpose of this project is to have some fun with movie names. The main part of the game is that it does not require any internet access. Therefore, the user can play the game anywhere, anytime without internet access. As the number of chances to guess a movie are limited for a user, playing the game becomes more fun and at the same time challenging.

**INTRODUCTION**

Hangman is a game in which you have to guess a movie name within a limited number of chances. A proper set of blank blocks will be provided which indirectly indicates the length of the words in the movie. It is obvious that if user uses all his chances and still could not guess the movie, the game ends. (Hanged!)

On the other hand, if the user tends to guess the correct words of the movie within those limited number of chances, the user wins!

The total number of chances provided to a user is nine.

**ABSTRACT**

In the following game, if the user enters a correct letter for a given movie, the screen displays ‘CORRECT’.

Else if, the user enters a wrong letter, the screen displays ‘WRONG’. Here if the user loses all his nine chances, the movie name will be displayed in which, the red letters indicates those letters which the user couldn’t guess. On each wrong hit, the program will display the hanged stick figure in parts.

Also, if the user enters a letter which is already entered, the screen displays ‘ALREADY ENTERED’.

**WORKING OF THE SYSTEM**

The code mainly uses Python Turtle Graphics.

“Turtle” is a python feature like a drawing board, which lets you command a turtle to draw all over it!

We can use functions

like turtle.forward(...) and turtle.left(...) which can move the turtle around.

Before we use turtle, we have to import it.

**import turtle**

**CODE**

import random

import turtle

import sys

import time

#----------------------------global variables-------------------------------------#

flagW=False

flagL=False

flafA=False

flagWr=False

gameFlag=True

wrongCount=0

hmTC=(0,0)

hm=turtle.Turtle()

hm.speed(0)

state=turtle.Turtle()

state.ht()

#---------------------------turtle library related functions---------------------------------------------------#

def correct():

state.clear()

state.color("green")

state.write("Correct!!",font=("Arial",50,"bold"))

def lose():

state.clear()

state.color("green")

state.write("YOU LOSE!!", font=("Arial",50,"bold"))

cordxy=letter[0].pos()

temp=turtle.Turtle()

temp.ht()

temp.speed(0)

temp.color("red")

temp.penup()

temp.setpos(cordxy[0],cordxy[1]+40)

temp.pendown()

temp.write("This was the movie:",align="left", font=("Arial", 30, "bold"))

for i in range(len(movie)):

if(movie[i] not in user\_list\_correct):

letter[i].color("red")

letter[i].write(movie[i],align="left", font=("Arial", 20, "bold"))

def win():

state.clear()

state.color("green")

state.write("YOU WIN!!", font=("Arial",50,"bold"))

def already():

state.clear()

state.color("purple")

state.write("You already pressed that key!!", align="center",font=("Arial",50,"bold"))

def wrong():

state.clear()

state.color("red")

state.write("wrong", font=("Arial",50,"bold"))

def A():

evaluate(movie,"A")

def B():

evaluate(movie,"B")

def C():

evaluate(movie,"C")

def D():

evaluate(movie,"D")

def E():

evaluate(movie,"E")

def F():

evaluate(movie,"F")

def G():

evaluate(movie,"G")

def H():

evaluate(movie,"H")

def I():

evaluate(movie,"I")

def J():

evaluate(movie,"J")

def K():

evaluate(movie,"K")

def L():

evaluate(movie,"L")

def M():

evaluate(movie,"M")

def N():

evaluate(movie,"N")

def O():

evaluate(movie,"O")

def P():

evaluate(movie,"P")

def Q():

evaluate(movie,"Q")

def R():

evaluate(movie,"R")

def S():

evaluate(movie,"S")

def T():

evaluate(movie,"T")

def U():

evaluate(movie,"U")

def V():

evaluate(movie,"V")

def W():

evaluate(movie,"W")

def X():

evaluate(movie,"X")

def Y():

evaluate(movie,"Y")

def Z():

evaluate(movie,"Z")

