



Prize Winner

**Programming, Apps &
Robotics
Year 9-10**

Linh Huynh

Glenunga International High School

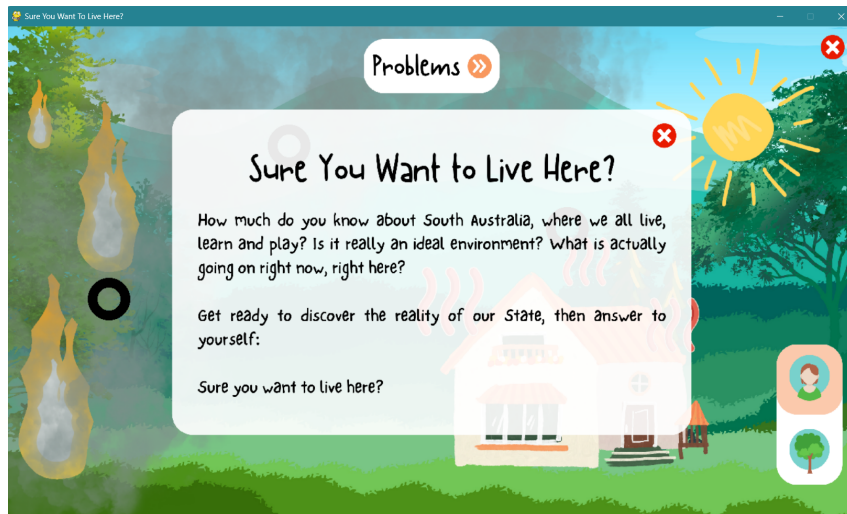


Note: I have visited online tutorials on Google to learn some to code for some parts of the program

Aim

The environment has been and is continuing to be damaged in South Australia. Therefore, I choose this as the topic for my program, calling South Australians to take responsible actions for environmental degradation, which tremendously impacts not only on nature but also the human population. The program is designed to be user-friendly and appealing-looking to engage with young audiences. Research-based facts and statistics are also used to persuade awareness, which then is followed by realistic and do-able suggestions of actions that can help improve the situation. My program could be used to spread awareness and educate South Australia's students as a potential application.

The program is coded using IDLE (Python 3.9 version) and is expected to work on similar applications. To load the entry, please download the file folders and open the program using IDLE. Then an interface should be displayed on a shell like the picture below.



