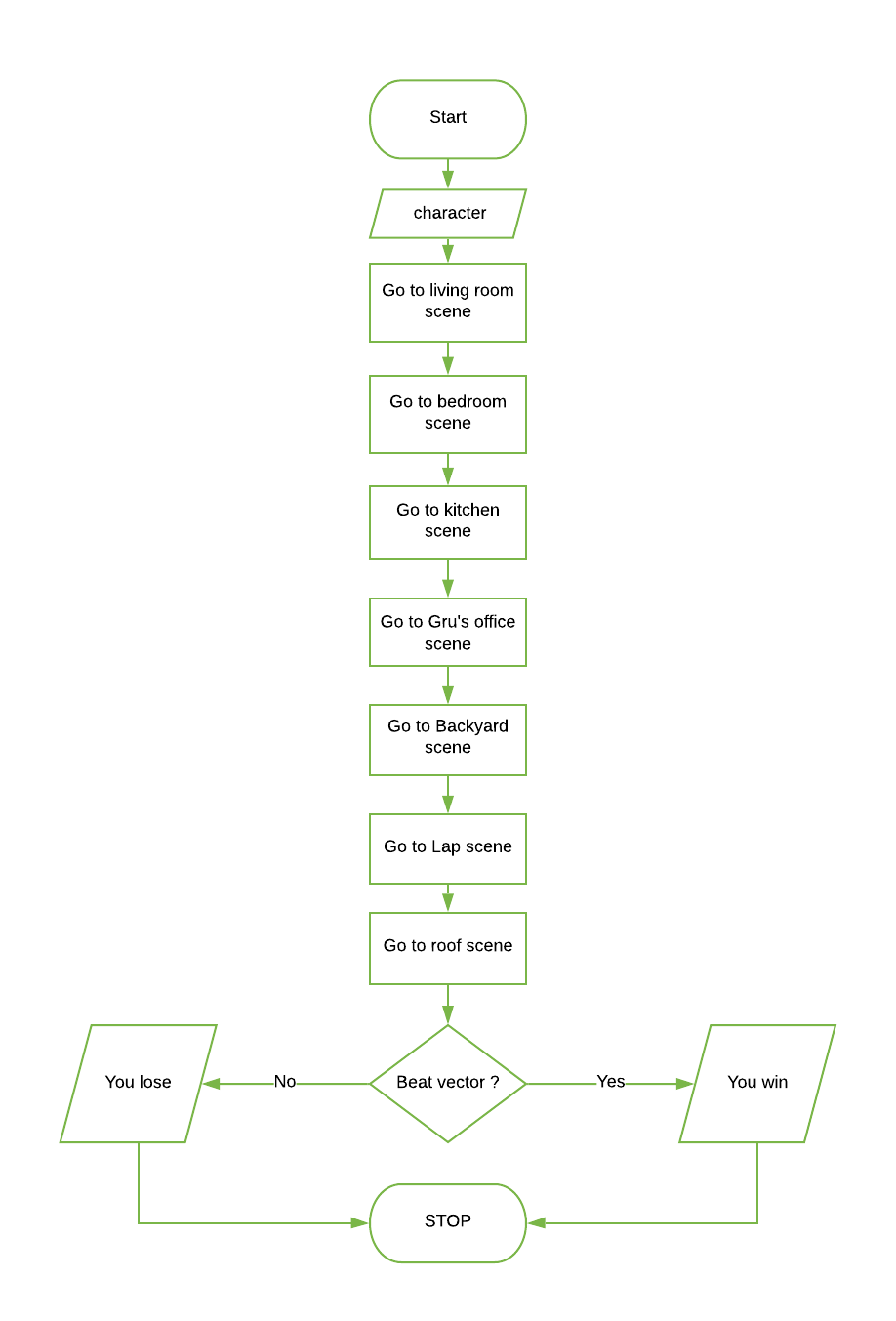
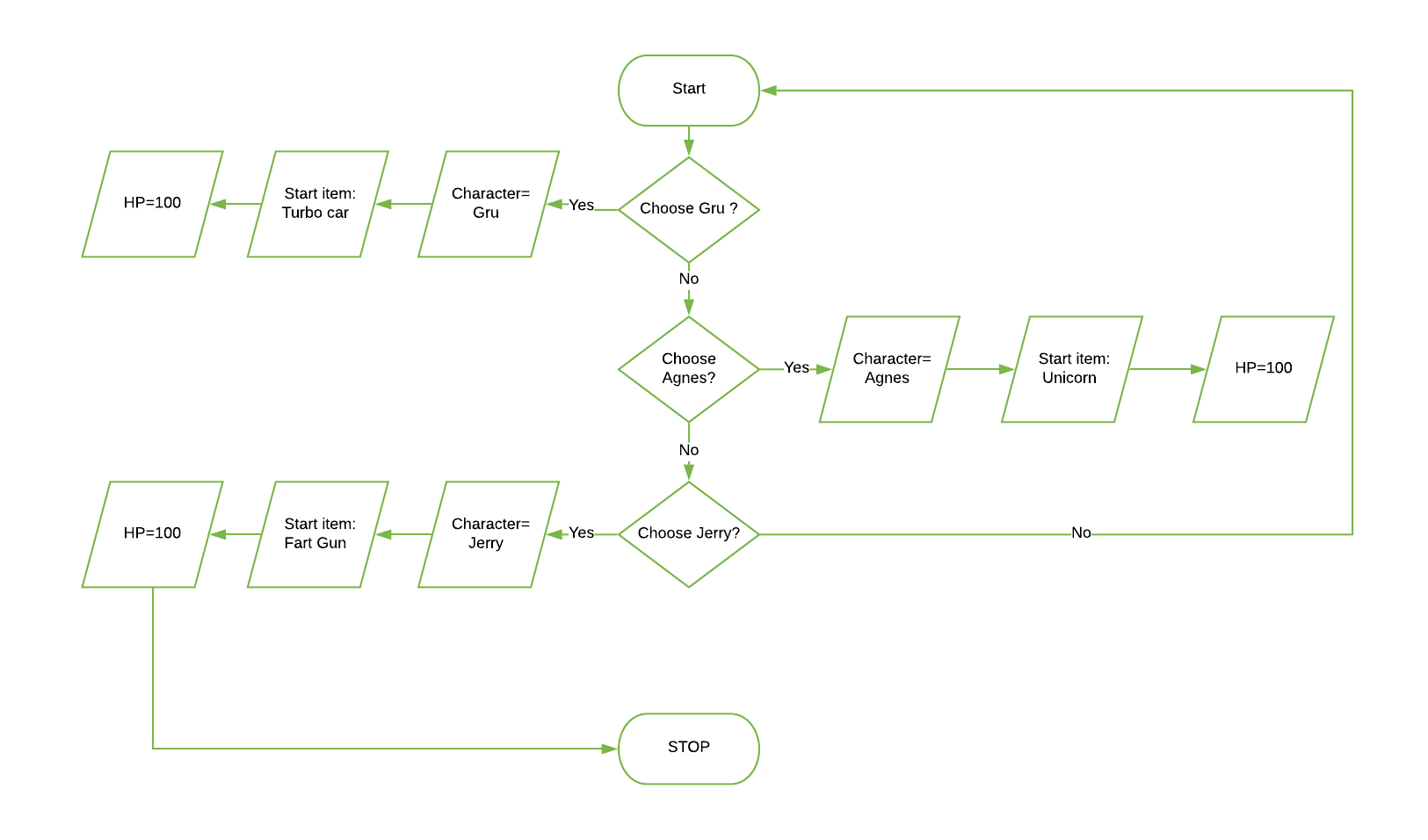
Program 3 : Fantasy Adventure Game Report

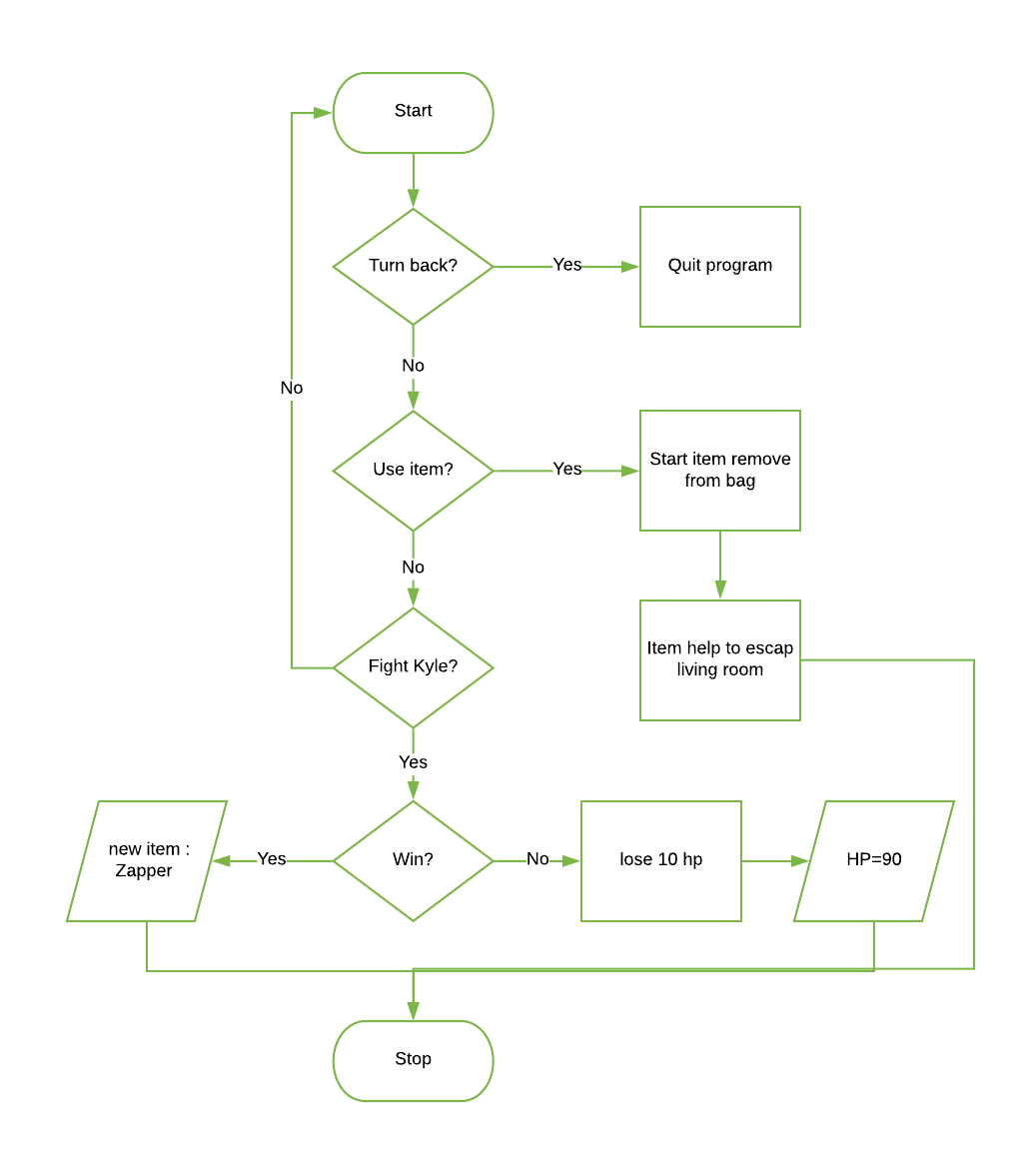
# Fantasy adventure game program flow chart



