

Maximiliano Soberano Ramón

A01733902



**Tecnológico
de Monterrey**

**Pensamiento computacional
para la ingeniería (Gpo 6)**

1er Semestre

**Título del proyecto:
Figuras y Colores**

Bienvenido
a
Figuras y Colores



Introducción:

Este juego pretende enseñarles a niños de 6 a 14 años (nivel de primaria a secundaria) operaciones aritméticas, colores y figuras geométricas todo al mismo tiempo a través de la integración de problemas que combinen todos estos elementos, a fin de poder crear una representación visual clara y divertida de estos temas. Los temas incluyen: vértices, perímetro, área, incógnitas, combinación de colores, geometría, entre otros.

El programa mostrará en pantalla un enunciado seguido de un gráfico que lo represente, y los niños resolverán el problema y lo contestarán acto seguido. Todas las preguntas abiertas, pero algunas tienen un par de respuestas correctas.

Para los niveles fáciles, los niños usarán operadores aritméticos junto con el nombre de figuras y colores para llegar a un resultado, por ejemplo:

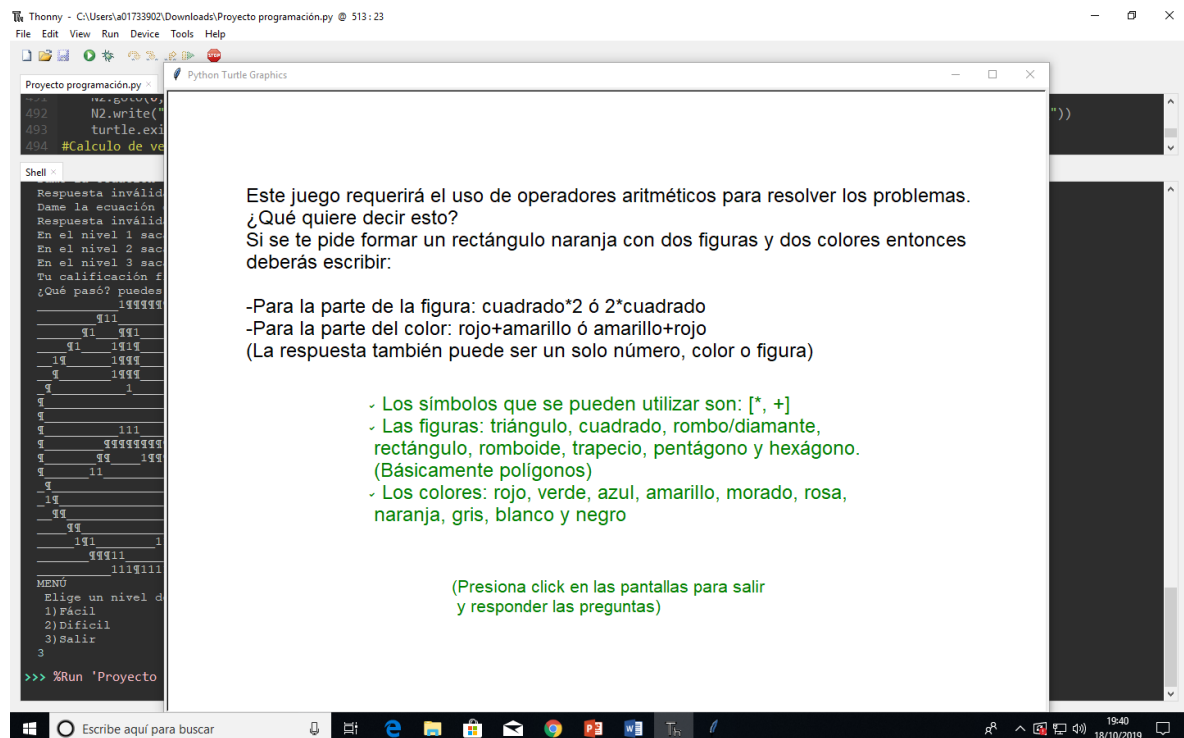
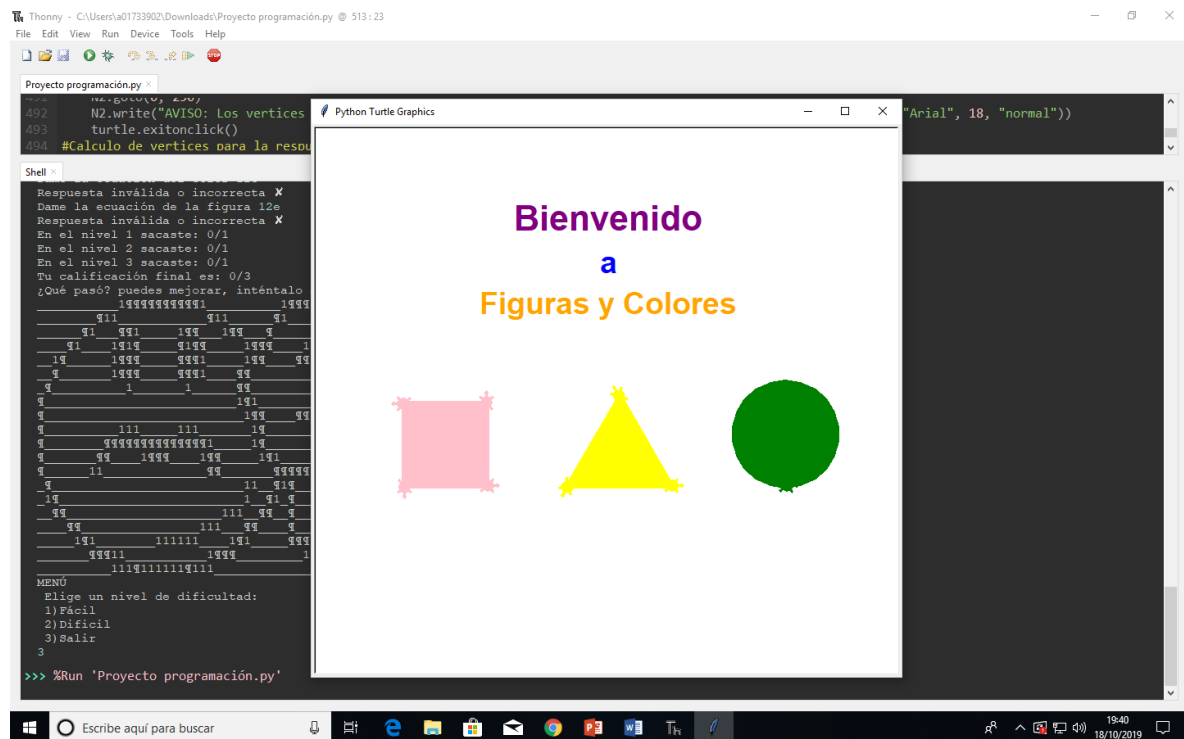
Si un problema te muestra un hexágono naranja y te pide la formula de su color y del polígono, pero solo con triángulos entonces la respuesta será. Para el color: rojo+amarillo, y para el polígono: triangulo*6