#function which controls the drawing of hangman figure.

def Hangman(num):

global hmTC

if(num==0):

hm.penup()

hm.setpos(-670,-340)

hm.pendown()

hm.ht()

hm.begin\_fill()

hm.fd(400)

hm.lt(90)

hm.fd(50)

hm.lt(90)

hm.fd(390)

hm.rt(90)

hm.fd(600)

hm.rt(90)

hm.fd(400)

hm.rt(90)

hm.fd(20)

hm.lt(90)

hm.fd(10)

# print(hmTC[0])

hm.lt(90)

hmTC=hm.pos()

hm.fd(30)

hm.lt(90)

hm.fd(420)

hm.lt(90)

hm.fd(550)

hm.end\_fill()

#hm.end\_fill()

hm.pensize(5)

elif(num==1):

hm.penup()

hm.setpos(hmTC[0],hmTC[1])

hm.rt(90)

hm.pendown()

hm.circle(40)

#print(hm.pos())

elif(num==2):

hm.penup()

hm.setpos(hmTC[0],hmTC[1]-80)

hm.lt(90)

hm.pendown()

hm.fd(140)

hmTC=hm.pos()

elif(num==3):

hm.penup()

hm.rt(30)

hm.pendown()

hm.fd(100)

hm.rt(60)

hm.fd(10)

hm.penup()

hm.rt(180)

hm.fd(10)

hm.rt(30)

hm.fd(100)

hm.rt(90)

hm.pendown()

elif(num==4):

hm.penup()

hm.setpos(hmTC[0],hmTC[1])

hm.pendown()

hm.lt(60)

hm.fd(100)

hm.lt(60)

hm.fd(10)

#hm.rt(90)

elif(num==5):

hm.penup()

hm.setpos(hmTC[0],hmTC[1])

hm.lt(90)

hm.fd(90)

hm.lt(180)

hm.rt(30)

hm.pendown()

hm.fd(80)

hm.rt(60)

elif(num==6):

hm.penup()

hm.setpos(hmTC[0],hmTC[1])

hm.rt(90)

hm.fd(90)

hm.lt(180)

hm.lt(30)

hm.pendown()

hm.fd(80)

elif(num==7):

hm.penup()

hm.setpos(-235.00,265.00)

hm.rt(90)

hm.pendown()

hm.circle(4)

elif(num==8):

hm.penup()

hm.setpos(-265,265.00)

hm.rt(90)

hm.pendown()

hm.circle(4)

else:

hm.penup()

hm.setpos(hm.pos()[0]+25,hm.pos()[1]-35)

hm.pendown()

hm.rt(30)

hm.circle(10,180)

#creates boxes to type characters in

def createDash(num):

dash=[]

letter=[]

for i in range(num):

#print(movie\_space\_position)

dash.append(turtle.Turtle())

dash[i].color("#800000","white")

dash[i].pensize(4)

letter.append(turtle.Turtle())

dash[i].ht()

dash[i].speed(0)

letter[i].ht()

dash[i].penup()

letter[i].penup()

dash[i].setpos(-600+i\*50,-200)

letter[i].setpos(-595+i\*50,-200)

dash[i].pendown()

letter[i].pendown()

if(i in movie\_space\_position):

dash[i].penup()

continue

dash[i].begin\_fill()

dash[i].fd(30)

dash[i].lt(90)

dash[i].fd(30)

dash[i].lt(90)

dash[i].fd(30)

dash[i].lt(90)

dash[i].fd(30)

dash[i].end\_fill()

return letter

#----------------------------core working of the game------------------------------#

def evaluate(movie,user\_input):

global wrongCount

global gameFlag

if(gameFlag==True):

if(user\_input in movie and user\_input not in user\_list\_correct):

tempPositionList=[]

for i in range(len(movie)):