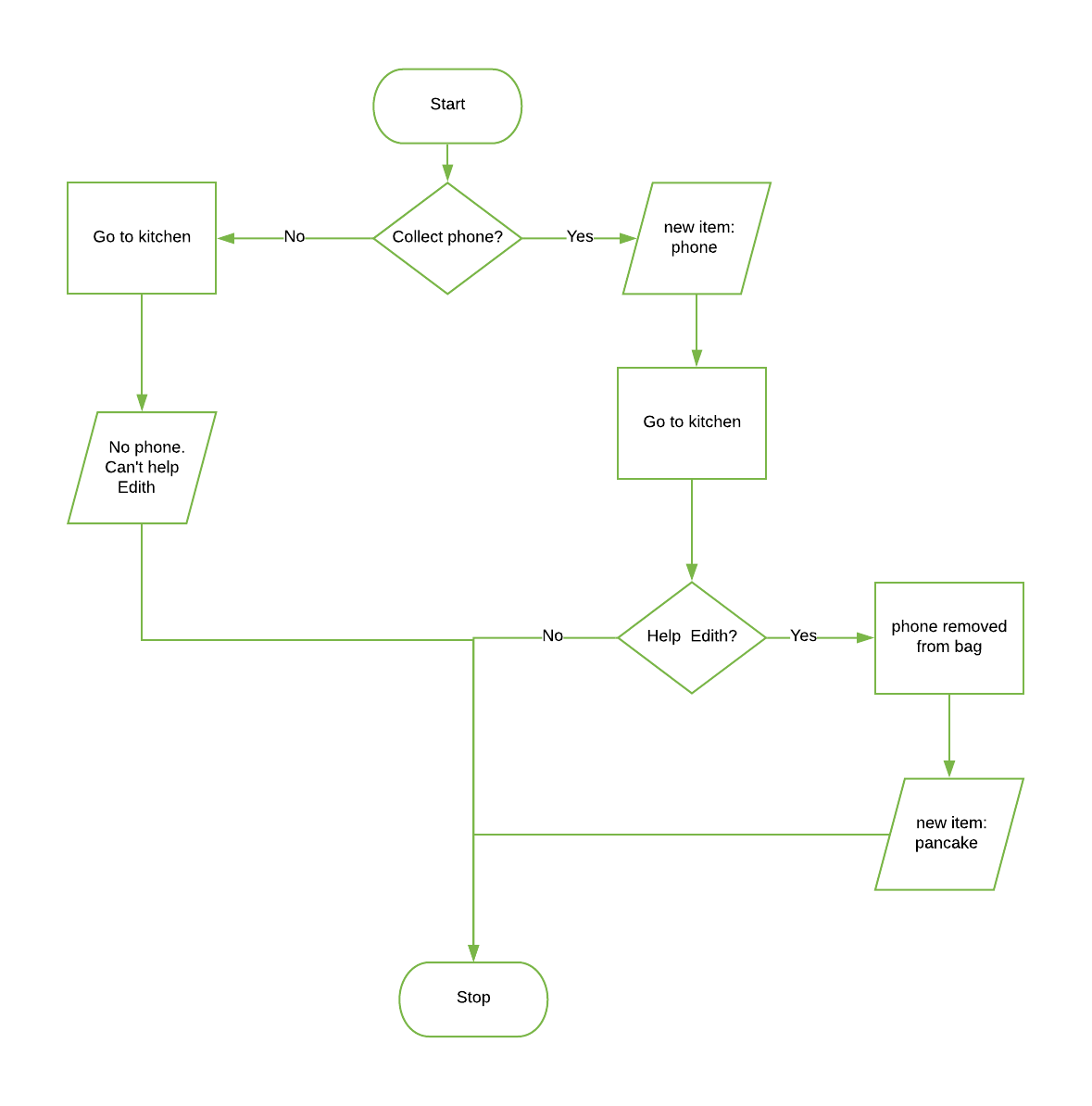
## Subprocess: Choose character



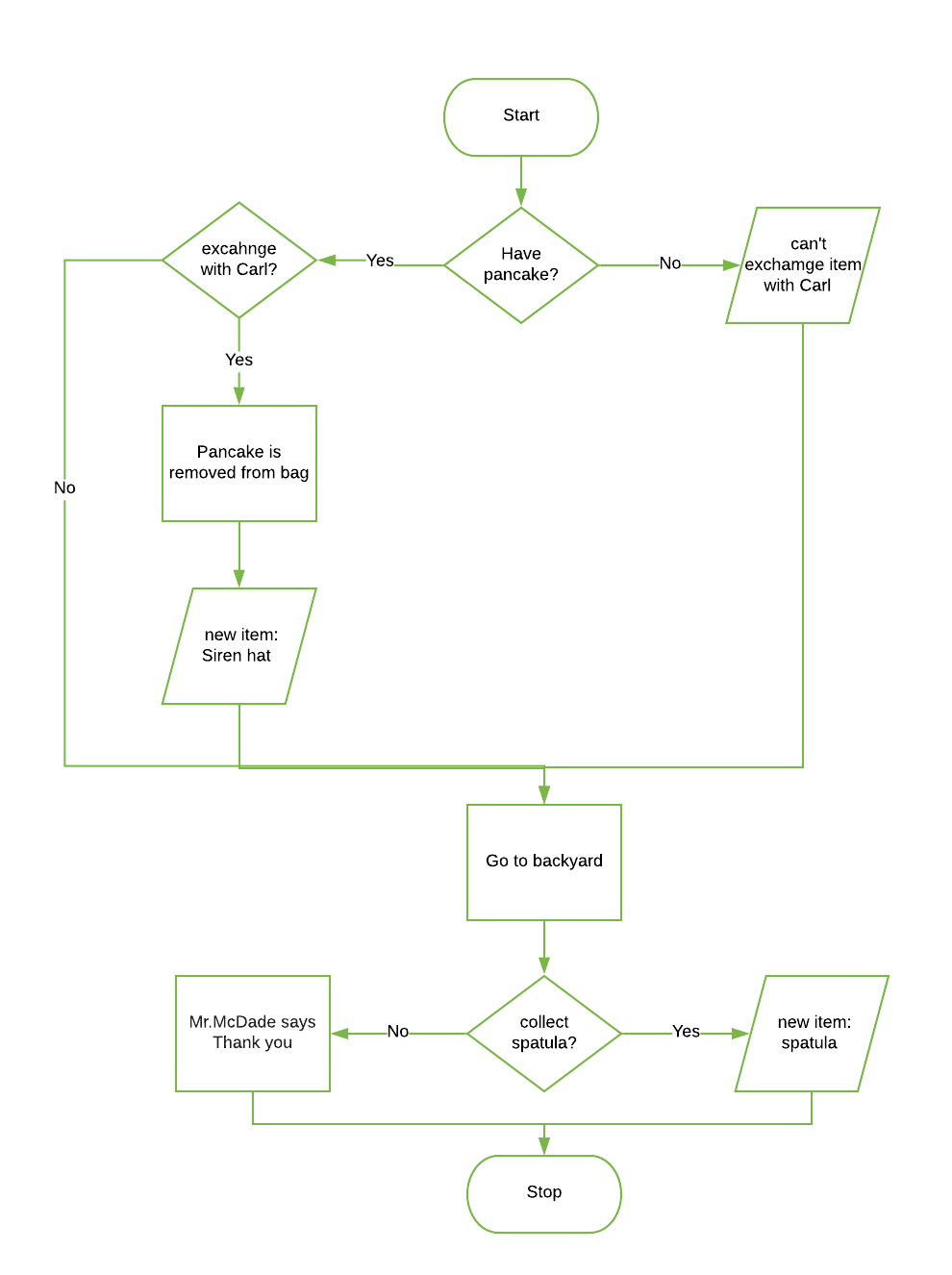
## Sunprocess: Living room scene



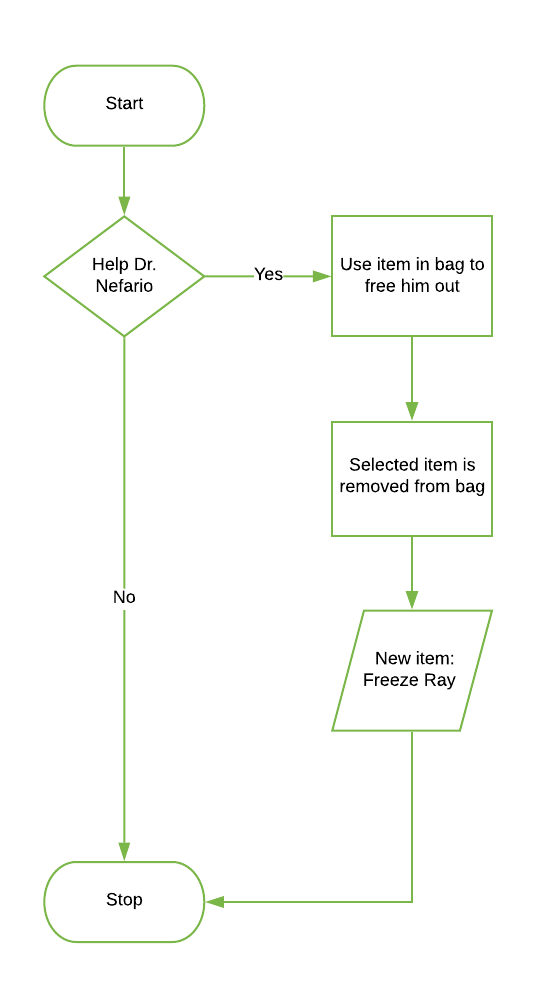
## Subprocess: Bedroom and Kitchen



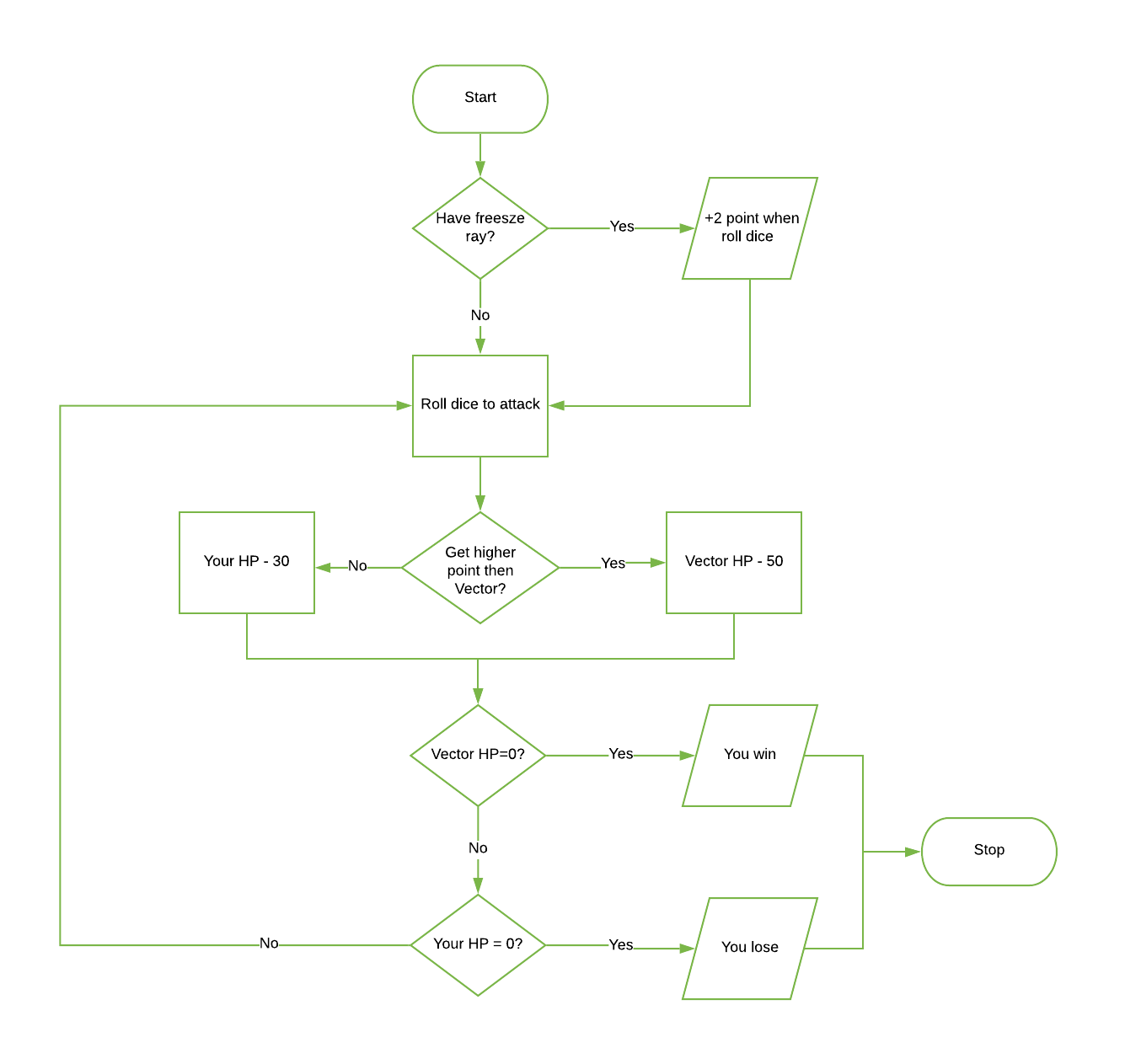
## Subprocess: Gru’s office and backyard



## Subprocess: Lap scene



## Sunprocesse: Roof scene

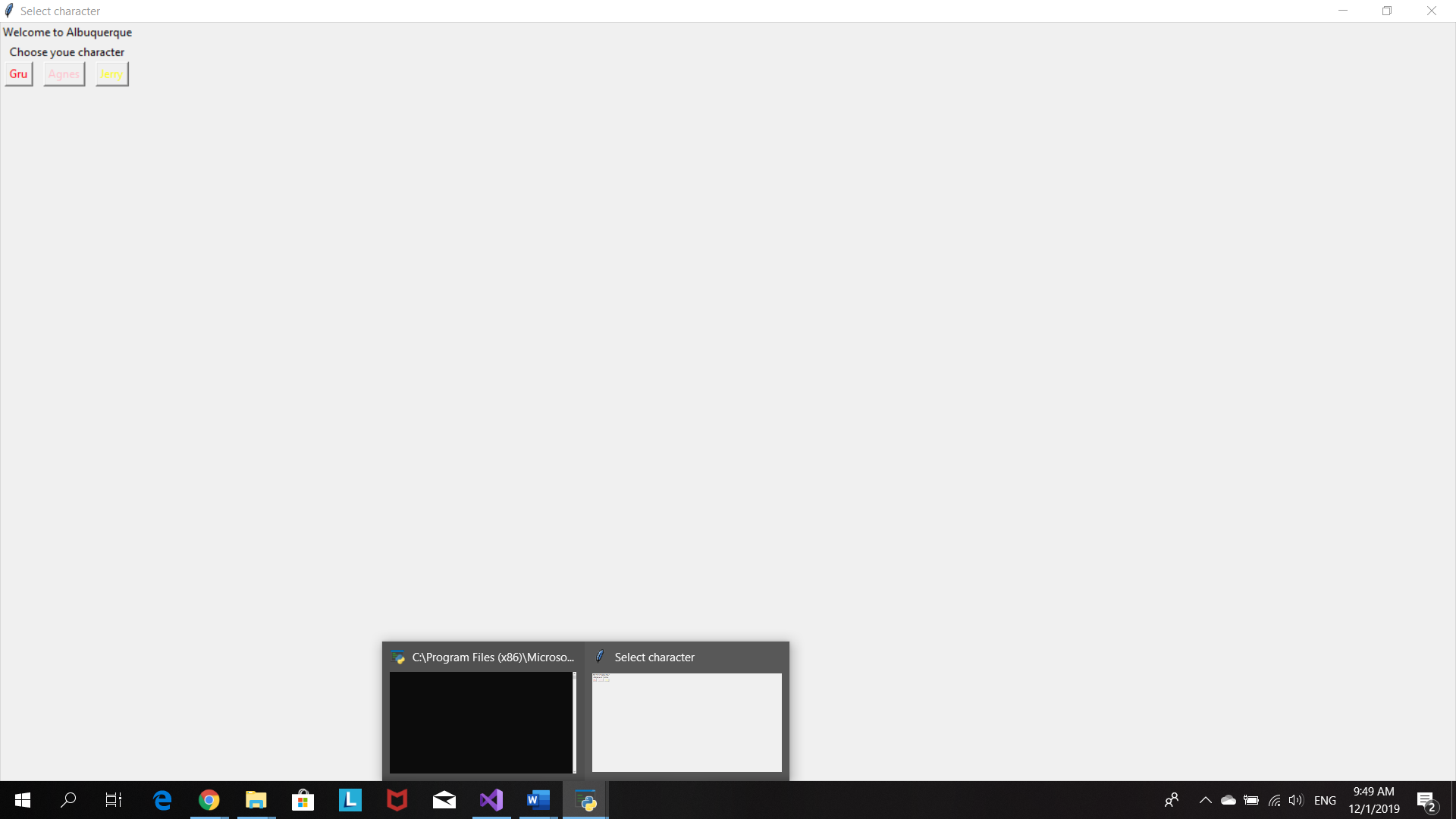


# Testing Table- Program 3: Fantasy adventure game

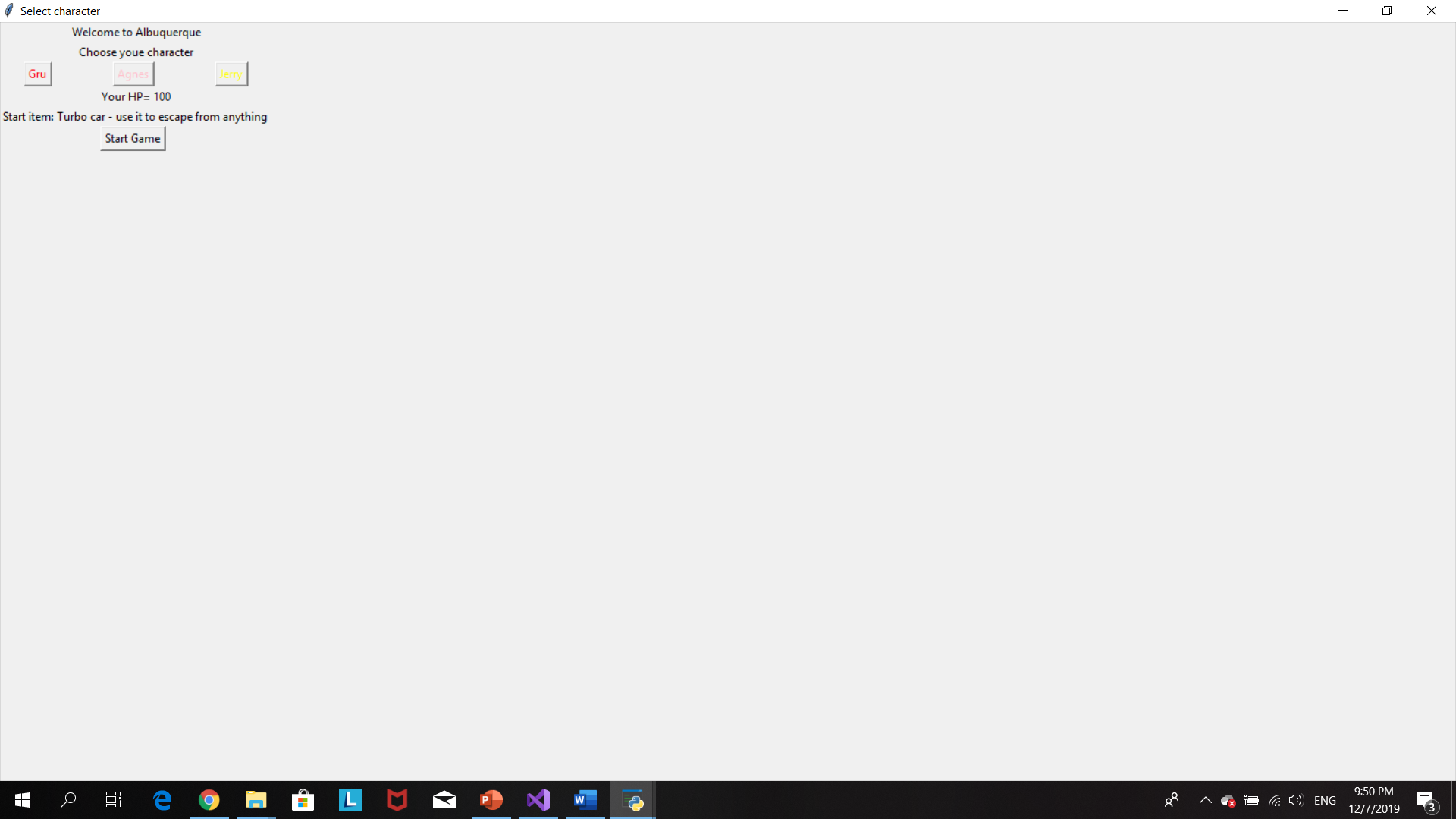
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No. | Item to test | Test description | Test action | Expected result | Actual result | Comment |
| 1 | Button(root,text="Gru",fg="red",command=Gruitem) | Valid data | Click Button | Text and button pop up | Text display and start button pop up | - |
| 2 | Button(root,text="Gru",fg="red",command=Gruitem) | Valid command function | command=Gruitem | Text and button pop up | Text display and start button pop up | - |
| 3 | Button(root,text="Gru",fg="red",command=Gruitem) | Function is renamed | def whatwhat(): | Text and button pop up | Program crashed-command function do not define | Build back-up command function |
| 4 | Button(root,text="Gru",fg="red",command=Gruitem) | Function doesn’t not exit | Delete function Gruitem | Text and button pop up | Program crashed- function do not define | Build back-up command function |
| 5 | Button(root,text="Agnes",fg="pink",command=Agnesitem) | Valid data | Click Button | Text and button pop up | Text display and start button pop up | - |
| 6 | Button(root,text="Agnes",fg="pink",command=Agnesitem) | Valid command function | command= Agnesitem | Text and button pop up | Text display and start button pop up | - |
| 7 | Button(root,text="Agnes",fg="pink",command=Agnesitem) | Function is renamed | def whatwhat(): | Text and button pop up | Program crashed-command function do not define | Build back-up command function |
| 8 | Button(root,text="Agnes",fg="pink",command=Agnesitem) | Function doesn’t not exit | Delete function Agnesitem | Text and button pop up | Program crashed- function do not define | Build back-up command function |
| 9 | Button(root,text="Jerry",fg="yellow",command=Jerryitem) | Valid data | Click Button | Text and button pop up | Text display and start button pop up | - |
| 10 | Button(root,text="Jerry",fg="yellow",command=Jerryitem) | Valid command function | command= Jerryitem | Text and button pop up | Text display and start button pop up | - |
| 11 | Button(root,text="Jerry",fg="yellow",command=Jerryitem) | Function is renamed | def whatwhat(): | Text and button pop up | Program crashed-command function do not define | Build back-up command function |
| 12 | Button(root,text="Jerry",fg="yellow",command=Jerryitem) | Function doesn’t not exit | Delete function Jerryitem | Text and button pop up | Program crashed- function do not define | Build back-up command function |
| 13 | Button(root,text="Start Game",command=HouseGru) | Valid data | Click Button | New window pop up | New window display – program follow the command | - |
| 14 | Button(root,text="Start Game",command=HouseGru) | Valid command function | command=HouseGru | New window pop up | New window display – program follow the command | - |
| 15 | Button(root,text="Start Game",command=HouseGru) | Function is renamed | def whatwhat(): | New window pop up | Program crashed-command function do not define | Build back-up command function |
| 16 | Button(root,text="Start Game",command=HouseGru) | Function doesn’t not exit | Delete function HouseGru | New window pop up | Program crashed- function do not define | Build back-up command function |
| 17 | Button(house,text="Turn back",command=quit) | Valid data | Click Button | Quit program | Program crash | Fix the command ‘quit’ |
| 18 | Button(house,text="Turn back",command=quit) | Valid command | command= quit | Quit program | Program crash | Fix the command ‘quit’ |
| 19 | Button(house,text="Turn back",command=quit) | Invalid command | Command=QuIt | Quit program | Program crash | Fix the command ‘quit’ |
| 20 | Button(house,text="fight",command=(fight)) | Valid data | Click Button | Labels and button pop up | Labels and button pop up – program continue as normal | - |
| 21 | Button(house,text="fight",command=(fight)) | Valid command function | Command= fight | Labels and button pop up | Labels and button pop up – program continue as normal | - |
| 22 | Button(house,text="fight",command=(fight)) | Function is renamed | def whatwhat(): | Labels and button pop up | Program crashed- function do not define | Double check spelling |
| 23 | Button(house,text="fight",command=(fight)) | Function doesn’t not exit | Delete function fight | Labels and button pop up | Program crashed- function do not define | Double check command |
| 24 | Button(house,text="Use item",command=houseitemGru) | Valid data | Click Button | Program follow command | start item is removed from the item bag and labels pop up | - |
| 25 | Button(house,text="Use item",command=houseitemGru) | Valid command function | command= houseitemGru | Program follow command | start item is removed from the item bag and labels pop up | - |
| 26 | Button(house,text="Use item",command=houseitemGru) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 27 | Button(house,text="Use item",command=houseitemGru) | Function doesn’t not exit | Delete function  houseitemGru | Error massage | Program crashed- function do not define | Double check command |
| 28 | Button(house,text="continue",command=bedroom) | Valid data | Click Button | Program follow command | New window pop up – program continue as normal | - |
| 29 | Button(house,text="continue",command=bedroom) | Valid command function | command=bedroom | Program follow command | New window pop up – program continue as normal | - |
| 30 | Button(house,text="continue",command=bedroom) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 31 | Button(house,text="continue",command=bedroom) | Function doesn’t not exit | Delete function  bedroom | Error massage | Program crashed- function do not define | Double check command |
| 32 | Button(bed,text="Leave the room",anchor="e",command=Kitchen) | Valid data | Click Button | Program follow command | New window pop up – program follow command | - |
| 33 | Button(bed,text="Leave the room",anchor="e",command=Kitchen) | Valid command function | command= Kitchen | Program follow command | New window pop up – program follow command | - |
| 34 | Button(bed,text="Leave the room",anchor="e",command=Kitchen) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 35 | Button(bed,text="Leave the room",anchor="e",command=Kitchen) | Function doesn’t not exit | Delete function  Kitchen | Error massage | Program crashed- function do not define | Double check command |
| 36 | Button(bed,text="collect Margo phone",anchor="w",command=MargoPhone) | Valid data | Click Button | Program follow command | Phone add to item list, labels display - program run as normal | - |
| 37 | Button(bed,text="collect Margo phone",anchor="w",command=MargoPhone) | Valid command function | command=MargoPhone | Program follow command | Phone add to item list, labels display - program run as normal | - |
| 38 | Button(bed,text="collect Margo phone",anchor="w",command=MargoPhone) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 39 | Button(bed,text="collect Margo phone",anchor="w",command=MargoPhone) | Function doesn’t not exit | Delete function  MargoPhone | Error massage | Program crashed- function do not define | Double check command |
| 40 | Button(bed,text="continue",anchor="e",command=Kitchen) | Valid data | Click Button | Program follow command | New window pop up – program follow command | - |
| 41 | Button(bed,text="continue",anchor="e",command=Kitchen) | Valid command function | command= Kitchen | Program follow command | New window pop up – program follow command | - |
| 42 | Button(bed,text="continue",anchor="e",command=Kitchen) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 43 | Button(bed,text="continue",anchor="e",command=Kitchen) | Function doesn’t not exit | Delete function  Kitchen | Error massage | Program crashed- function do not define | Double check command |
| 44 | Button(kit,text="help Edith",command=HelpEdith) | Valid data | Click Button | Program follow command | List box pop up – program run as normal | - |
| 45 | Button(kit,text="help Edith",command=HelpEdith) | Valid command function | command=HelpEdith | Program follow command | List box pop up – program run as normal | - |
| 46 | Button(kit,text="help Edith",command=HelpEdith) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 47 | Button(kit,text="help Edith",command=HelpEdith) | Function doesn’t not exit | Delete function  HelpEdith | Error massage | Program crashed- function do not define | Double check command |
| 48 | Button(kit,text="dont't have phone",command=GruRoom) | Valid data | Click Button | Program follow command | New window (Gru’s office) pop up- program follow command | - |
| 49 | Button(kit,text="dont't have phone",command=GruRoom) | Valid command function | command=GruRoom | Program follow command | New window (Gru’s office) pop up- program follow command | - |
| 50 | Button(kit,text="dont't have phone",command=GruRoom) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 51 | Button(kit,text="dont't have phone",command=GruRoom) | Function doesn’t not exit | Delete function GruRoom | Error massage | Program crashed- function do not define | Double check command |
| 52 | Button(kit,text="Continue",command=GruRoom) | Valid data | Click Button | Program follow command | New window (Gru’s office) pop up- program follow command | - |
| 53 | Button(kit,text="Continue",command=GruRoom) | Valid command function | command=GruRoom | Program follow command | New window (Gru’s office) pop up- program follow command | - |
| 54 | Button(kit,text="Continue",command=GruRoom) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 55 | Button(kit,text="Continue",command=GruRoom) | Function doesn’t not exit | Delete function GruRoom | Error massage | Program crashed- function do not define | Double check command |
| 56 | Button(kit,text="Select",command=selectedItemHelpEdith) | Valid data | Click Button | Program follow command | Phone is removed from list and Pancake is added to list | Selected item is removed from the list |
| 57 | Button(kit,text="Select",command=selectedItemHelpEdith) | Valid command function | command= selectedItemHelpEdith | Program follow command | Phone is removed from list and Pancake is added to list | Selected item is removed from the list |
| 58 | Button(kit,text="Select",command=selectedItemHelpEdith) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Error massage |
| 59 | Button(kit,text="Select",command=selectedItemHelpEdith) | Function doesn’t not exit | Delete function selectedItemHelpEdith | Error massage | Program crashed- function do not define | Error massage |
| 60 | Button(gru,text="Exchange",command=ExchangeCarl) | Valid data | Click Button | Program follow command | Pancake is removed from list and Siren hat is added to list | Selected item is removed from the list |
| 61 | Button(gru,text="Exchange",command=ExchangeCarl) | Valid command function | command= ExchangeCarl | Program follow command | Pancake is removed from list and Siren hat is added to list | Selected item is removed from the list |
| 62 | Button(gru,text="Exchange",command=ExchangeCarl) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 63 | Button(gru,text="Exchange",command=ExchangeCarl) | Function doesn’t not exit | Delete function  ExchangeCarl | Error massage | Program crashed- function do not define | Display Error massage |
| 64 | Button(gru,text="Don't have pancake", command=Backyard) | Valid data | Click Button | Program follow command | New window (Backyard)pop up – program follow command | - |
| 65 | Button(gru,text="Don't have pancake", command=Backyard) | Valid command function | command= Backyard | Program follow command | New window (Backyard)pop up – program follow command | - |
| 66 | Button(gru,text="Don't have pancake", command=Backyard) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage  Double check spelling |
| 67 | Button(gru,text="Don't have pancake", command=Backyard) | Function doesn’t not exit | Delete function  Backyard | Error massage | Program crashed- function do not define | Display Error massage |
| 68 | Button(gru,text="Continue",command=Backyard) | Valid data | Click Button | Program follow command | New window (Backyard)pop up – program follows command | - |
| 69 | Button(gru,text="Continue",command=Backyard) | Valid command function | command= Backyard | Program follow command | New window (Backyard)pop up – program follows command | - |
| 70 | Button(gru,text="Continue",command=Backyard) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage  Double check spelling |
| 71 | Button(gru,text="Continue",command=Backyard) | Function doesn’t not exit | Delete function  Backyard | Error massage | Program crashed- function do not define | Display Error massage |
| 72 | Button(back,text="return the spatular",command=returnSpatula) | Valid data | Click Button | Program follow command | Label pop up – program continue as normal | - |
| 73 | Button(back,text="return the spatular",command=returnSpatula) | Valid command function | command= returnSpatula | Program follow command | Label pop up – program continue as normal | - |
| 74 | Button(back,text="return the spatular",command=returnSpatula) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 75 | Button(back,text="return the spatular",command=returnSpatula) | Function doesn’t not exit | Delete function  returnSpatula | Error massage | Program crashed- function do not define | Display Error massage |
| 76 | Button(back,text="Keep the spactular",command=keepSpatulat) | Valid data | Click Button | Program follow command | Spatula is added to list - program follow command | - |
| 77 | Button(back,text="Keep the spactular",command=keepSpatulat) | Valid command function | command= returnSpatula | Program follow command | Spatula is added to list - program follow command | - |
| 78 | Button(back,text="Keep the spactular",command=keepSpatulat) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 79 | Button(back,text="Keep the spactular",command=keepSpatulat) | Function doesn’t not exit | Delete function  returnSpatula | Error massage | Program crashed- function do not define | Display Error massage |
| 80 | Button(back,text="continue",command=Lab) | Valid data | Click Button | Program follow command | New window (Minion Lab) pop up – program follow command | - |
| 81 | Button(back,text="continue",command=Lab) | Valid command function | command= Lab | Program follow command | New window (Minion Lab) pop up – program follow command | - |
| 82 | Button(back,text="continue",command=Lab) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage  Double check spelling |
| 83 | Button(back,text="continue",command=Lab) | Function doesn’t not exit | Delete function  Lab | Error massage | Program crashed- function do not define | Display Error massage |
| 84 | Button(lab,text="Help him",anchor='w',command=HelpNefario) | Valid data | Click Button | Program follow command | List box pop up – program follow command | - |
| 85 | Button(lab,text="Help him",anchor='w',command=HelpNefario) | Valid command function | command= HelpNefario | Program follow command | List box pop up – program follow command | - |
| 86 | Button(lab,text="Help him",anchor='w',command=HelpNefario) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Double check spelling |
| 87 | Button(lab,text="Help him",anchor='w',command=HelpNefario) | Function doesn’t not exit | Delete function  HelpNefario | Error massage | Program crashed- function do not define | Display Error massage |
| 88 | Button(lab,text="Ignor him",anchor='e',command=Roof) | Valid data | Click Button | Program follow command | New window ( Roof) pop up – program follow command | - |
| 89 | Button(lab,text="Ignor him",anchor='e',command=Roof) | Valid command function | command= Roof | Program follow command | New window ( Roof) pop up – program follow command | - |
| 90 | Button(lab,text="Ignor him",anchor='e',command=Roof) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 91 | Button(lab,text="Ignor him",anchor='e',command=Roof) | Function doesn’t not exit | Delete function  Roof | Error massage | Program crashed- function do not define | Display Error massage |
| 92 | Button(lab, text = "select",command=print\_HelpNafario) | Valid data | Click Button | Program follow command | Label pop out when each item is selected | - |
| 93 | Button(lab, text = "select",command=print\_HelpNafario) | Valid command function | command= print\_HelpNafario | Program follow command | Label pop out when each item is selected | - |
| 94 | Button(lab, text = "select",command=print\_HelpNafario) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage   Build back-up command function |
| 95 | Button(lab, text = "select",command=print\_HelpNafario) | Function doesn’t not exit | Delete function  print\_HelpNafario) | Error massage | Program crashed- function do not define | Display Error massage   Build back-up command function |
| 96 | Button(lab,text="Continue",command=RoofHelp) | Valid data | Click Button | Program follow command | New window pop up – program follow command | - |
| 97 | Button(lab,text="Continue",command=RoofHelp) | Valid command function | command= RoofHelp | Program follow command | New window pop up – program follow command | - |
| 98 | Button(lab,text="Continue",command=RoofHelp) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 99 | Button(lab,text="Continue",command=RoofHelp) | Function doesn’t not exit | Delete function  RoofHelp | Error massage | Program crashed- function do not define | Display Error massage |
| 100 | Button(roof,text="roll dice",command=rollDiceVector) | Valid data | Click Button | Program follow command | Program continue as normal | - |
| 101 | Button(roof,text="roll dice",command=rollDiceVector) | Valid command function | command= rollDiceVector | Program follow command | Program continue as normal | - |
| 102 | Button(roof,text="roll dice",command=rollDiceVector) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 103 | Button(roof,text="roll dice",command=rollDiceVector) | Function doesn’t not exit | Delete function  rollDiceVector | Error massage | Program crashed- function do not define | Display Error massage |
| 104 | Button(roofH,text="Use Freez Ray",command=FreezRay) | Valid data | Click Button | Program follow command | Labels and Button pop up – program follow command | - |
| 105 | Button(roofH,text="Use Freez Ray",command=FreezRay) | Valid command function | command= FreezRay | Program follow command | Labels and Button pop up – program follow command | - |
| 106 | Button(roofH,text="Use Freez Ray",command=FreezRay) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 107 | Button(roofH,text="Use Freez Ray",command=FreezRay) | Function doesn’t not exit | Delete function  FreezRay | Error massage | Program crashed- function do not define | Display Error massage |
| 108 | Button(roofH,text="roll dice",command=rollDiceVectorH) | Valid data | Click Button | Program follow command | +2 point on number get from rolling dice | - |
| 109 | Button(roofH,text="roll dice",command=rollDiceVectorH) | Valid command function | command= rollDiceVectorH | Program follow command | +2 point on number get from rolling dice | - |
| 110 | Button(roofH,text="roll dice",command=rollDiceVectorH) | Function is renamed | def whatwhat(): | Error massage | Program crashed- function do not define | Display Error massage |
| 111 | Button(roofH,text="roll dice",command=rollDiceVectorH) | Function doesn’t not exit | Delete function  rollDiceVectorH | Error massage | Program crashed- function do not define | Display Error massage |
| 112 | Button(roof,text="Exite Game",command=quit) | Valid data | Click Button | Quit program | Program crash | Fix the command ‘quit’ |
| 113 | Button(roof,text="Exite Game",command=quit) | Valid command | command= quit | Quit program | Program crash | Fix the command ‘quit’ |
| 114 | Button(roof,text="Exite Game",command=quit) | Invalid command | Command=QuIt | Quit program | Program crash | Fix the command ‘quit’ |