Navigation



Explained Code (explanation after #)

```
import pygame
pygame.init()
```

#creating a class for displayed pictures and text

```
class dot():
```

```
    def __init__(self):
```

```
        self.color=(0,0,0)
```

```
        self.x=0
```

```
        self.y=0
```

```
        self.mx=0
```

```
        self.mx2=0
```

```
        self.my=0
```

```
        self.my2=0
```

```
        self.time=0
```

```
        self.size=65
```

```
        self.image=pygame.transform.scale(pygame.image.load("./Oliphant/bushfire.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
    def draw(self,screen):
```

```
        pygame.draw.ellipse(screen, self.color, [self.x,self.y,self.size,self.size], 15)
```

```
##### Colours #####
```

```
BLACK = ( 0, 0, 0)
```

```
WHITE = (255, 255, 255)
```

```
##### Screen Initialisation #####
```

```
SCREEN_WIDTH = 1300
```

```
SCREEN_HEIGHT = 750
```

```
size = (SCREEN_WIDTH, SCREEN_HEIGHT)
```

```
screen = pygame.display.set_mode(size)
```

```
pygame.display.set_caption("Sure You Want To Live Here?")
```

```
#Impacts on Human Mode
```

```
bg=pygame.transform.scale(pygame.image.load("./Oliphant/bg.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
bushfire=pygame.transform.scale(pygame.image.load("./Oliphant/bushfire.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
air=pygame.transform.scale(pygame.image.load("./Oliphant/air.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
heatwave=pygame.transform.scale(pygame.image.load("./Oliphant/heatwave.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
dot_image=[bushfire,air,heatwave]
```

```
dot_cor=[125,384],[400,150],[825,276]
```

```
#Impacts on Animal Mode
```

```
a_bg=pygame.transform.scale(pygame.image.load("./Oliphant/a_bg.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
a_bushfire=pygame.transform.scale(pygame.image.load("./Oliphant/b_a.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
a_specie=pygame.transform.scale(pygame.image.load("./Oliphant/s_a.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
a_heatwave=pygame.transform.scale(pygame.image.load("./Oliphant/h_a.png"),  
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
a_dot_image=[a_bushfire,a_specie,a_heatwave]
```

```
a_dot_cor=[125,384],[359,590],[825,276]
```

```
a_dot_lst=[]
```

```
#Turning each image to an object belong to the class Dot above, so they can be clicked,  
displayed, interact with user,etc.
```

```
for x, y in a_dot_cor:
    d = dot()
    d.x = x
    d.y = y
    a_dot_lst.append(d)
    d.image=a_dot_image[a_dot_lst.index(d)]

a_dot_lst[1].image2=pygame.transform.scale(pygame.image.load("./Oliphant/p_habitat.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
```

```
#Actions Mode pictures upload
bg2=pygame.transform.scale(pygame.image.load("./Oliphant/bg2.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
bf=pygame.transform.scale(pygame.image.load("./Oliphant/p_b.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
a=pygame.transform.scale(pygame.image.load("./Oliphant/p_a.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
h=pygame.transform.scale(pygame.image.load("./Oliphant/p_h.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
dot_image2=[bf,a,h]
```

```
dot_lst=[]
```

```
for x, y in dot_cor:
    d = dot()
    d.x = x
    d.y = y
    dot_lst.append(d)
    d.image=dot_image[dot_lst.index(d)]
    d.image2=dot_image2[dot_lst.index(d)]
```

```
#Uploading Instructions: starting, ending screen
start_pic=pygame.transform.scale(pygame.image.load("./Oliphant/start.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])#change
sa1=pygame.transform.scale(pygame.image.load("./Oliphant/sa1.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
sa2=pygame.transform.scale(pygame.image.load("./Oliphant/sa2.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
end=pygame.transform.scale(pygame.image.load("./Oliphant/end.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
end2=pygame.transform.scale(pygame.image.load("./Oliphant/exit.png"),
[SCREEN_WIDTH,SCREEN_HEIGHT])
instr_m=[[546,20,752,97],[982,1024,150,180],[1247,14,1280,44],[551,755,508,580]]
```

```
instr=[sa1,sa2,end,end2]
instr_lst=[]
for b,c,d,e in instr_m:
    a=dot()
    a.mx=b
    a.mx2=d
    a.my=c
    a.my2=e
    instr_lst.append(a)
    a.image=instr[instr_lst.index(a)]
```

```
done = False
clock = pygame.time.Clock()
```

```
start="yes"
dot_image_dis="no"
a_dot_image_dis="no"
instr_image_dis="no"
screen_dis=1
instr_dis=[0]
```

```
mode="human"
```

```
##### Main Program Loop #####
```

```
while not done:
```

```
    screen.fill(BLACK)
```

```
    while screen_dis==1:
```

```
        if mode=="human":
```

```
            #set up background for human mode
```

```
            screen.blit(bg,[0,0])
```

```
            screen.blit(end2,[0,0])
```

```
            #display objects on screen
```

```
            for i in dot_lst:
```

```
                i.draw(screen)
```

```
            #display the text if user click into it
```

```
            while dot_image_dis=="yes":
```

```
                screen.blit(dot_dis[0].image,(0,0))
```

```
                break
```

```
#similar to code for impacts on human mode, just using different pictures
if mode=="animal":
```

```
    screen.blit(a_bg,[0,0])
```

```
    screen.blit(end2,[0,0])
```

```
    for i in a_dot_lst:
```

```
        i.draw(screen)
```

```
    while a_dot_image_dis=="yes":
```

```
        screen.blit(a_dot_dis[0].image,(0,0))
```

```
        break
```

```
    #break
```

```
    while instr_image_dis=="yes" and instr_dis[0].time<1:
```

```
        screen.blit(instr_dis[0].image,(0,0))
```

```
        break
```

```
    if start=="yes":
```

```
        screen.blit(start_pic,[0,0])
```

```
    break
```

```
#switch screen to Actions mode
```

```
    while screen_dis==-1:
```

```
        screen.blit(bg2,[0,0])
```

```
        screen.blit(end2,[0,0])
```

```
        dot_lst.append(a_dot_lst[1])
```

```
    for i in dot_lst:
```

```
        i.draw(screen)
```

```
# display dots (where user click to see more information)
```

```
    while dot_image_dis=="yes":
```

```
        screen.blit(dot_dis[0].image2,(0,0))
```

```
        break
```

```
    #display instructions
```

```
    while instr_image_dis=="yes" and instr_dis[0].time<1:
```

```
        screen.blit(instr_dis[0].image,(0,0))
```

```
        break
```

```
    break
```

```

##### Events Loop #####
for event in pygame.event.get():
    pressed_pos=pygame.mouse.get_pos()

    if event.type == pygame.QUIT:
        done = True

    if event.type == pygame.MOUSEBUTTONDOWN:
        print ("x = {}, y = {}".format(pos[0], pos[1]))
#if user clicks red x button then change start to "no" to display a text and close
    if start=="yes":
        if 987<= pressed_pos[0] < 1024 and 150<= pressed_pos[1] <180:
            start="no"
            #display text/pictures if user change modes
        for i in instr_lst:
            if i.mx <= pressed_pos[0] < i.mx2 and i.my <= pressed_pos[1] < i.my2:
                instr_image_dis="yes"
                instr_dis=[i]

    if instr_dis[0] == instr_lst[0]:
        if 987<= pressed_pos[0] < 1024 and 150<= pressed_pos[1] <180:
            instr_image_dis="no"
            instr_dis=[instr_lst[1]]
            instr_image_dis="yes"
            pressed_pos=[0,0]

    if instr_dis[0] == instr_lst[1]:
        if 987<= pressed_pos[0] < 1024 and 150<= pressed_pos[1] <180:
            instr_image_dis="no"
            instr_lst[0].time=1
            instr_lst[1].time=1

    if instr_dis[0] == instr_lst[2]:
        if 551<= pressed_pos[0] < 755 and 508<= pressed_pos[1] <580:
            done = True

#switch between impacts on human/nature mode
    if 1192<= pressed_pos[0] < 1261 and 505<= pressed_pos[1] <570:
        mode="human"
    if 1192<= pressed_pos[0] < 1261 and 618<= pressed_pos[1] <678:
        mode="animal"