if(movie[i]==user\_input):

letter[i].write(user\_input,align="left", font=("Arial", 20, "bold"))

user\_list\_correct.append(user\_input)

tempPositionList.append(i)

correct()

if(len(user\_list\_correct)==len(movie)):

win()

gameFlag=False

#return tempPositionList

else:

if(user\_input in user\_list\_correct):

already()

else:

wrong()

user\_list\_wrong.append(user\_input)

wrongCount+=1

if(wrongCount>=9):

lose()

gameFlag=False

Hangman(wrongCount)

f = open("movies.txt")

movie\_list=[]

for line in f:

line = line.strip()

movie\_list.append(line)

movie = random.choice(movie\_list)

movie=movie.upper()

l = len(movie)

movie = list(movie)

movie\_space\_count=0

movie\_space\_position=[]

user\_list\_correct = []

user\_list\_wrong = []

user\_input=[]

for c in range(len(movie)):

if(movie[c]==' '):

movie\_space\_count+=1

movie\_space\_position.append(c)

user\_list\_correct.append(' ')

wn = turtle.Screen()

wn.setup (width=1920, height=1080, startx=0, starty=0)

wn.bgpic("imain.gif")

wn.tracer(2)

Hangman(0)

letter = createDash(len(movie))

wn.tracer(1)

wn.onkey(A,"a")

wn.onkey(B,"b")

wn.onkey(A,"a")

wn.onkey(A,"A")

wn.onkey(B,"b")

wn.onkey(B,"B")

wn.onkey(C,"c")

wn.onkey(C,"C")

wn.onkey(D,"d")

wn.onkey(D,"D")

wn.onkey(E,"e")

wn.onkey(E,"E")

wn.onkey(F,"f")

wn.onkey(F,"F")

wn.onkey(G,"g")

wn.onkey(G,"G")

wn.onkey(H,"h")

wn.onkey(H,"H")

wn.onkey(I,"i")

wn.onkey(I,"I")

wn.onkey(J,"j")

wn.onkey(J,"J")

wn.onkey(K,"k")

wn.onkey(K,"K")

wn.onkey(L,"l")

wn.onkey(L,"L")

wn.onkey(M,"m")

wn.onkey(M,"M")

wn.onkey(N,"n")

wn.onkey(N,"N")

wn.onkey(O,"o")

wn.onkey(O,"O")

wn.onkey(P,"p")

wn.onkey(P,"P")

wn.onkey(Q,"q")

wn.onkey(Q,"Q")

wn.onkey(R,"r")

wn.onkey(R,"R")

wn.onkey(S,"s")

wn.onkey(S,"S")

wn.onkey(T,"t")

wn.onkey(T,"T")

wn.onkey(U,"u")

wn.onkey(U,"U")

wn.onkey(V,"v")

wn.onkey(V,"V")

wn.onkey(W,"w")

wn.onkey(W,"W")

wn.onkey(X,"x")

wn.onkey(X,"X")

wn.onkey(Y,"y")

wn.onkey(Y,"Y")

wn.onkey(Z,"z")

wn.onkey(Z,"Z")