# Screen shot

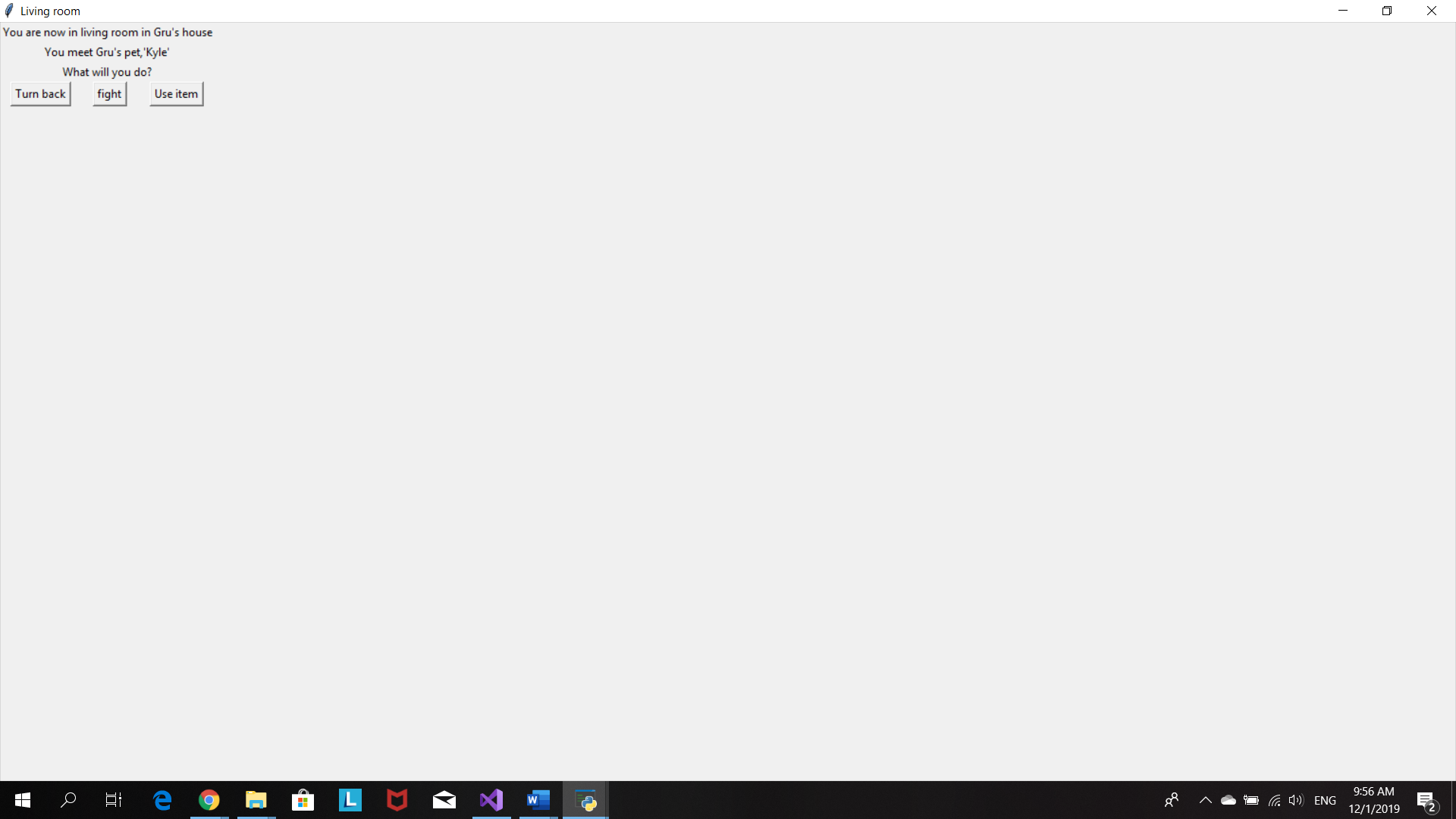
Select character window show up when run the program



Click on button Gru to choose him as player character

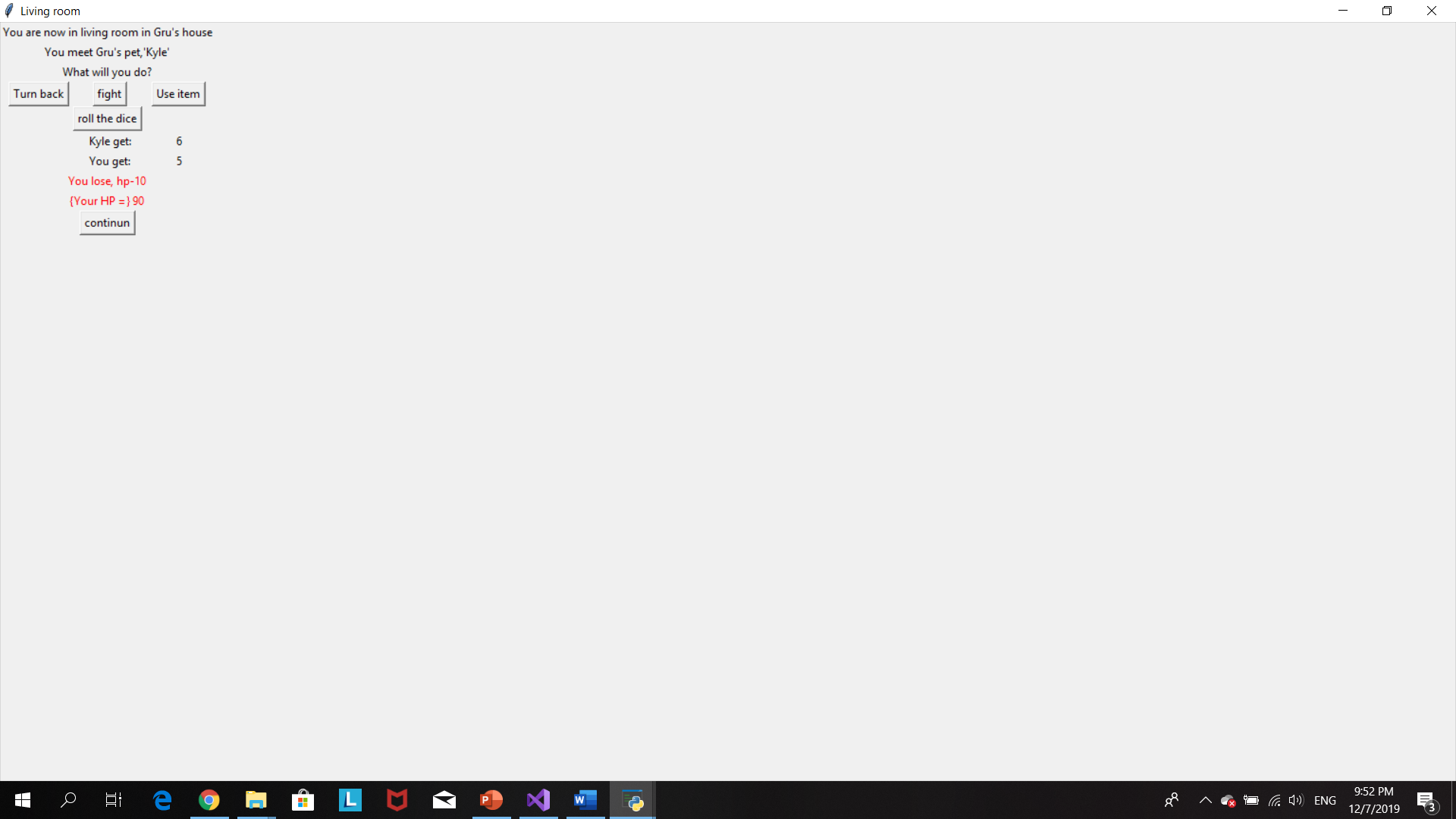


Click ‘start game’ and living room window pop up

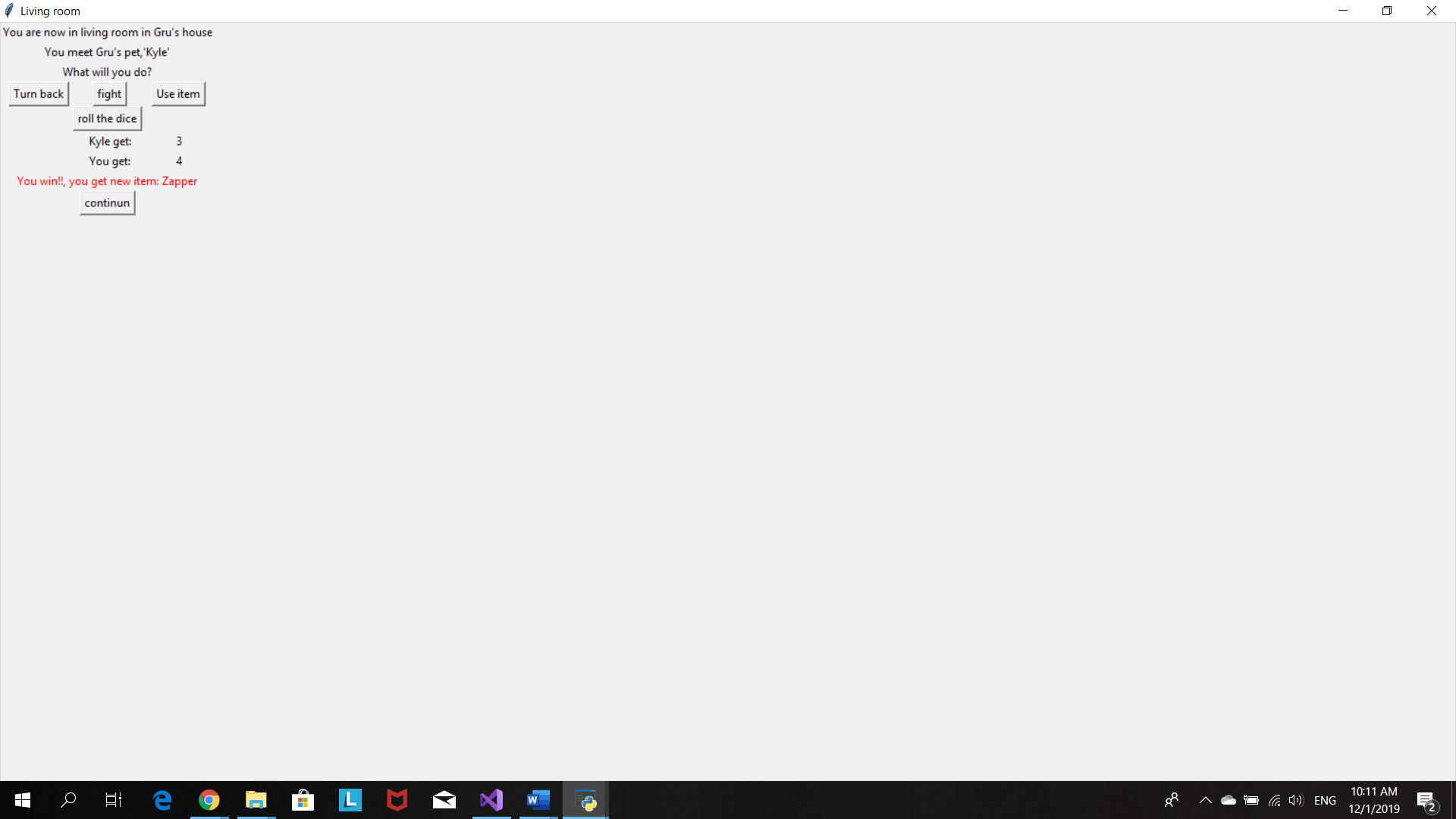


Click fight and the button roll dice will pop up then click on roll dice button the program will randon pick the number 1-6 for both player and Klye.