```



```

#display picture/text if user click on dot
if mode=="human":
    for i in dot_lst:
        if i.x <= pressed_pos[0] < i.x + i.size and i.y <= pressed_pos[1] < i.y + i.size:
            dot_image_dis="yes"
            dot_dis=[i]
#close if user click x
    if 987<= pressed_pos[0] < 1024 and 150<= pressed_pos[1] <180:
        dot_image_dis="no"
        dot_dis=[]
#switch modes
    if 548<= pressed_pos[0] < 756 and 18<= pressed_pos[1] <102:
        screen_dis=screen_dis*-1

if mode=="animal":
    for i in a_dot_lst:
        if i.x <= pressed_pos[0] < i.x + i.size and i.y <= pressed_pos[1] < i.y + i.size:
            a_dot_image_dis="yes"
            a_dot_dis=[i]
    if 987<= pressed_pos[0] < 1024 and 150<= pressed_pos[1] <180:
        a_dot_image_dis="no"
        a_dot_dis=[]
    if 548<= pressed_pos[0] < 756 and 18<= pressed_pos[1] <102:
        screen_dis=screen_dis*-1

if event.type == pygame.MOUSEMOTION:
    if mode=="human":
        pos = pygame.mouse.get_pos()
#just to let user know they are hovering over a clickable button
        for i in dot_lst:
            if i.x <= pos[0] < i.x + i.size and i.y <= pos[1] < i.y + i.size:
                i.color=(255,255,255)
            else:
                i.color=(0,0,0)

if mode=="animal":
    for i in a_dot_lst:
        if i.x <= pos[0] < i.x + i.size and i.y <= pos[1] < i.y + i.size:
            i.color=(255,255,255)
        else:
            i.color=(0,0,0)

```

```
pygame.display.flip()
```

```
clock.tick(60)
```

```
pygame.quit()
```