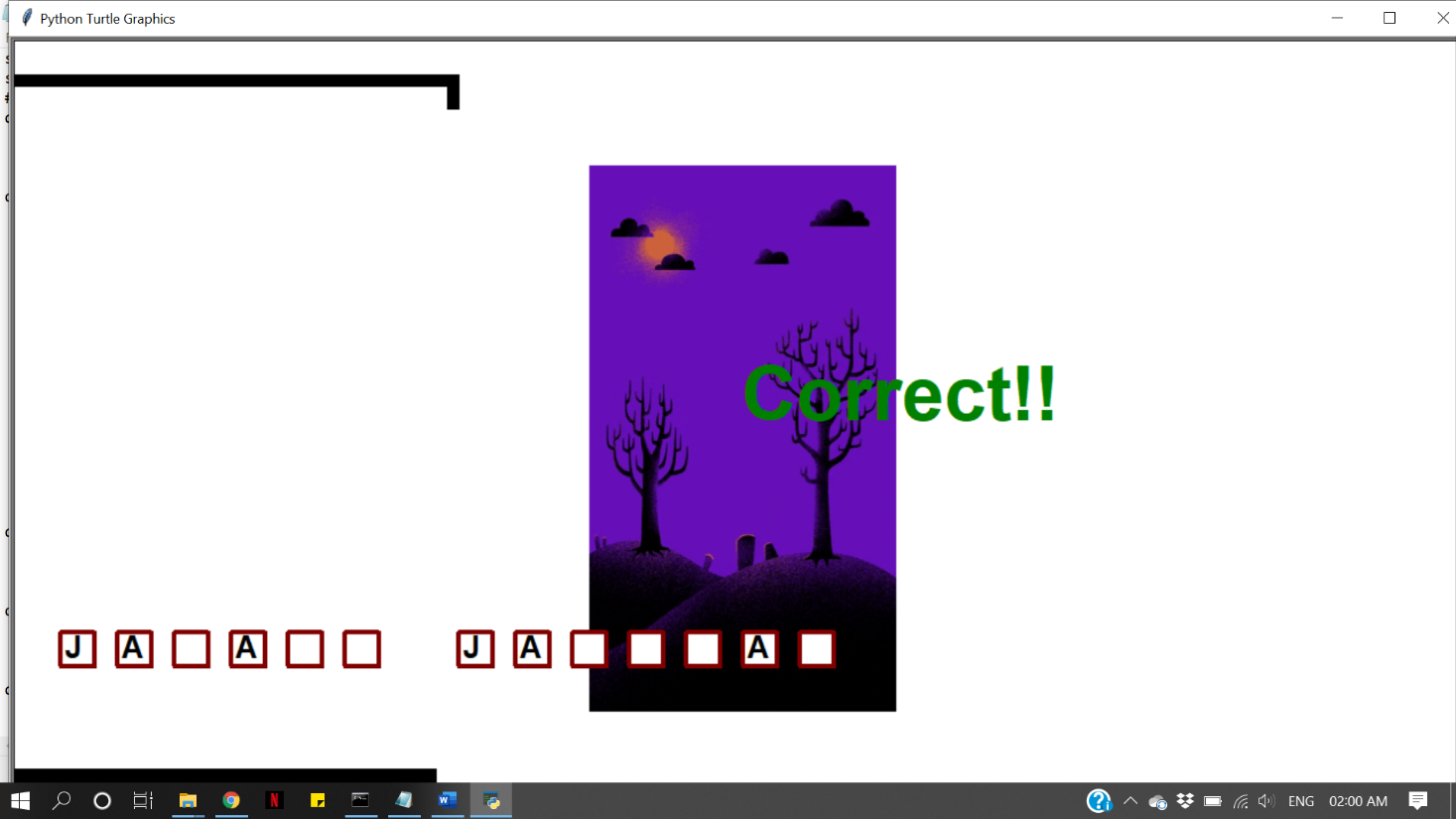
if(gameFlag==True):

wn.listen()