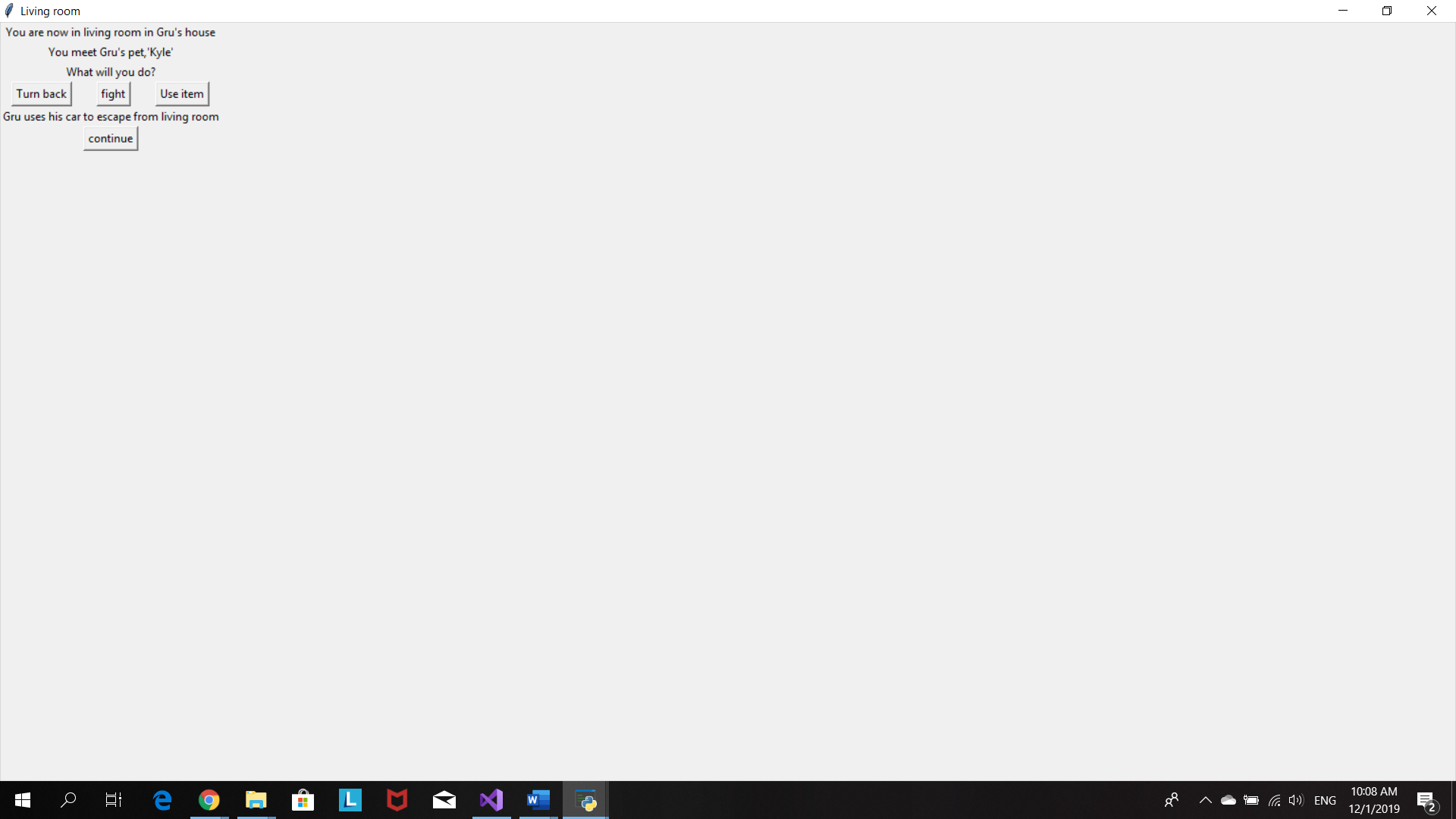
If player lost he/she will lose 10 hp



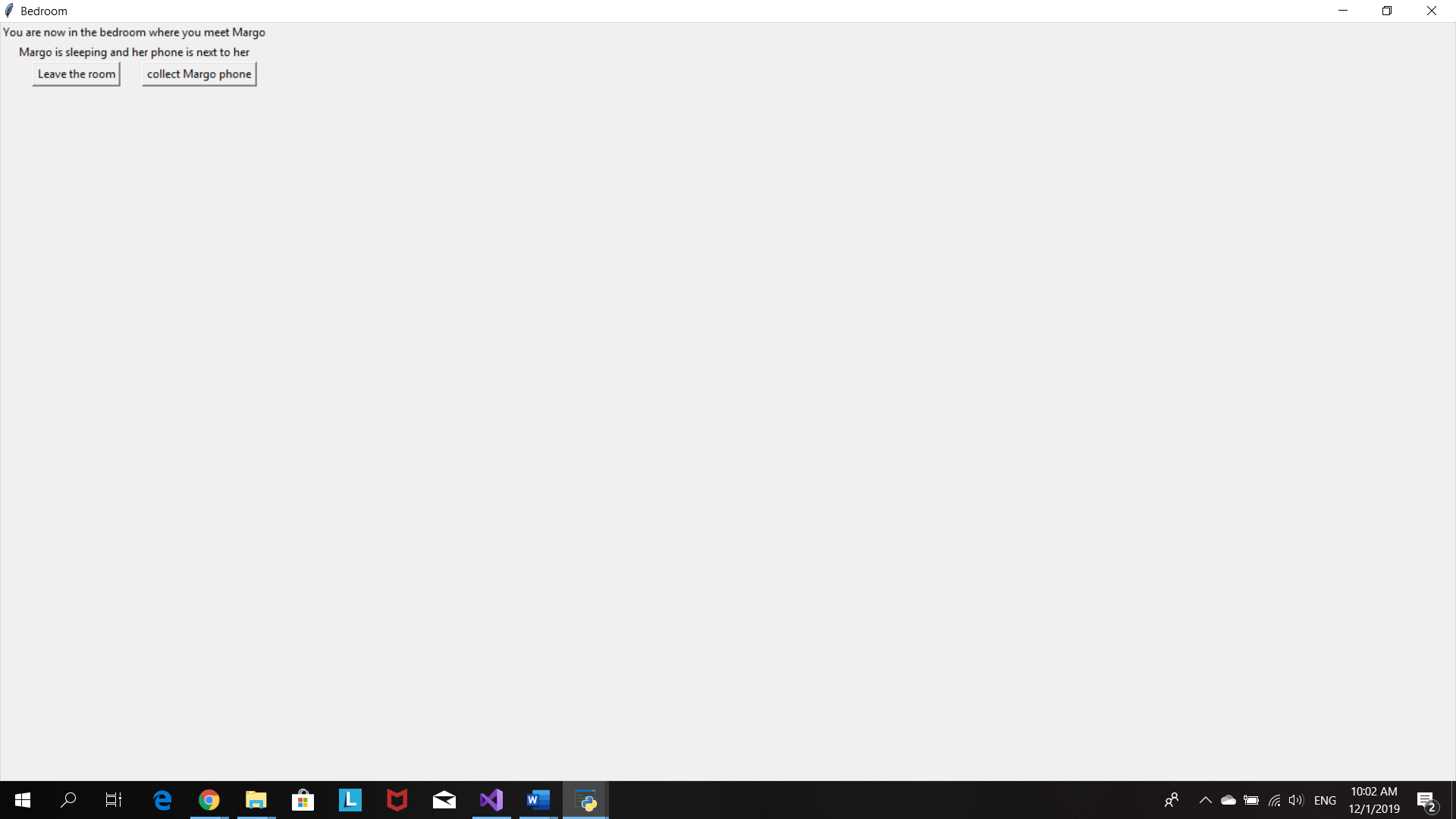
If player win they receive a Zapper item



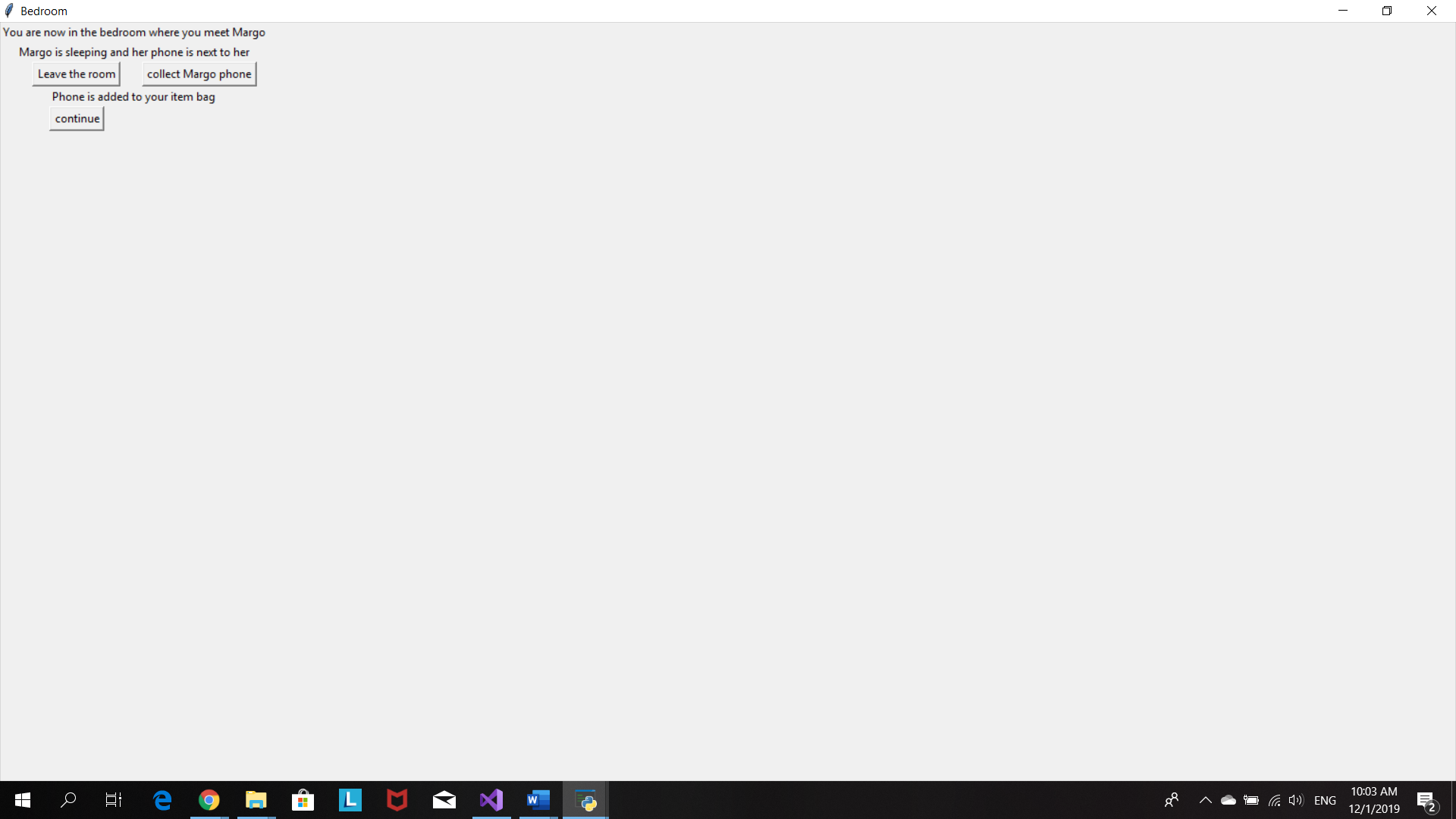
Start item will be romoved from bag when player click use item



Click continue and ‘Bedroom’ window will show up

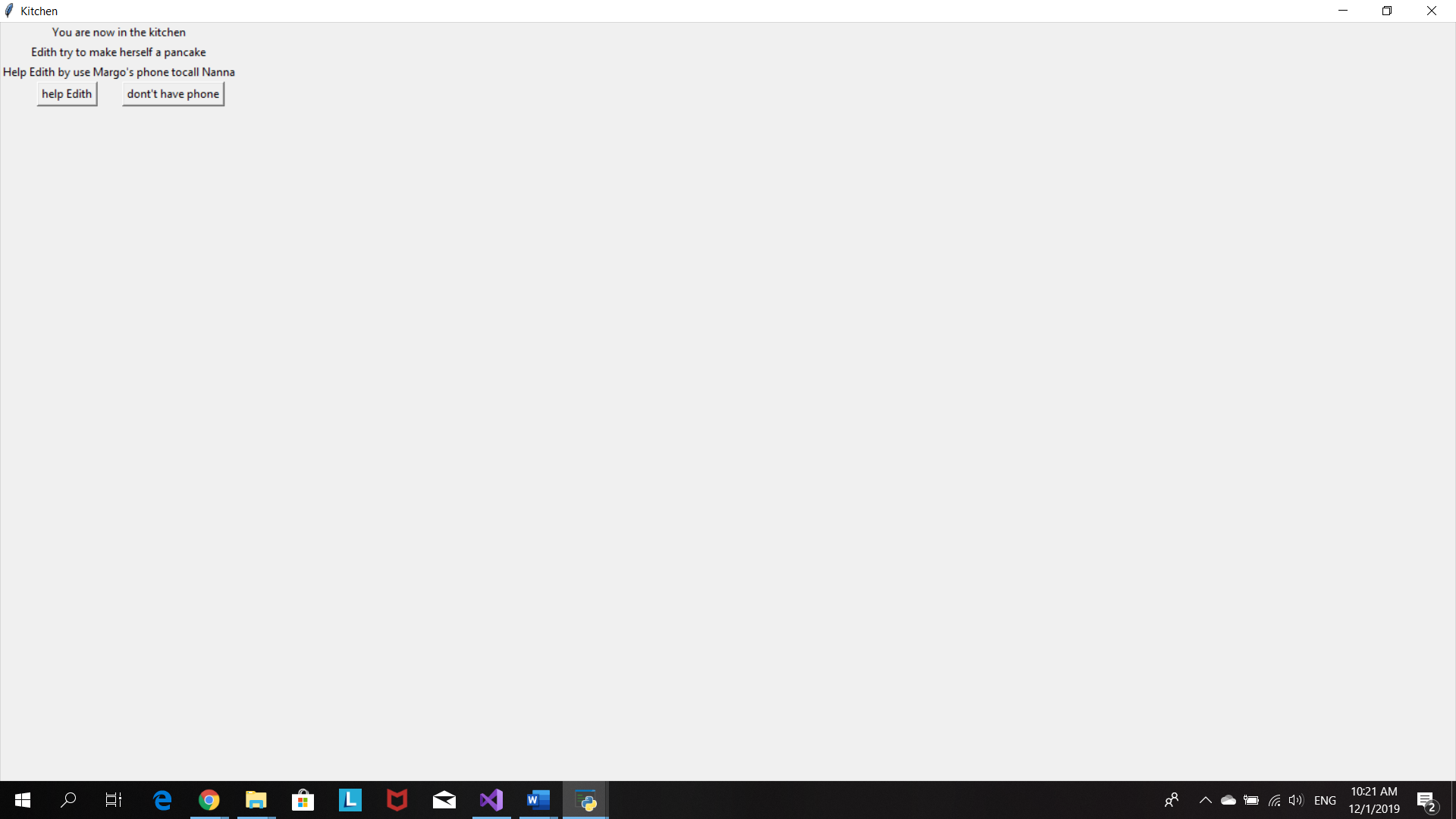


Click collect margo phone to add the item Phone into player bag.

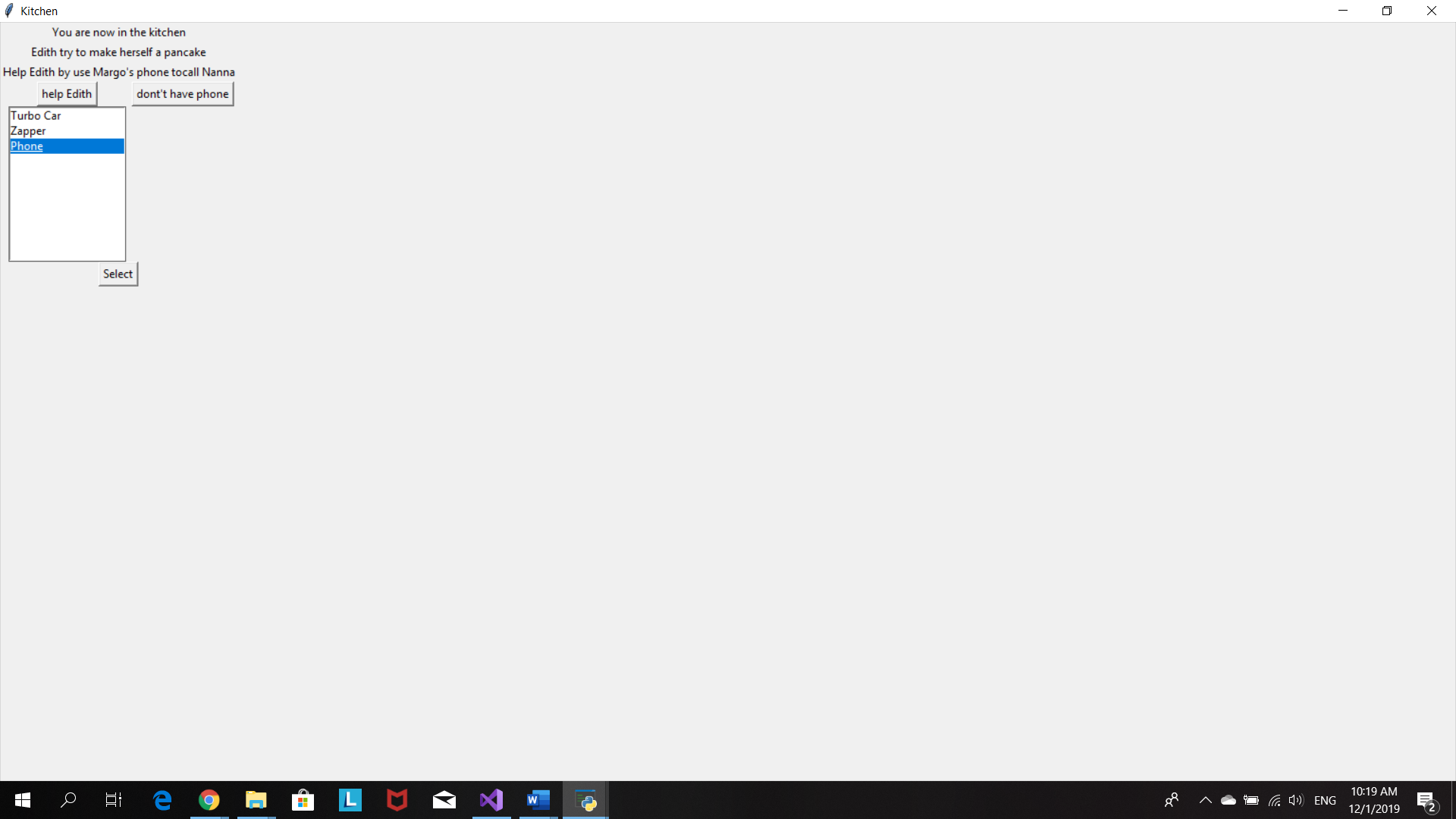


Or click leave room to go to the next scene window, kitchen .