Bibliography

Information

- =South%20Australia%20is%20a%20national,South%20Australia's%20Recycling%20Act ivity%20reports.
- <https://7news.com.au/weather/australia-claims-14-of-the-15-hottest-places-in-the-world-as-heatwave-continues-c-614224>
- <https://byjus.com/biology/ozone-layer-depletion/>
- https://cdn.environment.sa.gov.au/environment/docs/nsl_overview.pdf
- <https://cdn.environment.sa.gov.au/environment/docs/South-Australia-Responding-to-Climate-Change.pdf>
- https://ehq-production-australia.s3.ap-southeast-2.amazonaws.com/011d1e1c681f3689e48085549aa7593e95a32860/original/1645063358/df36431249e37fd36587790271bd8c21_2022_0215_Air_Quality_-_information_for_Marino_residents.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA4KKNQAKIOR7VAOP4%2F20230629%2Fap-southeast-2%2Fs3%2Faws4_request&X-Amz-Date=20230629T051249Z&X-Amz-Expires=300&X-Amz-SignedHeaders=host&X-Amz-Signature=8c5806f077e140f7082831331c23dced52d11bd9b36b368ce94fbc7166393f65
- <https://naturaldisaster.royalcommission.gov.au/system/files/2020-07/FEC.360.001.0001.pdf>
- <https://southaustralia.com/plan-a-trip/weather-and-alerts>
- <https://southaustralia.com/plan-a-trip/weather-and-alerts#:~:text=SUMMER,high%20as%2045%C2%B0C.>
- <https://weatherspark.com/countries/AU/05>
- <https://www.adelaidenow.com.au/news/south-australia/thousands-of-adelaide-homes-lose-power-on-day-of-recordbreaking-heat/news-story/725d6f49b372d2e576a25bf477372684>
- <https://www.bbc.com/news/world-australia-30706391>
- <https://www.cancersa.org.au/donate-to-a-specific-cause/skin-cancer-fund/>
- <https://www.eea.europa.eu/en/topics/in-depth/air-pollution/eow-it-affects-our-health#:~:text=Both%20short%2D%20and%20long%2Dterm,asthma%20and%20lower%20respiratory%20infections.>
- <https://www.environment.sa.gov.au/topics/fire-management/managing-bushfire-risk>

- <https://www.environment.sa.gov.au/topics/plants-and-animals/threatened-species-and-ecological-communities/threatened-species/threatened-species-in-sa>
- <https://www.environment.sa.gov.au/topics/plants-and-animals/threatened-species-and-ecological-communities/threatened-species/threatened-species-in-sa#:~:text=At%20the%20state%20level%2C%20more,threatened%20on%20a%20regional%20level.>
- https://www.epa.sa.gov.au/community/waste_and_recycling#:~:text=
- [https://www.epa.sa.gov.au/soe_2013/main/biodi-3-what-are-the-pressures.html#:~:text=Soil%20erosion%20is%20the%20highest,Soil%20and%20Land%20Program%202007a b\).](https://www.epa.sa.gov.au/soe_2013/main/biodi-3-what-are-the-pressures.html#:~:text=Soil%20erosion%20is%20the%20highest,Soil%20and%20Land%20Program%202007a b).)
- https://www.epa.sa.gov.au/soe_2013/main/clima-2-what-do-we-know-about.html
- <https://www.epa.sa.gov.au/soe-2018/introduction/setting-the-scene>
- <https://www.extremestorms.com.au/last-major-hot-spell-and-inland-heatwave-for-spring-summer-2022-2023-nsw-16-to-the-20-march-2023/>
- <https://www.insureandgo.com.au/travel-hub/hottest-places/#:~:text=The%20record%20for%20the%20highest,of%20Oodnadatta%20in%20South%20Australia.>
- <https://www.safa.sa.gov.au/environmental-s-governance>
- <https://www.safa.sa.gov.au/environmental-s-governance#:~:text=South%20Australia%20has%20a%20statewide,targets%20to%20reduce%20greenhouse%20emissions.>
- <https://www.sahealth.sa.gov.au/wps/wcm/connect/public+content/sa+health+internet/healthy+living/protecting+your+health/environmental+health/healthy+in+the+heat/what+is+a+heatwave>
- <https://www.sydney.edu.au/news-opinion/news/2020/12/07/3-billion-animals-impacted-by-fires-including-60000-koalas-repor.html>
- [https://www.youtube.com/watch?v=I7STZsY_-Ps.](https://www.youtube.com/watch?v=I7STZsY_-Ps)

Code Tutorials

https://www.tutorialspoint.com/pygame/pygame_mouse_events.htm#:~:text=To%20obtain%20the%20coordinates%20of,function%20associated%20with%20event%20object