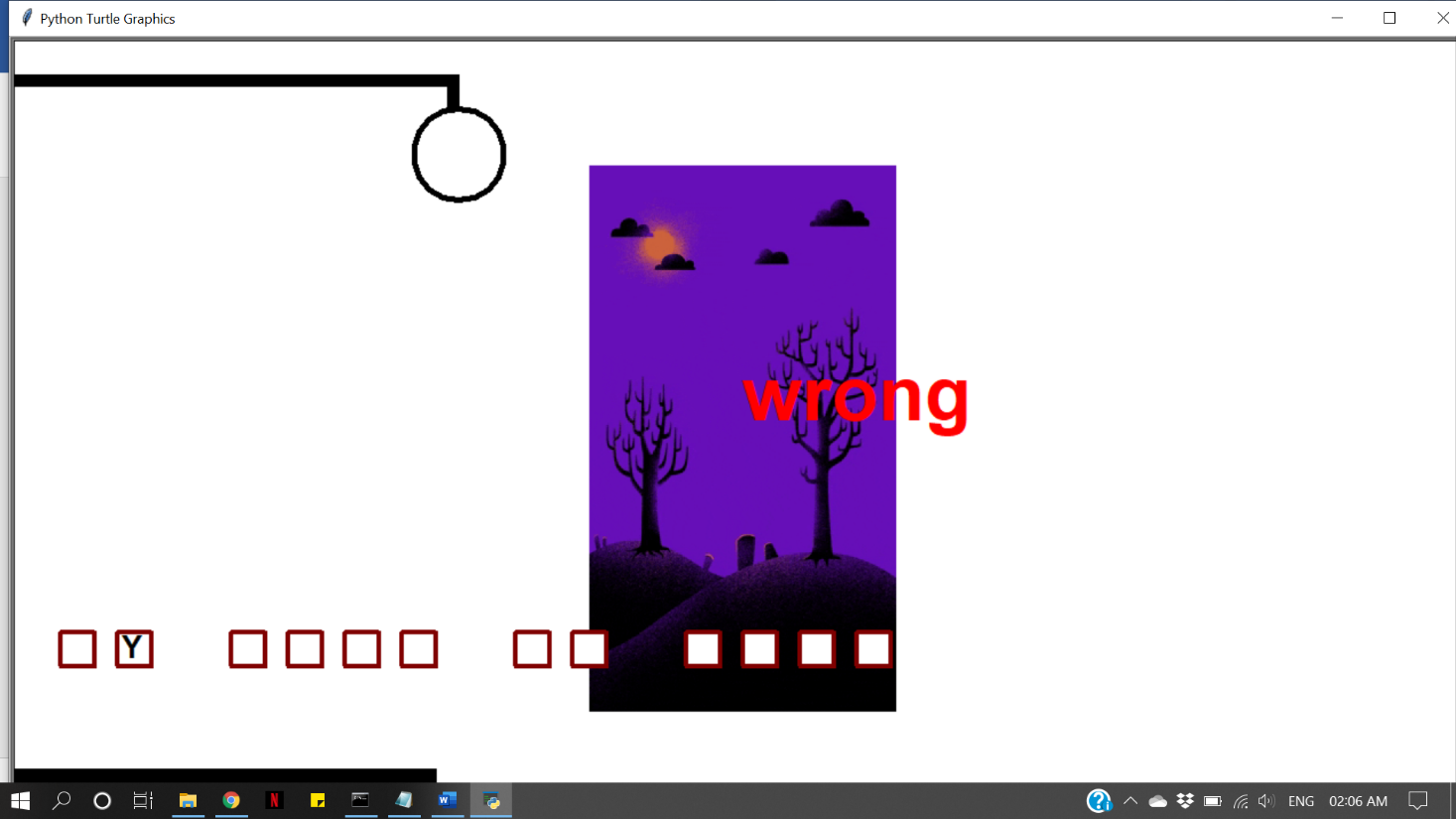
turtle.mainloop()