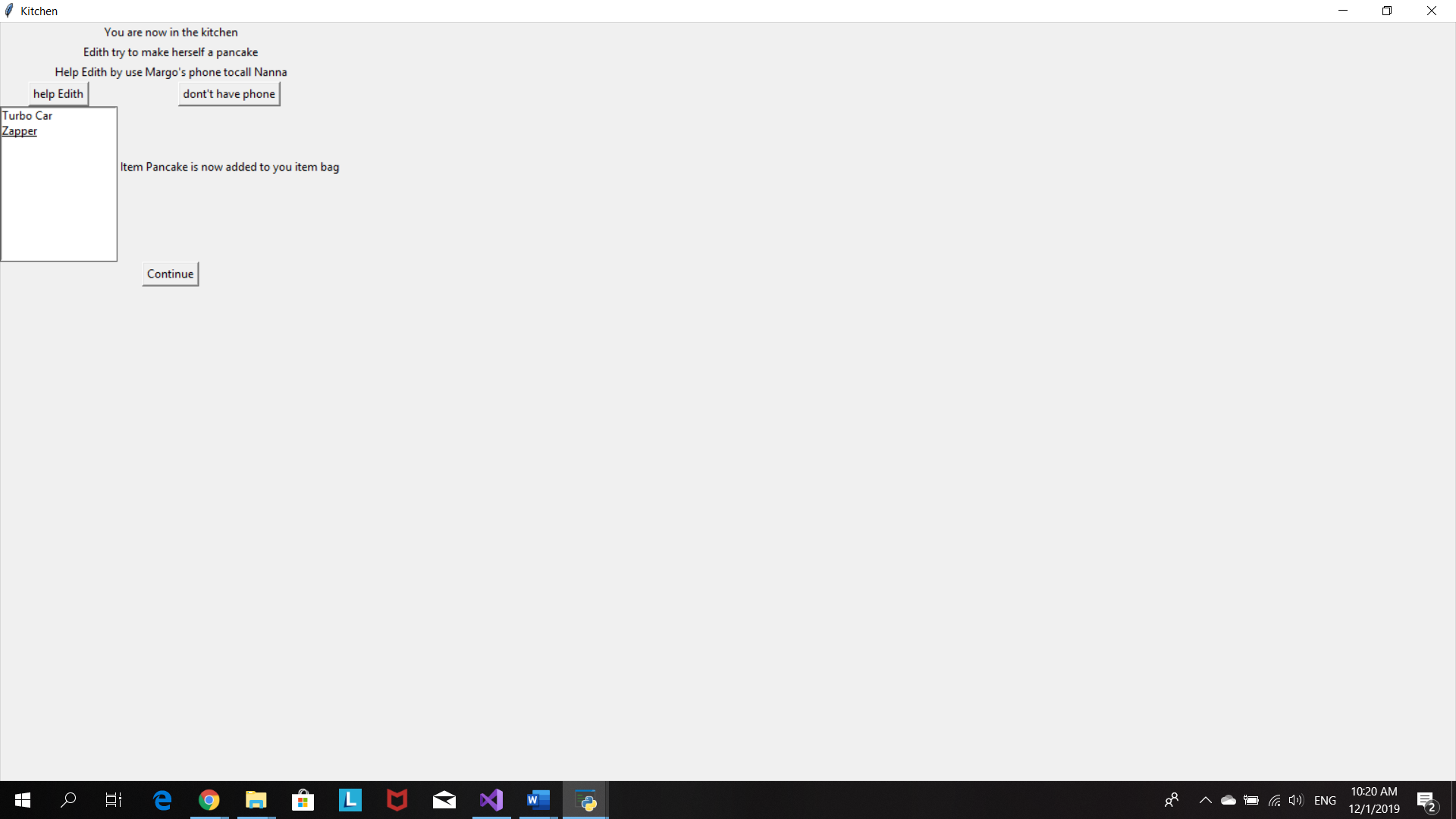
Kitchen window will pop up when player click continue after collect an Phone item as well



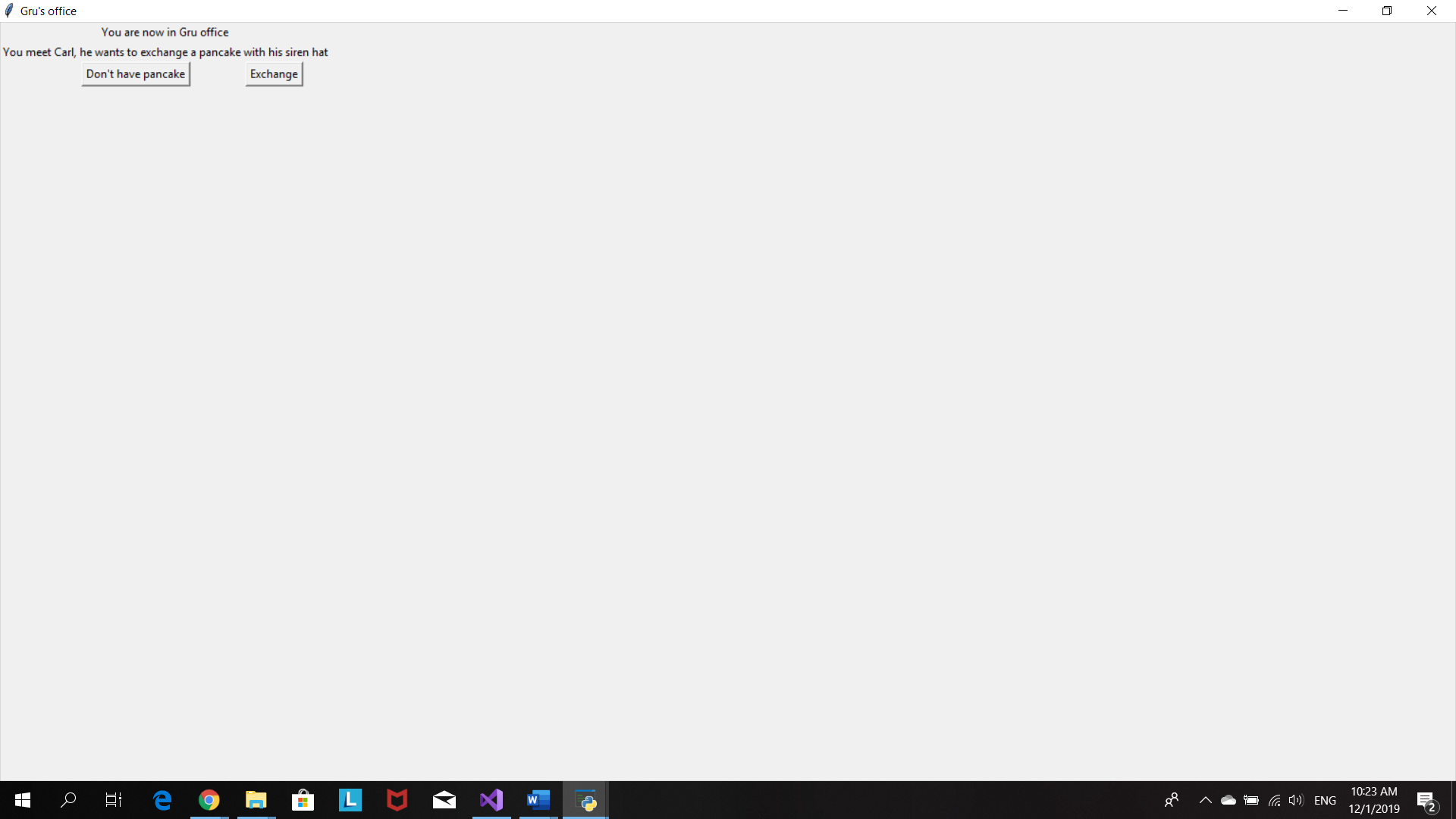
Player decied to help Edith make a pancake.Listbox pops up and all item in bag will show up for player to choose which item they want to use to help Edith



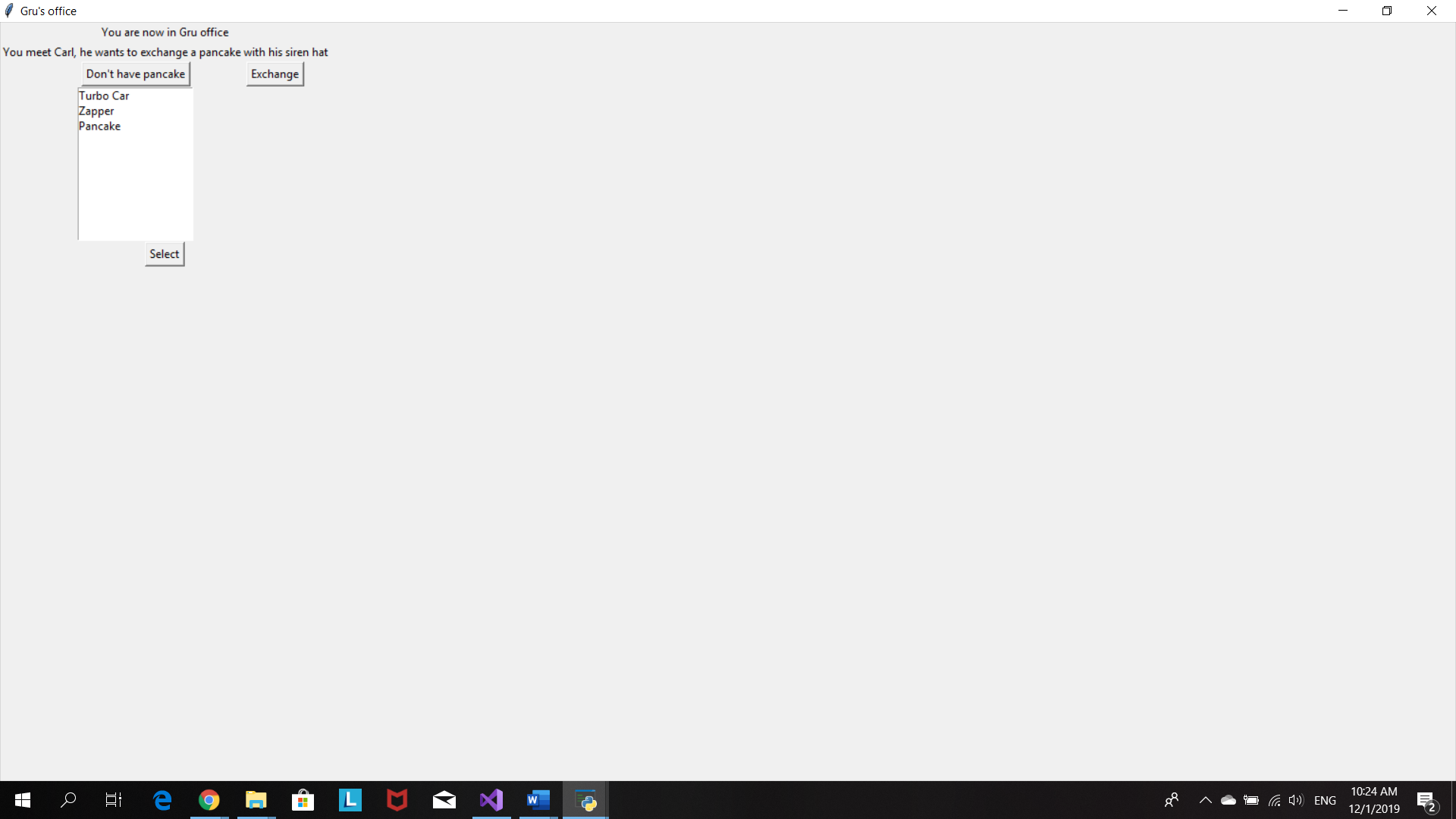
Player use phone to call Nanna. Phone is now removed from the bag and player receive new item, pancake



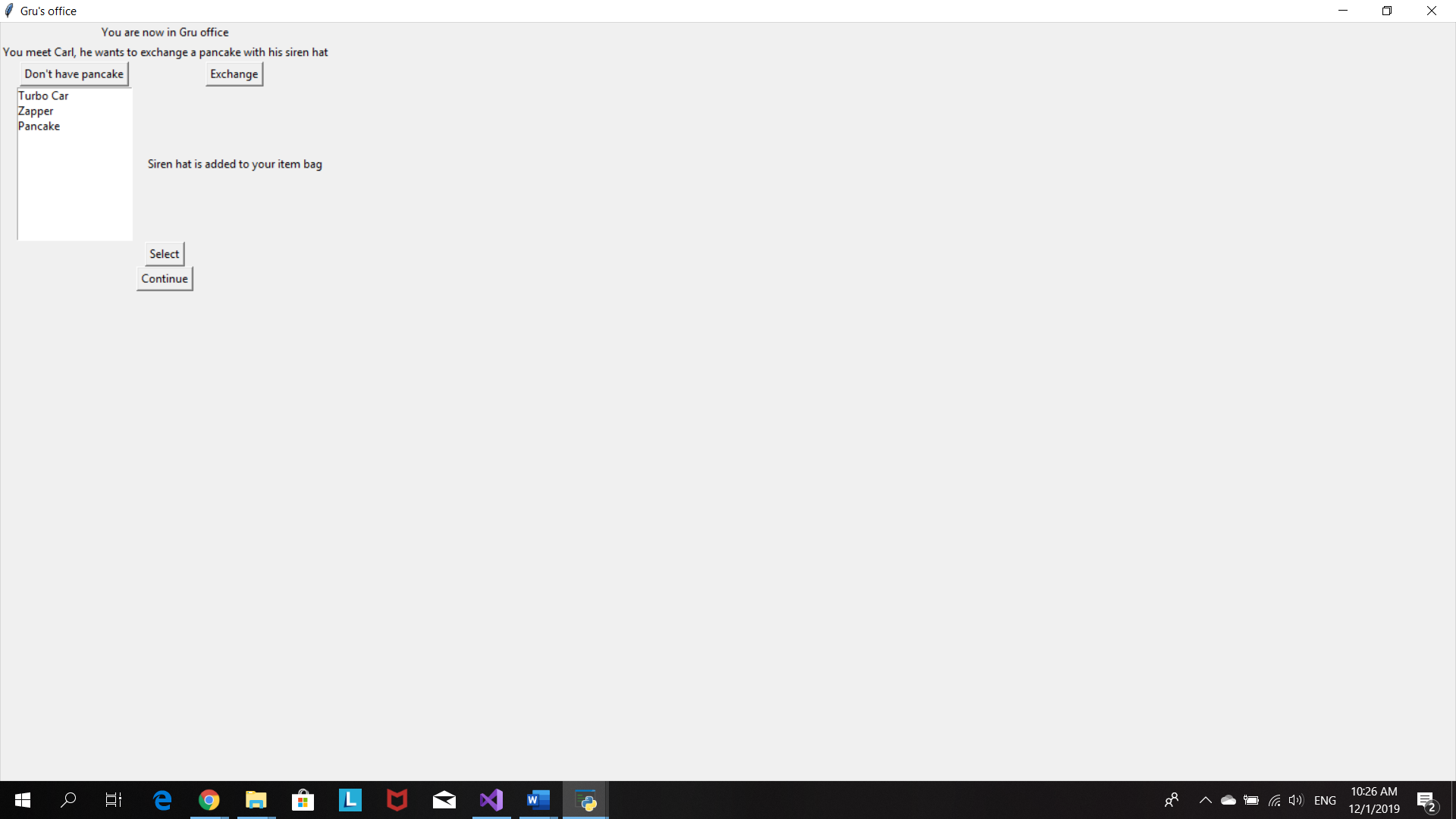
Gru’s office window will pop up when click on ‘continue’ or ‘don’t have phone’ buttons



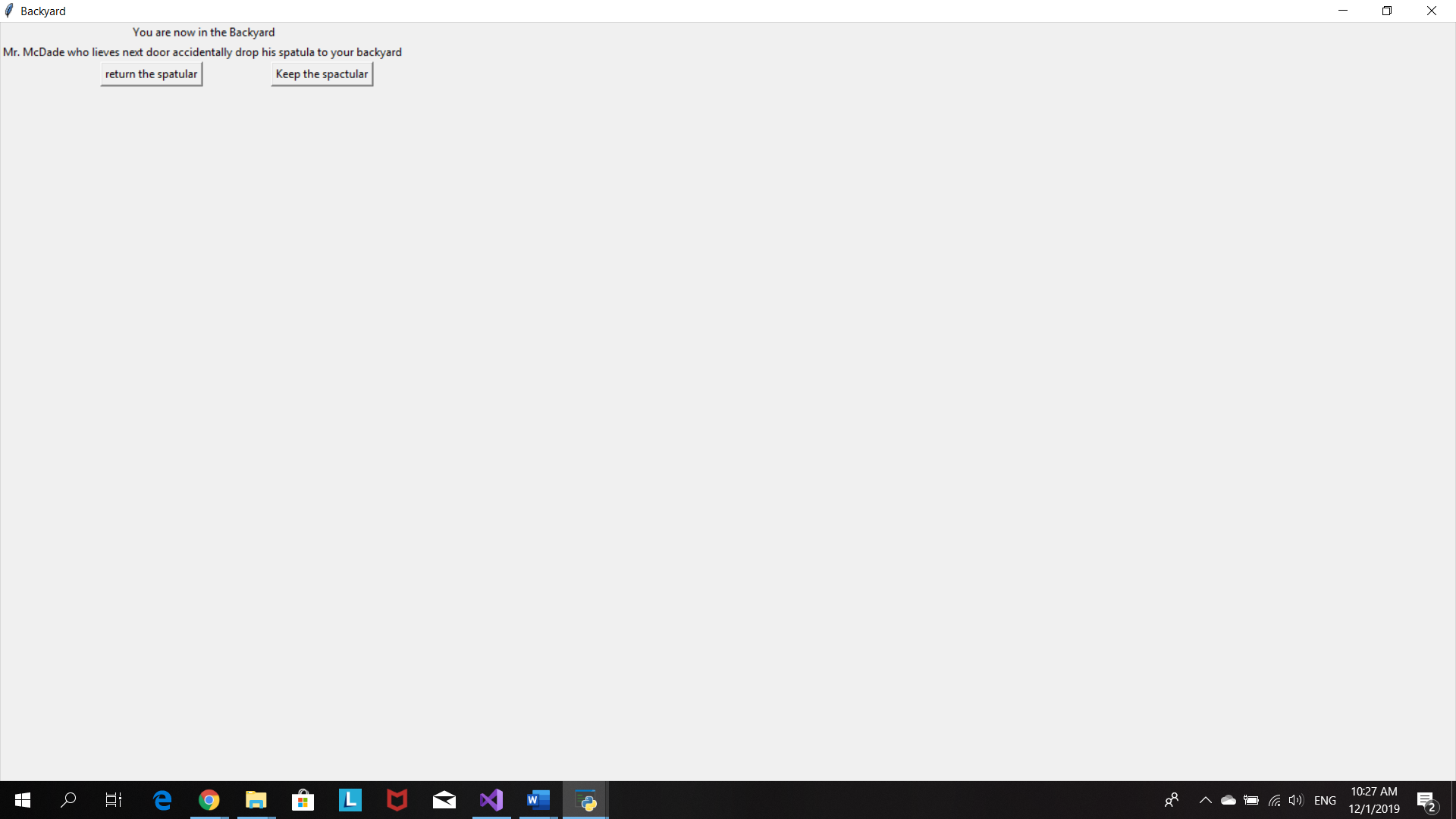
Player click on exchange button.Listbox pop up and player choose the item he/she wants to exchage with Carl’s Siren hat



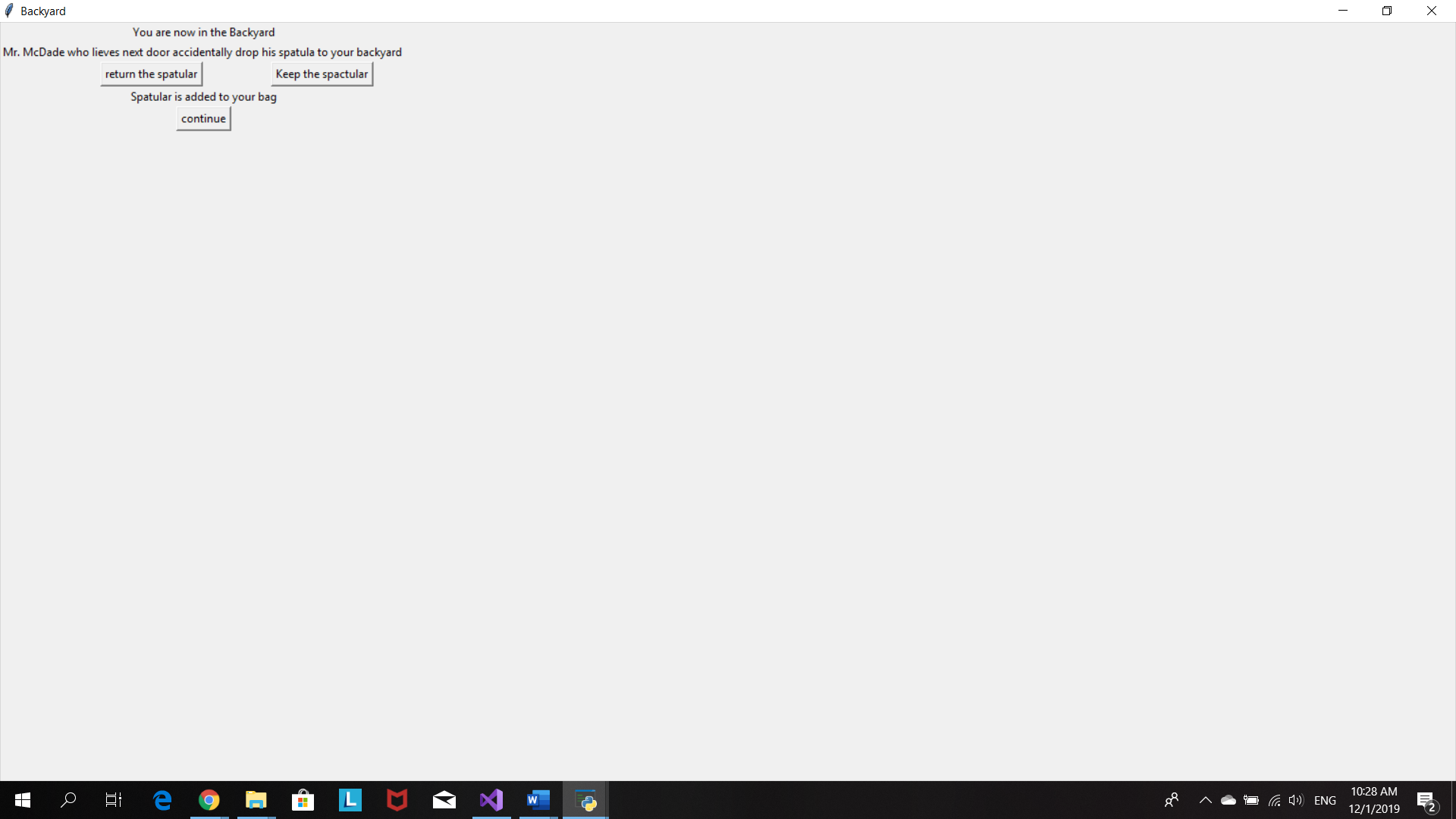
Player exchanges Pancake with Siren hat.



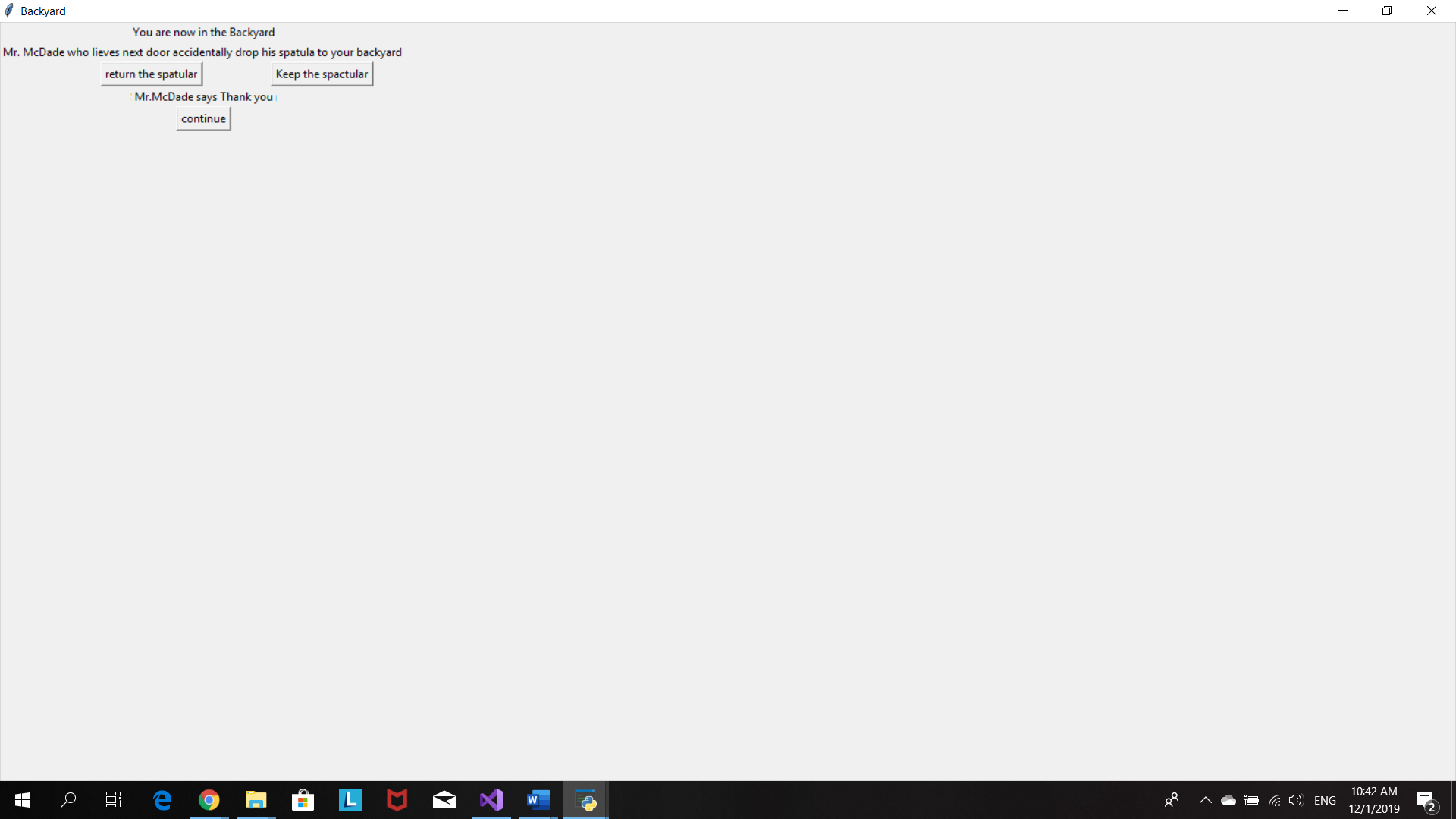
Both ‘don’t have pancake’ and ‘continue’ buttons will bring player to the Backyard window



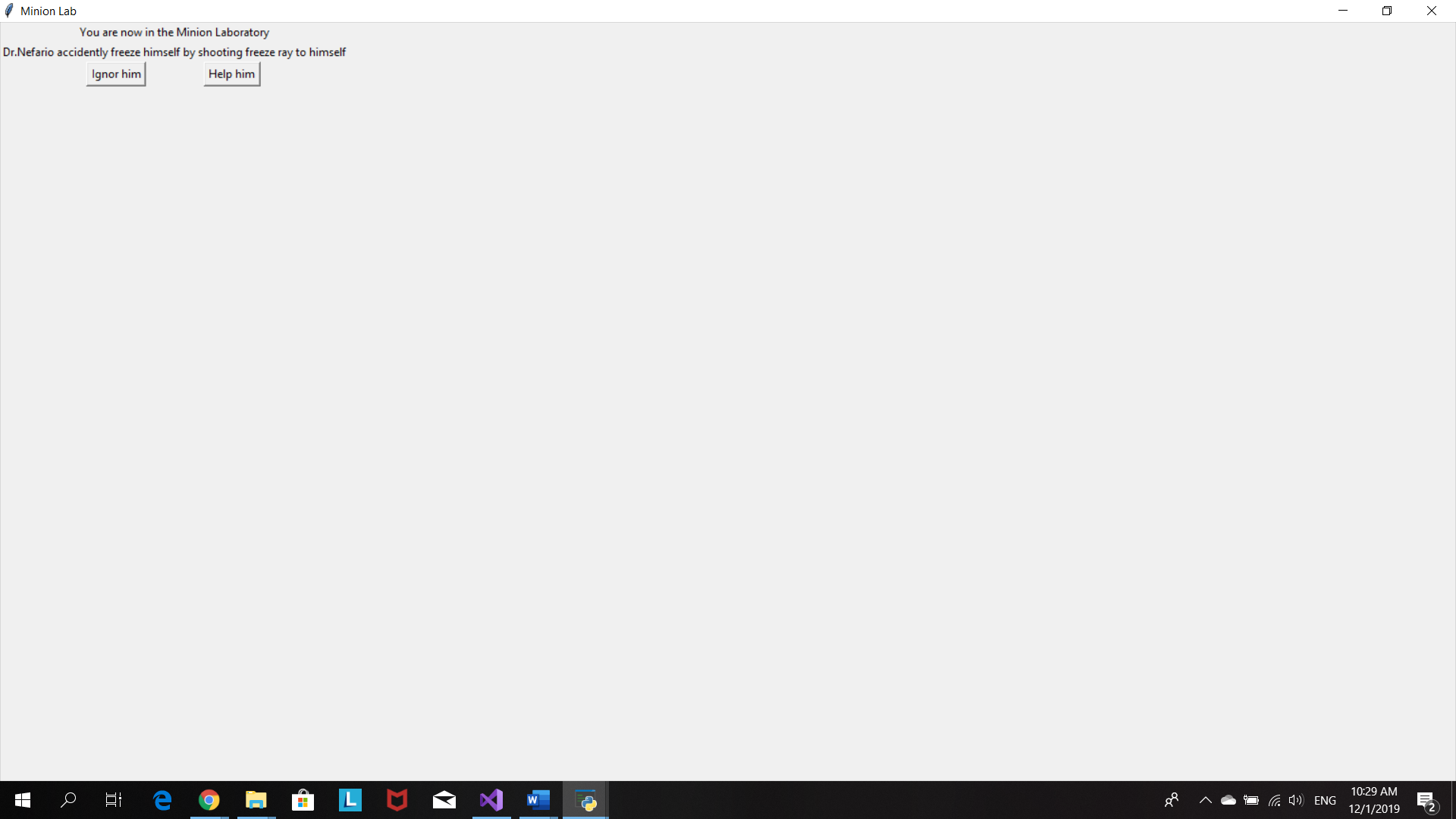
Player choose to collect Spatula



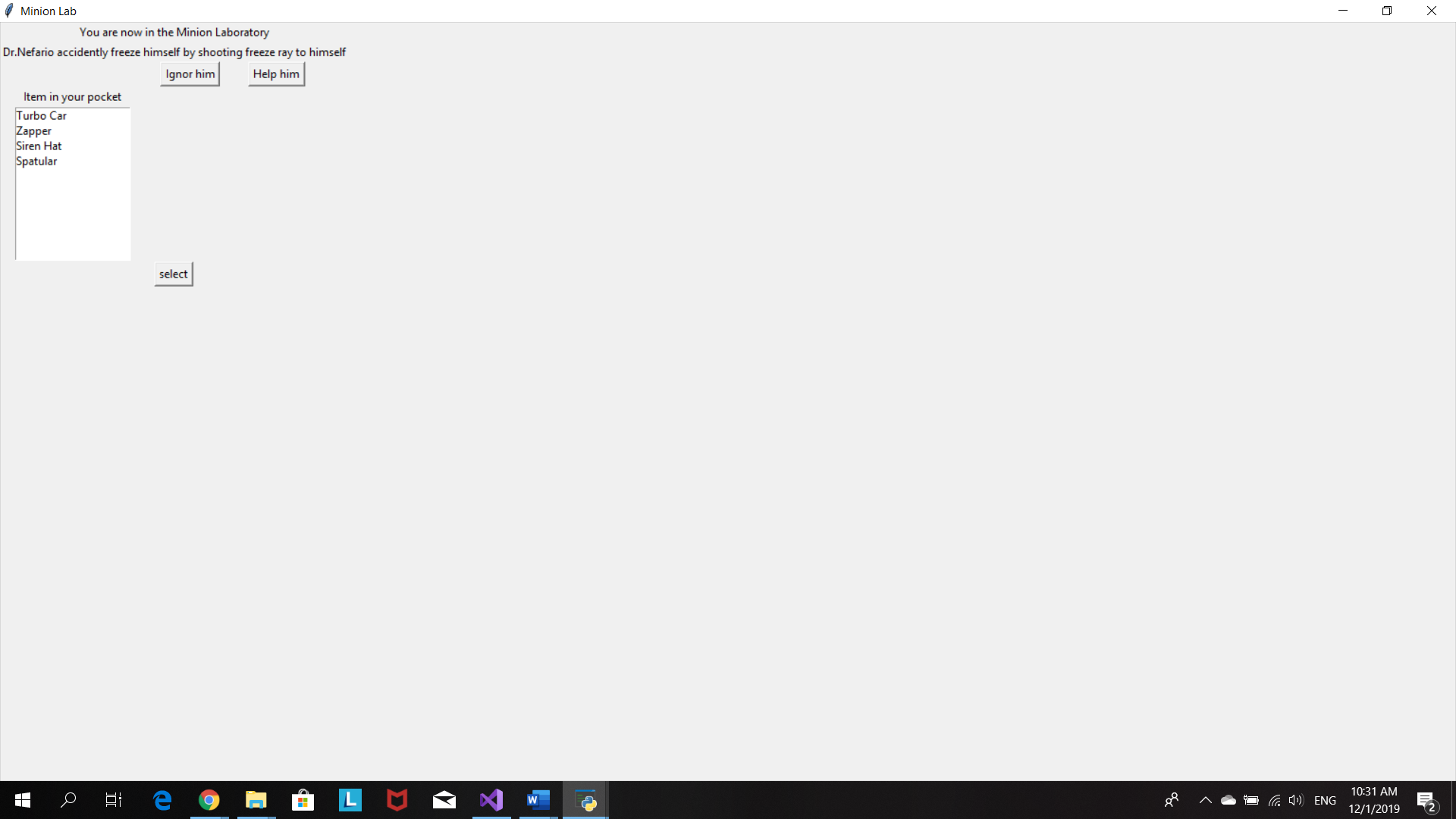
When player choose to return the Spactula



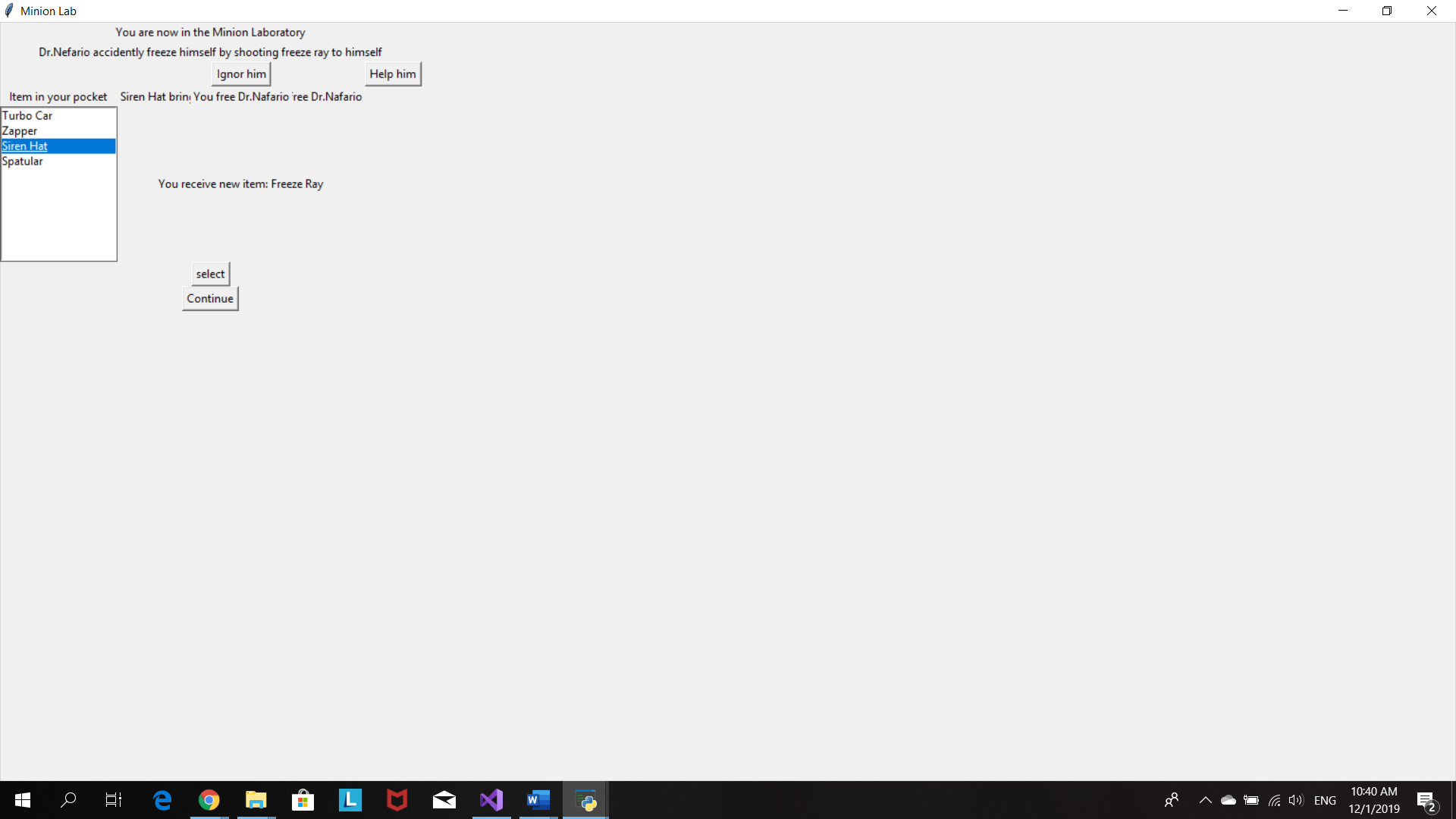
Continue button will bring player to this Minion lab window



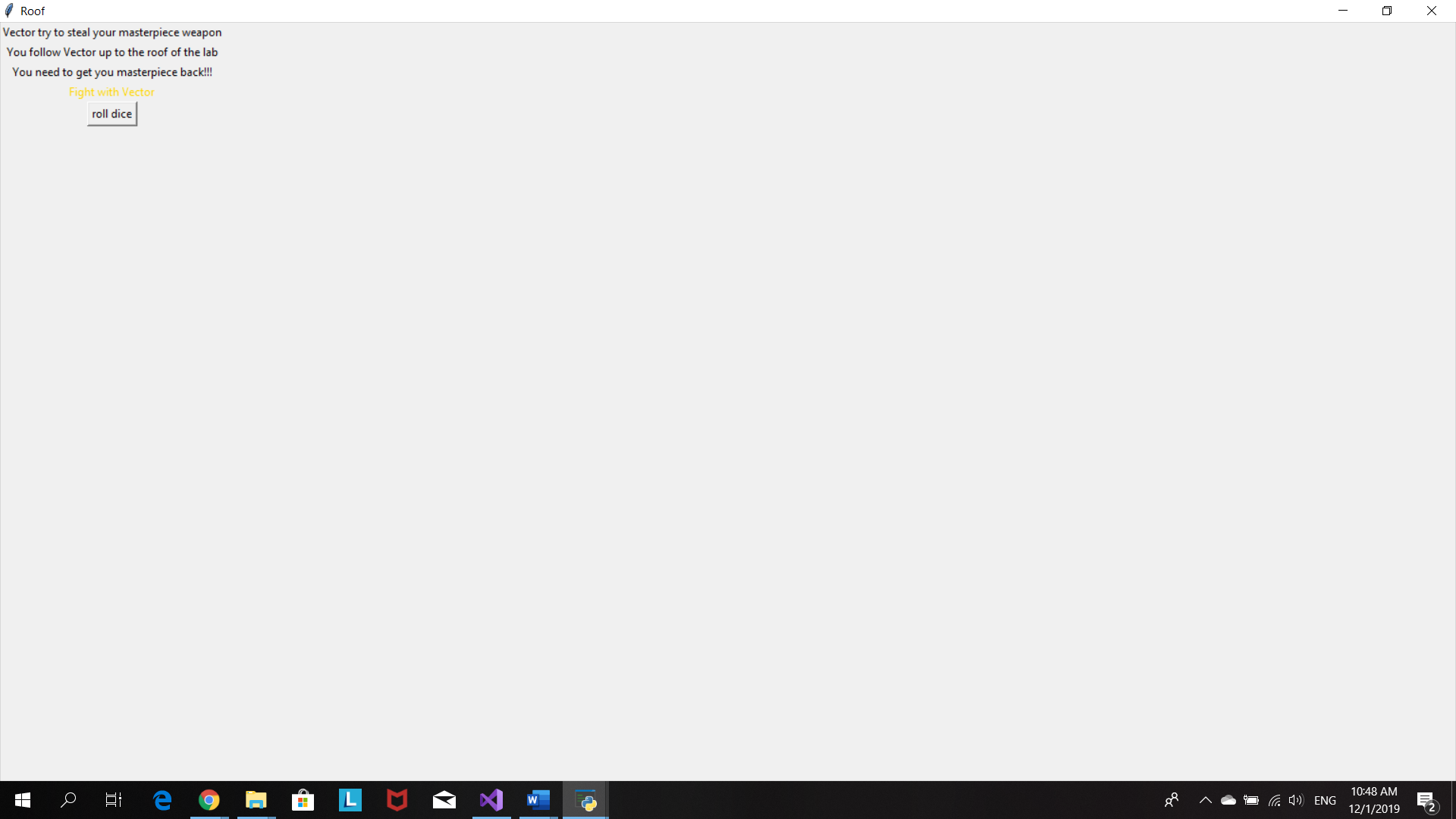
When player chooses to help Dr. Naferio, he/she will have to choose an item to help free Dr. Naferio from the ice