**RESULT**

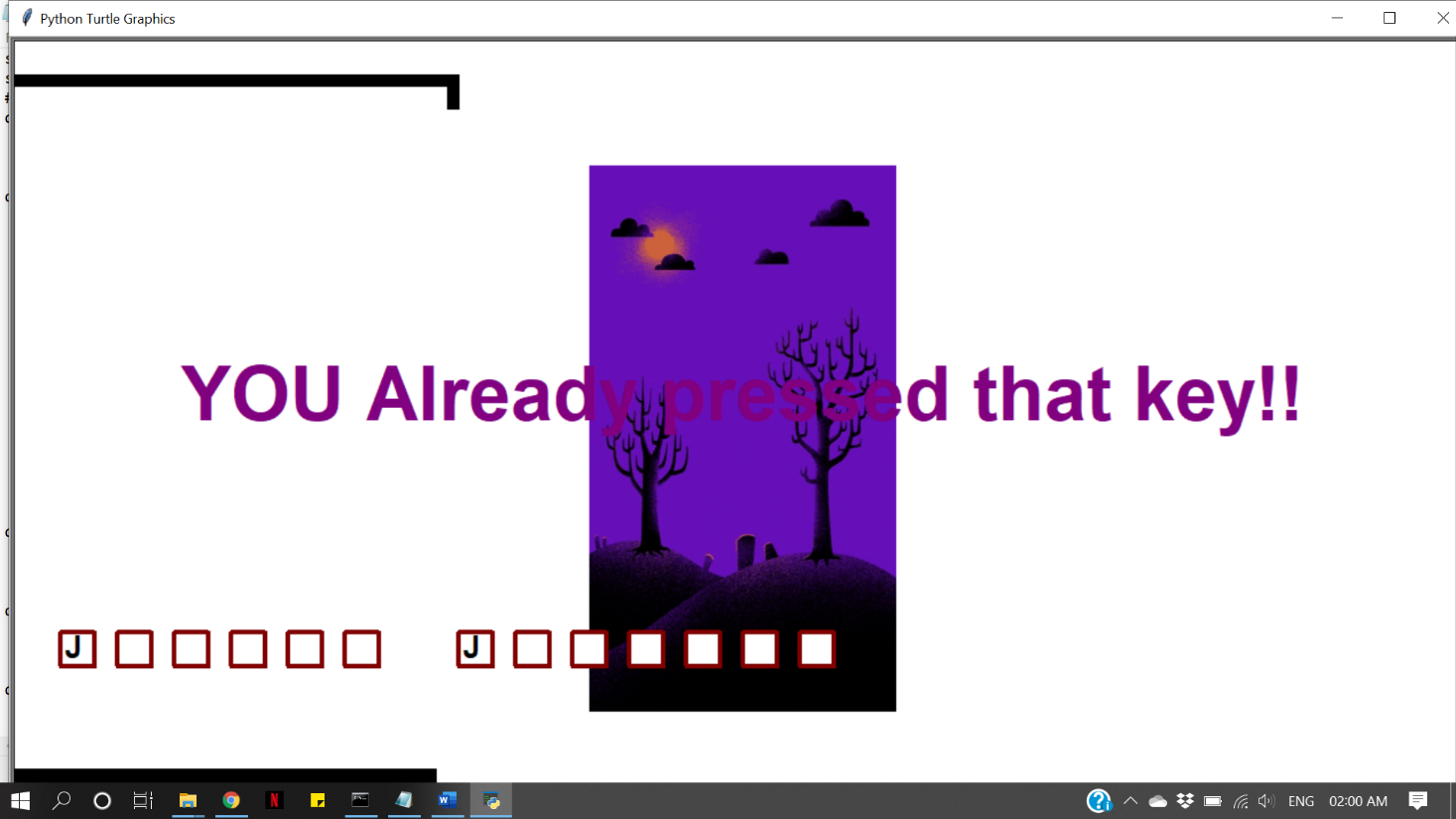
# On Correct letter entered.