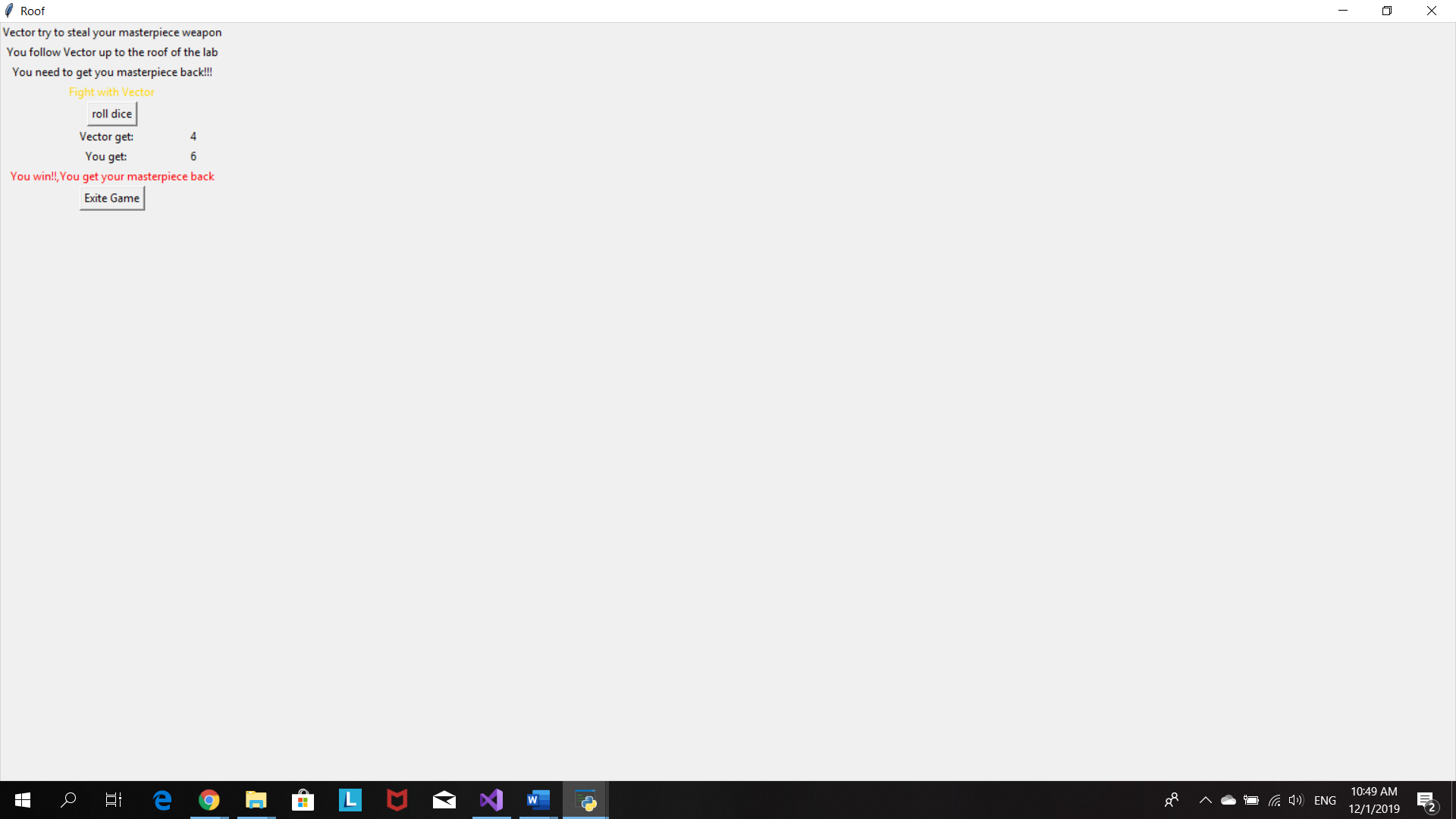
Player choose siren hat. Player free Dr. Nefario and get a new item frezze Ray



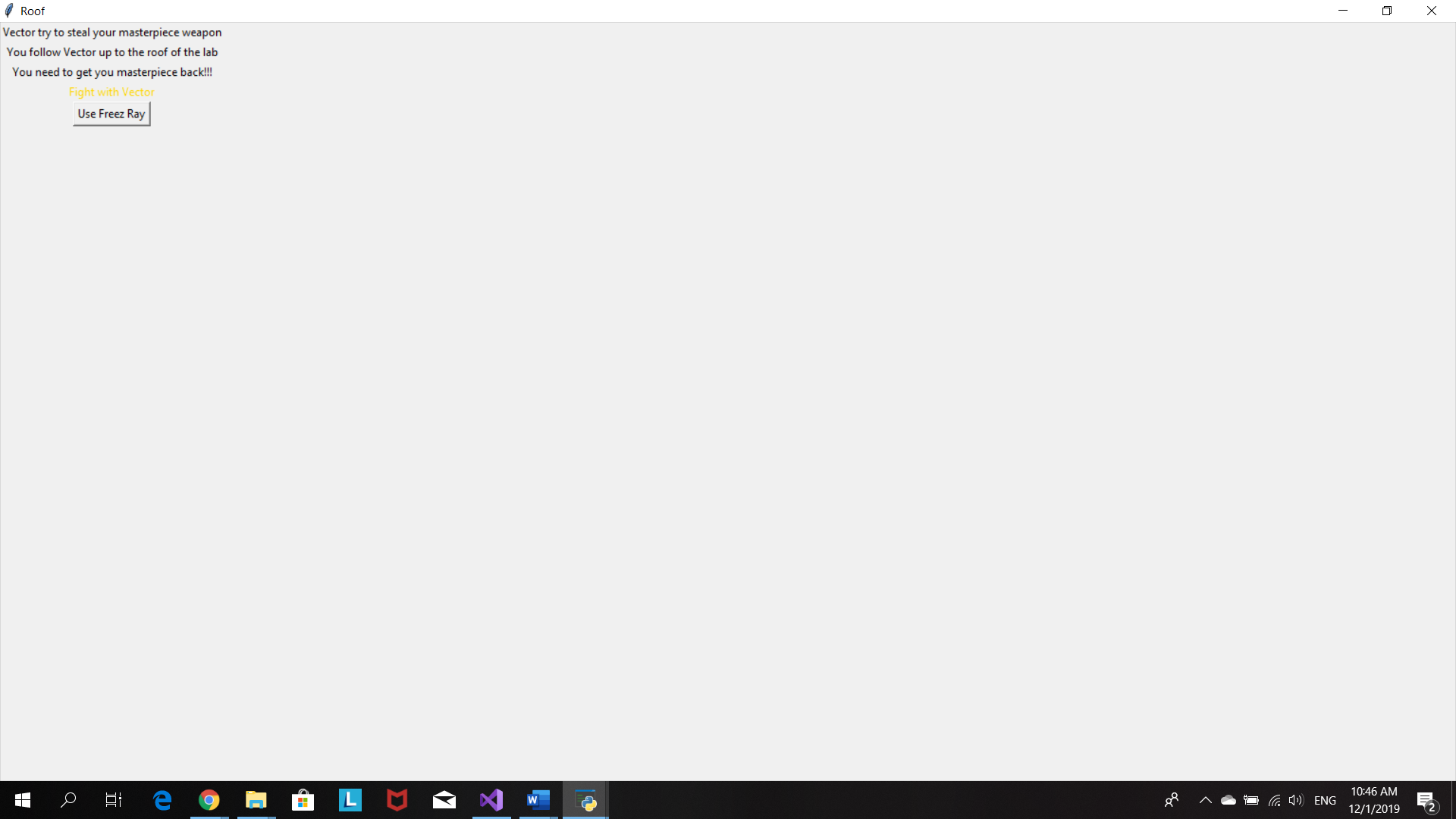
Player chooses to ignor Dr.Naferio so he/she doesn’t get the Freeze Ray.Therefore, no extra point



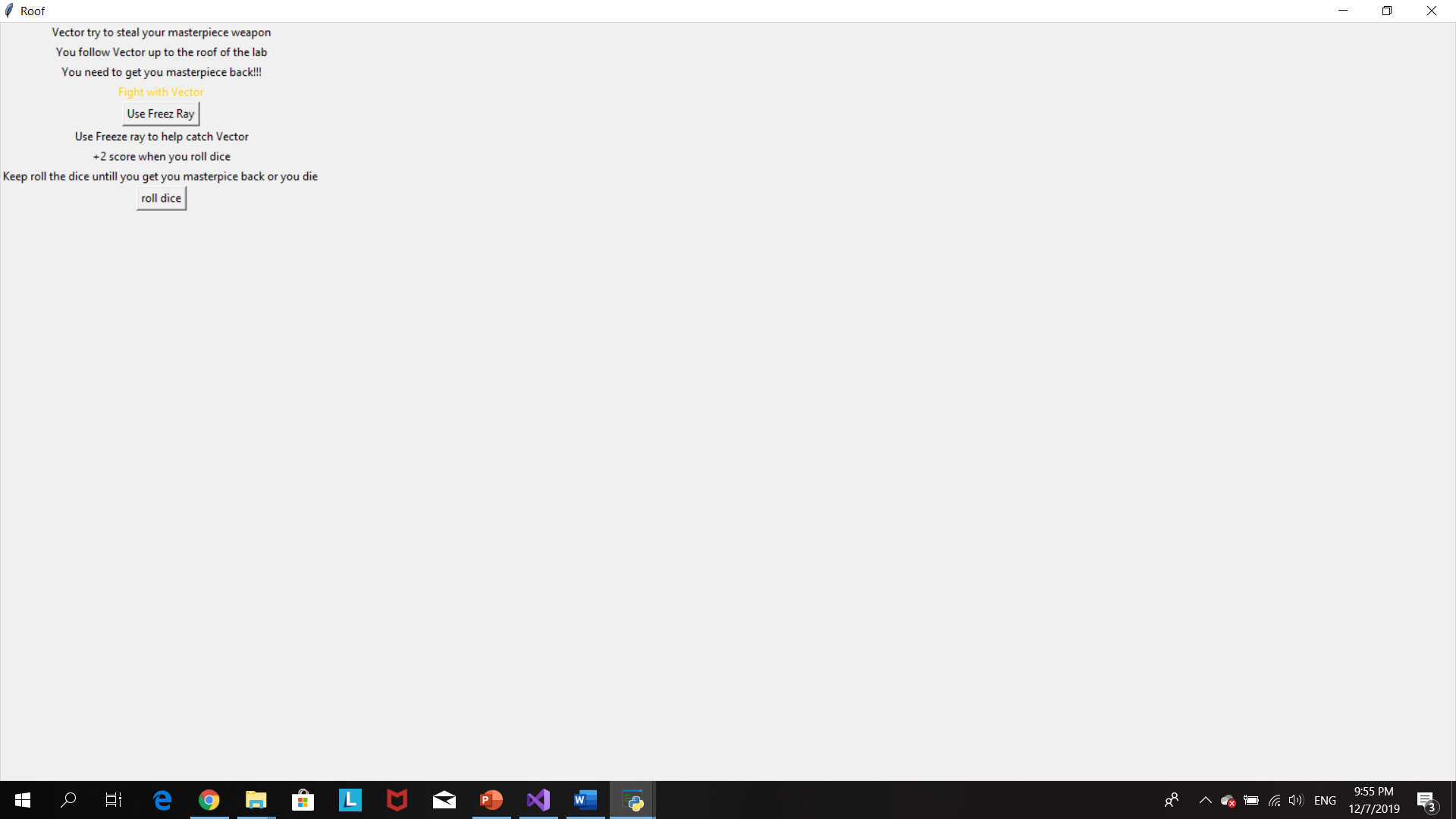
However, the player still beat Vector and have a victory



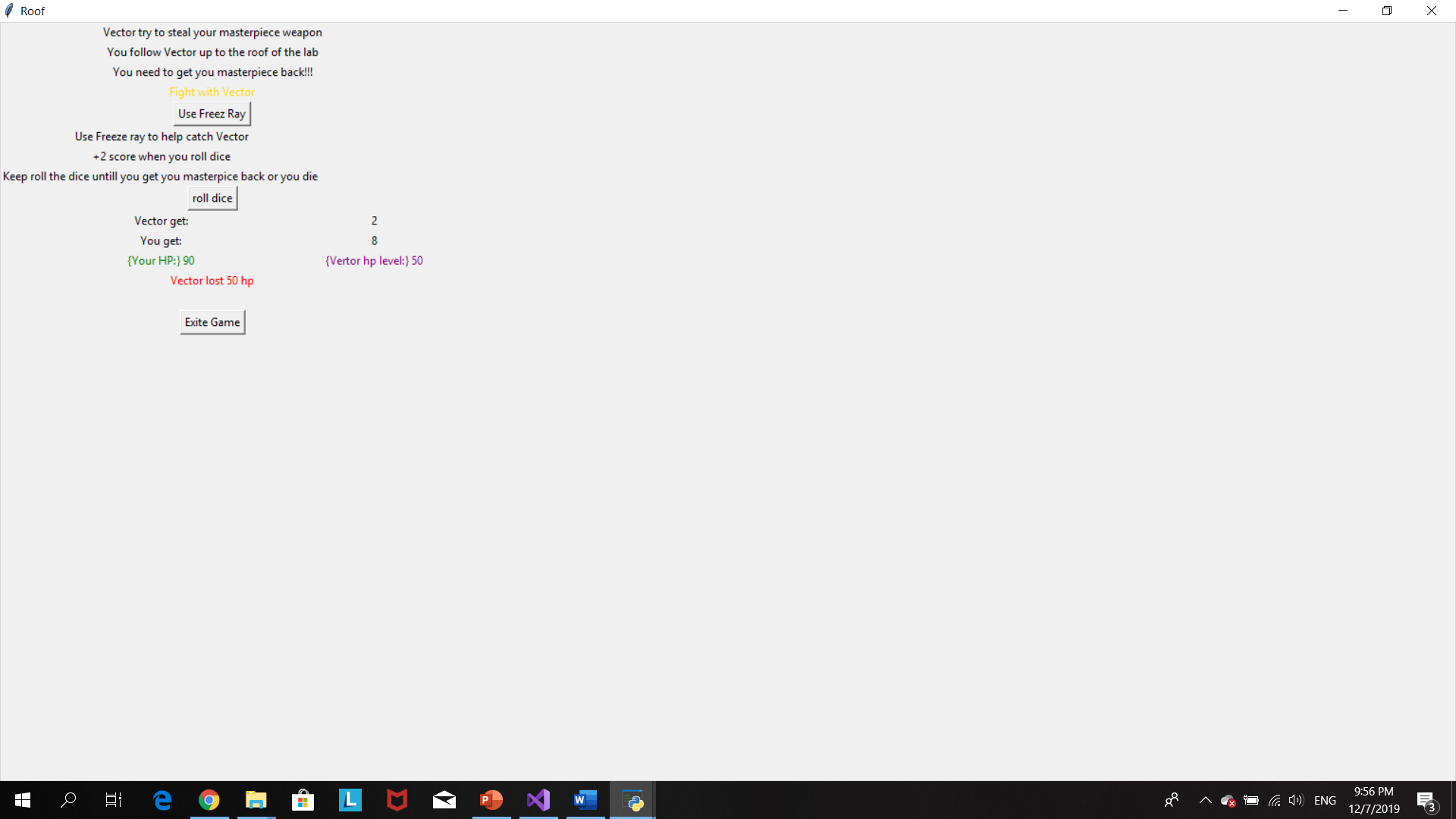
This window will pop up when user free Dr.Naferio and receive a Freeze Ray



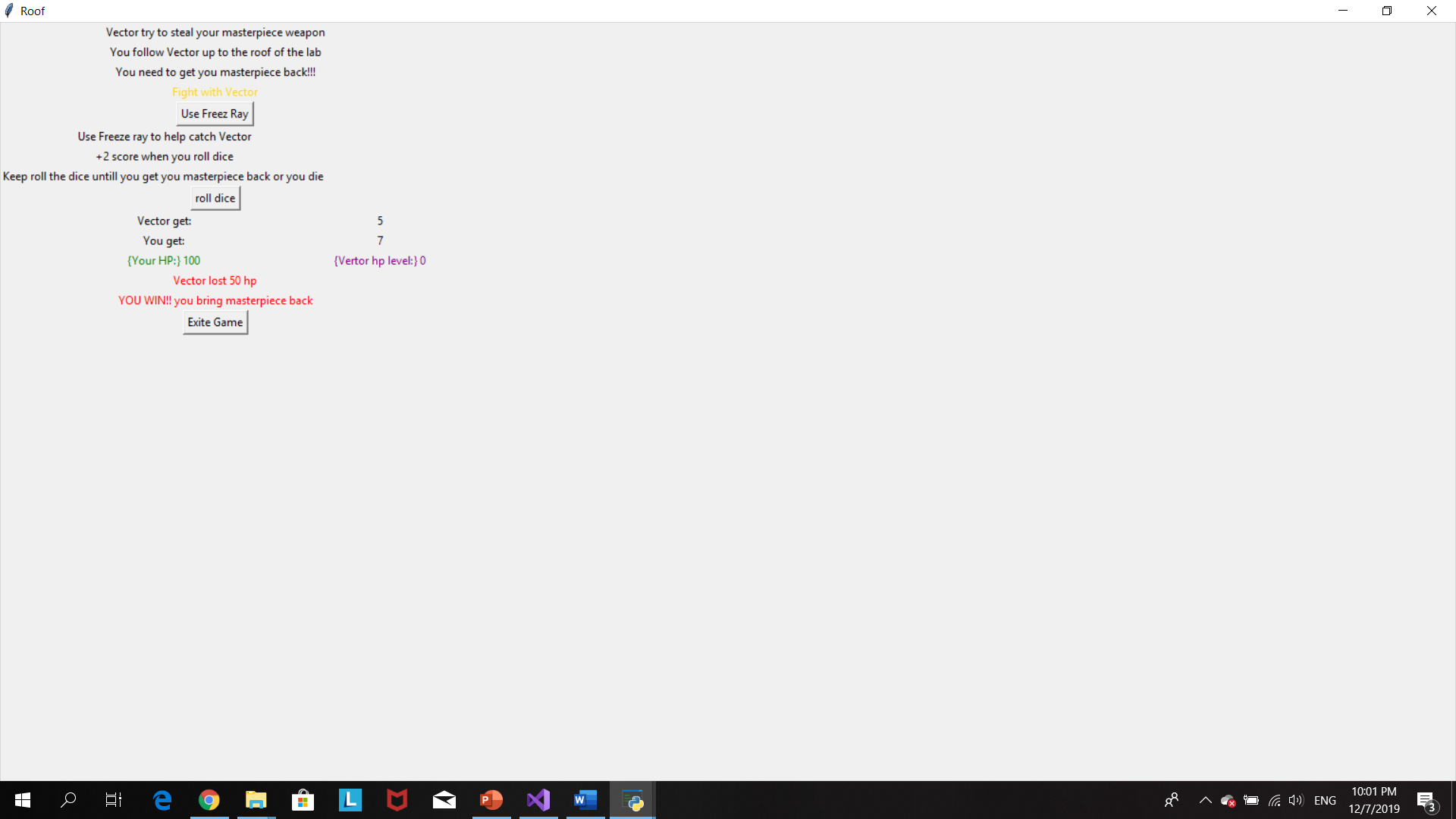
Freeze Ray +2 point fro rolling dice



Roll the dice to make an attack



Player beat Vector and get Masterpiece back



# Source Code

#Program 3: Fantasy adventure game

#The purpose of this program is to create a game inspired from the movie despicable me

#Author:Prang Kongthongluck

#Date:27/11/2019

#version:2.1

import random

from tkinter import\*

import tkinter as tk

item=[]

hplevel=100

vectorhp=100

#when player choosed Gru as his/her character

def Gruitem():

global itemlist

global item

itemlist=Listbox(root)

item.append("Turbo Car")

Label(root,text="Your HP= 100").grid(row=5,columnspan=3)

Label(root,text="Start item: Turbo car - use it to escape from anything ").grid(row=6,columnspan=3)

Button(root,text="Start Game",command=HouseGru).grid(row=7,column=1)

#window for living room scence

def HouseGru():

global house

house=Toplevel(root)

house.title("Living room")

Label(house,text="You are now in living room in Gru's house").grid(row=0,columnspan=3)

Label(house,text="You meet Gru's pet,'Kyle'").grid(row=1,columnspan=3)

Label(house,text="What will you do?").grid(row=2,columnspan=3)

Button(house,text="Turn back",command=quit).grid(row=3,column=0)

Button(house,text="fight",command=(fight)).grid(row=3,column=1)

Button(house,text="Use item",command=houseitemGru).grid(row=3,column=2)

#player use item to eascape from living room his/her start item will be removed from the item list

def houseitemGru():

Label(house,text="Gru uses his car to escape from living room").grid(row=4,columnspan=3)

item.remove("Turbo Car")

Button(house,text="continue",command=bedroom).grid(row=8,columnspan=3)

#---------------------------------------------------------------------------------------

#when player choosed Agnes as his/her character

def Agnesitem():

global item

itemlist=Listbox(root)

item.append("Unicorn")

Label(root,text="Start item: unicorn - use unicorn to trade for anything ").grid(row=4,columnspan=3)

Label(root,text="Your HP= 100").grid(row=5,columnspan=3)

Label(root,text="You have 3 lifes").grid(row=6,columnspan=3)

Button(root,text="Start Game",command=HouseAgnes).grid(row=7,column=1)

#window for living room scence

def HouseAgnes():

global house

house=Toplevel(root)

house.title("Living room")

Label(house,text="You are now in living room in Gru's house").grid(row=0,columnspan=3)

Label(house,text="You meet Gru's pet,'Kyle'").grid(row=1,columnspan=3)

Label(house,text="What will you do?").grid(row=2,columnspan=3)

Button(house,text="Turn back",command=quit).grid(row=3,column=0)

Button(house,text="fight",command=(fight)).grid(row=3,column=1)

Button(house,text="Use item",command=houseitemAgnes).grid(row=3,column=2)

#player use item to eascape from living room his/her start item will be removed from the item list

def houseitemAgnes():

Label(house,text="Agnes gives Kyle her unicorn and walk pass the living room").grid(row=4,columnspan=3)

item.remove("Unicorn")

Button(house,text="continue",command=bedroom).grid(row=8,columnspan=3)

#----------------------------------------------------------------------------------------

#when player choosed Jerry as his/her character

def Jerryitem():

global item

itemlist=Listbox(root)

item.append("Fart Gun")

Label(root,text="Start item: Fart Gun - distract attention").grid(row=4,columnspan=3)

Label(root,text="Your HP= 100").grid(row=5,columnspan=3)

Label(root,text="You have 3 lifes").grid(row=6,columnspan=3)

Button(root,text="Start Game",command=HouseJerry).grid(row=7,column=1)

#window for living room scence

def HouseJerry():

global house

house=Toplevel(root)

house.title("Living room")

Label(house,text="You are now in living room in Gru's house").grid(row=0,columnspan=3)

Label(house,text="You meet Gru's pet,'Kyle'").grid(row=1,columnspan=3)

Label(house,text="What will you do?").grid(row=2,columnspan=3)

Button(house,text="Turn back",command=quit).grid(row=3,column=0)

Button(house,text="fight",command=(fight)).grid(row=3,column=1)

Button(house,text="Use item",command=houseitemJerry).grid(row=3,column=2)

#player use item to eascape from living room his/her start item will be removed from the item list

def houseitemJerry():

Label(house,text="Jerry use Fart gun to distract Kyle and walk past the living room").grid(row=4,columnspan=3)

item.remove("Fart Gun")

Button(house,text="continue",command=bedroom).grid(row=8,columnspan=3)

#----------------------------------------------------------------------------------------

#fight with Kyle

def fight():

Button(house,text="roll the dice",command=(rollDice)).grid(row=4,columnspan=3)

#roll dice program

def rollDice():

global life

global hplevel

hplevel =100

tv\_rollKyle=StringVar()

tv\_rollplayer=StringVar()

tv\_winner=StringVar()

tv\_hp=StringVar()

#use randome for rolling dice

rollKyle = random.randint(1,6)

tv\_rollKyle.set(rollKyle)

rollplayer=random.randint(1,6)

tv\_rollplayer.set(rollplayer)

if (rollKyle>rollplayer):

winner="You lose, hp-10"

hplevel=90

heplevel=("Your HP =",hplevel)

tv\_hp.set(heplevel)

elif (rollKyle<rollplayer):

winner="You win!!, you get new item: Zapper"

item.append("Zapper")

tv\_winner.set(winner)

Label(house,text="Kyle get:").grid(row=5,column=1)

Label(house,textvariable=tv\_rollKyle).grid(row=5,column=2)

Label(house,text="You get:").grid(row=6,column=1)

Label(house,textvariable=tv\_rollplayer).grid(row=6,column=2)

Label(house,textvariable=tv\_winner,fg="red").grid(row=7,columnspan=3)

Label(house,textvariable=tv\_hp,fg="red").grid(row=8,columnspan=3)

Button(house,text="continun",command=bedroom).grid(row=9,columnspan=3)

#----------------------------------------------------------------------------------------

#bedroom scence

def bedroom():

global bed

bed=Toplevel(root)

bed.title("Bedroom")

Label(bed,text="You are now in the bedroom where you meet Margo").grid(row=0,columnspan=3)

Label(bed,text="Margo is sleeping and her phone is next to her").grid(row=1,columnspan=3)

Button(bed,text="Leave the room",anchor="e",command=Kitchen).grid(row=2,column=1)

Button(bed,text="collect Margo phone",anchor="w",command=MargoPhone).grid(row=2,column=2)

#player choose to collect Phone

def MargoPhone():

#Phone is added to the item list

item.append("Phone")

Label(bed,text="Phone is added to your item bag").grid(row=3,columnspan=3)

Button(bed,text="continue",anchor="e",command=Kitchen).grid(row=4,column=1)

#----------------------------------------------------------------------------------------

#kitchen sence

def Kitchen():

global kit

kit=Toplevel(root)

kit.title("Kitchen")

Label(kit,text="You are now in the kitchen").grid(row=0,columnspan=3)

Label(kit,text="Edith try to make herself a pancake").grid(row=1,columnspan=3)

Label(kit,text="Help Edith by use Margo's phone tocall Nanna").grid(row=2,columnspan=3)

Button(kit,text="help Edith",command=HelpEdith).grid(row=3,column=1)

Button(kit,text="dont't have phone",command=GruRoom).grid(row=3,column=2)

#display item the player have in form of listbox

def HelpEdith():

global itemlist

itemlist=Listbox(kit)

itemlist.grid(rowspan=4,column=1)

for i in item:

itemlist.insert("end",i)

Button(kit,text="Select",command=selectedItemHelpEdith).grid(rowspan =7,columnspan=3)

def selectedItemHelpEdith():

global item

#delete selected item out of th listbox

itemlist.delete(tk.ANCHOR)

#delete the selected item out from the list

try:

selection=itemlist.curselection()

value=eval(itemlist.get(selection))

ind=item.index(value)

del(item[ind])

except:

#Phone is removed out of the list

item.remove("Phone")

print("Phone is removed from the item list")

Label(kit,text="Pancake is made").grid(row=5,column=2)

Label(kit,text="Item Pancake is now added to you item bag").grid(row=5,column=2)

#Pancake is added to the item list

item.append("Pancake")

Button(kit,text="Continue",command=GruRoom).grid(row=8,columnspan=3)

#----------------------------------------------------------------------------------------

#Gru's working room scence

def GruRoom():

global gru

gru=Toplevel(root)

gru.title("Gru's office")

Label(gru,text="You are now in Gru office").grid(row=0,columnspan=3)

Label(gru,text="You meet Carl, he wants to exchange a pancake with his siren hat").grid(row=1,columnspan=3)

Button(gru,text="Exchange",command=ExchangeCarl).grid(row=2,column=2)

Button(gru,text="Don't have pancake", command=Backyard).grid(row=2,column=1)

def ExchangeCarl():

itemlist=Listbox(gru)

itemlist.grid(rowspan=3,column=1)

for i in item:

itemlist.insert("end",i)

Button(gru,text="Select",command=selectedItemExchangeCarl).grid(rowspan =6,columnspan=3)

def selectedItemExchangeCarl():

#delete item of the list when its selected by player

itemlist.delete(tk.ANCHOR)

#delete the selected item out from the list

try:

selection=itemlist.curselection()

value=eval(itemlist.get(selection))

ind=item.index(value)

del(item[ind])

except:

#pancake ir removed out of the list

item.remove("Pancake")

print("Pancake is removed from the item list")

Label(gru,text="Siren hat is added to your item bag").grid(row=4,column=2)

#Siren hat is added to the item list

item.append("Siren Hat")

Button(gru,text="Continue",command=Backyard).grid(row=20,columnspan=3)

#----------------------------------------------------------------------------------------

#backyard scence

def Backyard():

global back

back=Toplevel(root)

back.title("Backyard")

Label(back,text="You are now in the Backyard").grid(row=0,columnspan=3)

Label(back,text="Mr. McDade who lieves next door accidentally drop his spatula to your backyard ").grid(row=1,columnspan=3)

Button(back,text="return the spatular",command=returnSpatula).grid(row=2,column=1)

Button(back,text="Keep the spactular",command=keepSpatulat).grid(row=2,column=2)

def returnSpatula():

Label(back,text="Mr.McDade says Thank you").grid(row=3,columnspan=3)

Button(back,text="continue",command=Lab).grid(row=4,columnspan=3)

def keepSpatulat():

Label(back,text="Spatular is added to your bag").grid(row=3,columnspan=3)

item.append("Spatular")

Button(back,text="continue",command=Lab).grid(row=4,columnspan=3)

#----------------------------------------------------------------------------------------

#window for lab scence

def Lab():

global lab

lab=Toplevel(root)

lab.title("Minion Lab")

lab.geometry=("200\*250")

Label(lab,text="You are now in the Minion Laboratory").grid(row=0,columnspan=4)

Label(lab,text="Dr.Nefario accidently freeze himself by shooting freeze ray to himself").grid(row=1,columnspan=4)

Button(lab,text="Help him",anchor='w',command=HelpNefario).grid(row=2,column=2)

Button(lab,text="Ignor him",anchor='e',command=Roof).grid(row=2,column=1)

def HelpNefario():

global itemlist

Label(lab,text="Item in your pocket").grid(row=3,column=0)

itemlist=Listbox(lab)

itemlist.grid(row=4,column=0)

for i in item:

itemlist.insert("end",i)

Button(lab, text = "select",command=print\_HelpNafario).grid(row=6,columnspan=4)

def print\_HelpNafario():

#call a dictionary to print out the text when each item is selected

gg=Label(lab,text=aaa[itemlist.get(itemlist.curselection())]).grid(row=3,column=1)

Label(lab,text="You free Dr.Nafario",anchor='w').grid(row=4,column=1)

Label(lab,text="You receive new item: Freeze Ray").grid(row=5,column=1)

item.append("Freeze Ray")

Button(lab,text="Continue",command=RoofHelp).grid(row=7,columnspan=4)

#dictionary to display different text when each item is selected

aaa={

"Turbo Car":"Turbo Car can't melt Dr.Nafario",

"Unicorn":"Unicorn can't melt Dr.Nafario",

"Fart Gun":"Dave is anoyyed by your Fast Gun so he hits you with the Fast Gun",

"Zapper":"Zapper malt down the ice and free Dr. Nafario",

"Phone":"Turbo Car can't melt Dr.Nafario",

"Pancake":"Dave steal your pancake",

"Spatular":"Use Spatular hit the ice and free Dr.Nafario",

"Siren Hat":"Siren Hat bring all minion to help free Dr.Nafario"

}

#----------------------------------------------------------------------------------------

#this window will show up when player choose not to help Dr.Nafario

def Roof():

global roof

roof=Toplevel(root)

roof.title("Roof")

Label(roof,text="Vector try to steal your masterpiece weapon").grid(row=0,columnspan=3)

Label(roof,text="You follow Vector up to the roof of the lab").grid(row=1,columnspan=3)

Label(roof,text="You need to get you masterpiece back!!!",anchor='w').grid(row=2,columnspan=3)

Label(roof,text="Fight with Vector",anchor='w',fg="gold").grid(row=3,columnspan=3)

Button(roof,text="roll dice",command=rollDiceVector).grid(row=4,columnspan=3)

#this window will show up when the player don't have Freez ray

def rollDiceVector():

global life

global hplevel

global vectorhp

tv\_rollVector=StringVar()

tv\_rollplayer1=StringVar()

winner1=()

winner2=()

tv\_winner1=StringVar()

tv\_winner2=StringVar()

tv\_hp=StringVar()

tv\_Vhplevel=StringVar()

rollVector = random.randint(1,6)

tv\_rollVector.set(rollVector)

rollplayer1=(random.randint(1,6))

tv\_rollplayer1.set(rollplayer1)

if (rollVector>rollplayer1):

winner1="You lose, hp -30"

hplevel -=30

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

if (rollVector<rollplayer1):

winner1="Vector lost 50 hp"

vectorhp -=50

Vhplevel=("Vertor hp level:" ,(vectorhp))

HP=("Your HP:",hplevel)

if (rollVector == rollplayer1):

winner1=(" You are tie ")

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

if (vectorhp==0) or (vectorhp <=0):

winner2="YOU WIN!! you bring masterpiece back"

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

if (hplevel == 0) or (hplevel <= 0):

winner2="You lose"

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

tv\_Vhplevel.set(Vhplevel)

tv\_hp.set(HP)

tv\_winner2.set(winner2)

tv\_winner1.set(winner1)

Label(roofH,text="Vector get:").grid(row=9,column=1)

Label(roofH,textvariable=tv\_rollVector).grid(row=9,column=2)

Label(roofH,text="You get:").grid(row=10,column=1)

Label(roofH,textvariable=tv\_rollplayer1).grid(row=10,column=2)

Label(roofH,textvariable=tv\_winner1,fg="red").grid(row=12,columnspan=3)

Label(roofH,textvariable=tv\_winner2,fg="red").grid(row=13,columnspan=3)

Label(roofH,textvariable=tv\_hp,fg="green").grid(row=11,column=1)

Label(roofH,textvariable=tv\_Vhplevel,fg="purple",anchor="e").grid(row=11,column=2)

Button(roofH,text="Exite Game",command=quit).grid(row=14,columnspan=3)

#this window will show up when the player receive Freez ray by helping DR.Nafario

def RoofHelp():

global roofH

roofH=Toplevel(root)

roofH.title("Roof")

Label(roofH,text="Vector try to steal your masterpiece weapon").grid(row=0,columnspan=3)

Label(roofH,text="You follow Vector up to the roof of the lab").grid(row=1,columnspan=3)

Label(roofH,text="You need to get you masterpiece back!!!",anchor='w').grid(row=2,columnspan=3)

Label(roofH,text="Fight with Vector",anchor='w',fg="gold").grid(row=3,columnspan=3)

Button(roofH,text="Use Freez Ray",command=FreezRay).grid(row=4,columnspan=3)

def FreezRay():

Label(roofH,text="Use Freeze ray to help catch Vector",anchor='w').grid(row=5,column=1)

Label(roofH,text="+2 score when you roll dice",anchor='w').grid(row=6,column=1)

Label(roofH,text="Keep roll the dice untill you get you masterpiece back or you die ",anchor='w').grid(row=7,column=1)

Button(roofH,text="roll dice",command=rollDiceVectorH).grid(row=8,columnspan=3)

def rollDiceVectorH():

global life

global hplevel

global vectorhp

tv\_rollVector=StringVar()

tv\_rollplayer1=StringVar()

winner1=()

winner2=()

tv\_winner1=StringVar()

tv\_winner2=StringVar()

tv\_hp=StringVar()

tv\_Vhplevel=StringVar()

rollVector = random.randint(1,6)

tv\_rollVector.set(rollVector)

#+2 score for player when he use item freeze ray

rollplayer1=(random.randint(1,6))+2

tv\_rollplayer1.set(rollplayer1)

if (rollVector>rollplayer1):

winner1="You lose, hp -30"

hplevel -=30

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

if (rollVector<rollplayer1):

winner1="Vector lost 50 hp"

vectorhp -=50

Vhplevel=("Vertor hp level:" ,(vectorhp))

HP=("Your HP:",hplevel)

if (rollVector == rollplayer1):

winner1=(" You are tie ")

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

if (vectorhp==0) or (vectorhp <=0):

winner2="YOU WIN!! you bring masterpiece back"

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

if (hplevel == 0) or (hplevel <= 0):

winner2="You lose"

HP=("Your HP:",hplevel)

Vhplevel=("Vertor hp level:" ,(vectorhp))

tv\_Vhplevel.set(Vhplevel)

tv\_hp.set(HP)

tv\_winner2.set(winner2)

tv\_winner1.set(winner1)

Label(roofH,text="Vector get:").grid(row=9,column=1)

Label(roofH,textvariable=tv\_rollVector).grid(row=9,column=2)

Label(roofH,text="You get:").grid(row=10,column=1)

Label(roofH,textvariable=tv\_rollplayer1).grid(row=10,column=2)

Label(roofH,textvariable=tv\_winner1,fg="red").grid(row=12,columnspan=3)

Label(roofH,textvariable=tv\_winner2,fg="red").grid(row=13,columnspan=3)

Label(roofH,textvariable=tv\_hp,fg="green").grid(row=11,column=1)

Label(roofH,textvariable=tv\_Vhplevel,fg="purple",anchor="e").grid(row=11,column=2)

Button(roofH,text="Exite Game",command=quit).grid(row=14,columnspan=3)

#Start window

root = Tk()

root.title("Select character")

Label(root,text="Welcome to Albuquerque").grid(row=0,columnspan=3)

Label(root,text="Choose youe character",anchor='w').grid(row=1,columnspan=3)

Gru=Button(root,text="Gru",fg="red",command=Gruitem).grid(row=3,column=0)

agnes=Button(root,text="Agnes",fg="pink",command=Agnesitem).grid(row=3,column=1)

jerry=Button(root,text="Jerry",fg="yellow",command=Jerryitem).grid(row=3,column=2)

root.mainloop()

# Reference

Boris, S. (2015). *Despicable Me Source, Minions assemble!*. [online] Animationsource.org. Available at: https://www.animationsource.org/despicableme/en/

Effbot.org. (2019). *The Tkinter Label Widget*. [online] Available at: http://effbot.org/tkinterbook/label.htm

Giovanni, G. (2018). *Python and Tkinter - Listbox, how to add and delete items*. [online] Youtube.com. Available at: https://www.youtube.com/watch?v=Ea1LjAyqBs4

Gundlapalli, P. (2018). *Python: tkinter print numbers based on value selected in listbox single selection.*Youtube.com. Available at: https://www.youtube.com/watch?v=GrwoIvtKUVE

JavaTpoint. (2018). *Python Tkinter Listbox - Javatpoint*. [online] Available at: https://www.javatpoint.com/python-tkinter-listbox

Klein, B. and Mitchinson, D. (2018). *GUI Programming with Python: Buttons in Tkinter*. [online] Python-course.eu. Available at: https://www.python-course.eu/tkinter\_buttons.php

Python For Beginners. (2018). *Python Game : Rolling the dice*. [online] Available at: https://www.pythonforbeginners.com/code-snippets-source-code/game-rolling-the-dice/