****

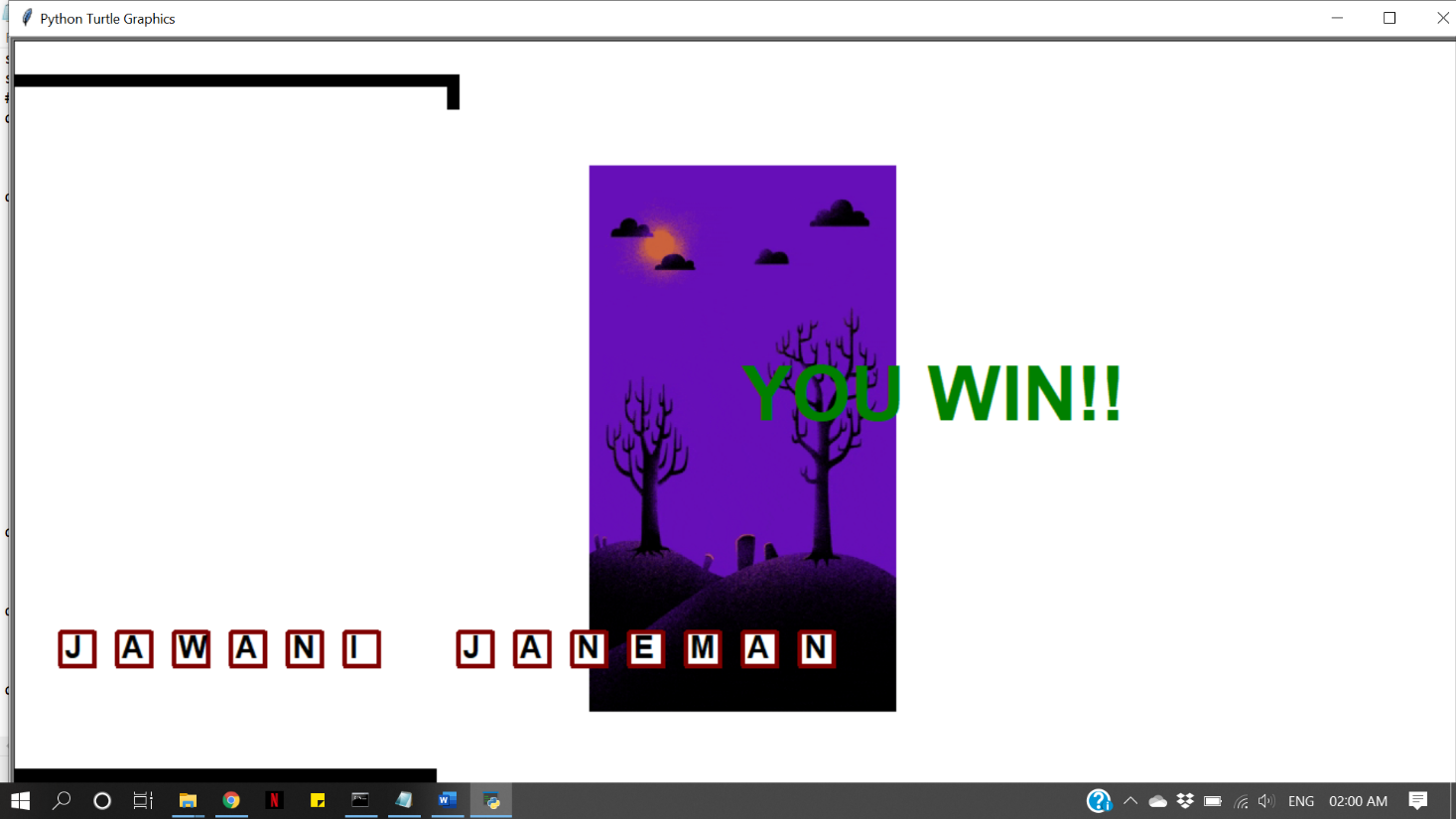
1. **On wrong letter entered.**

****

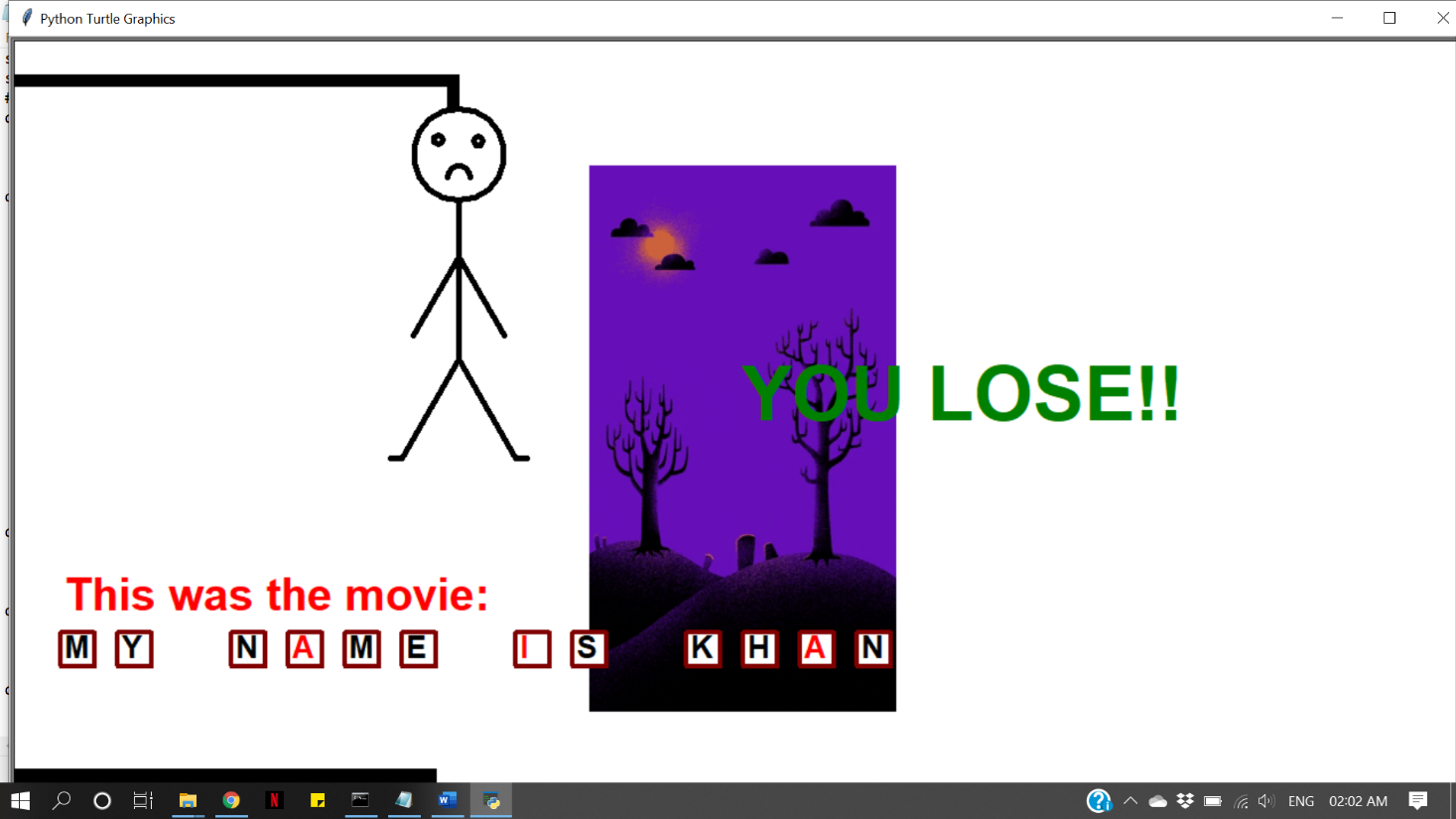
1. **On repeated letter entered.**

****

1. **On winning the game.**

****

1. **HANGED!**

****

**CONCLUSION**

Hence, we conclude Hangman on Python is fun to play with some challenging skills. Even without internet, this game is way better and interesting with Python as platform.

# References

* Google
* Wikipedia
* stackoverflow.com
* Youtube