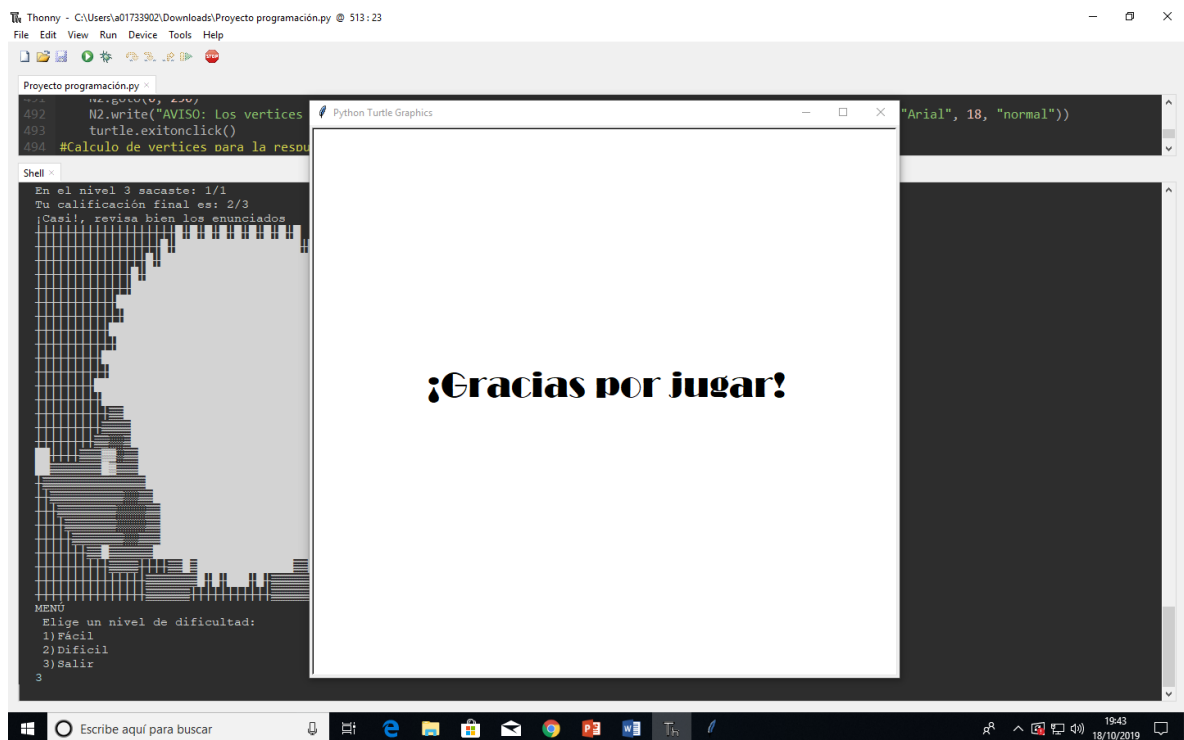
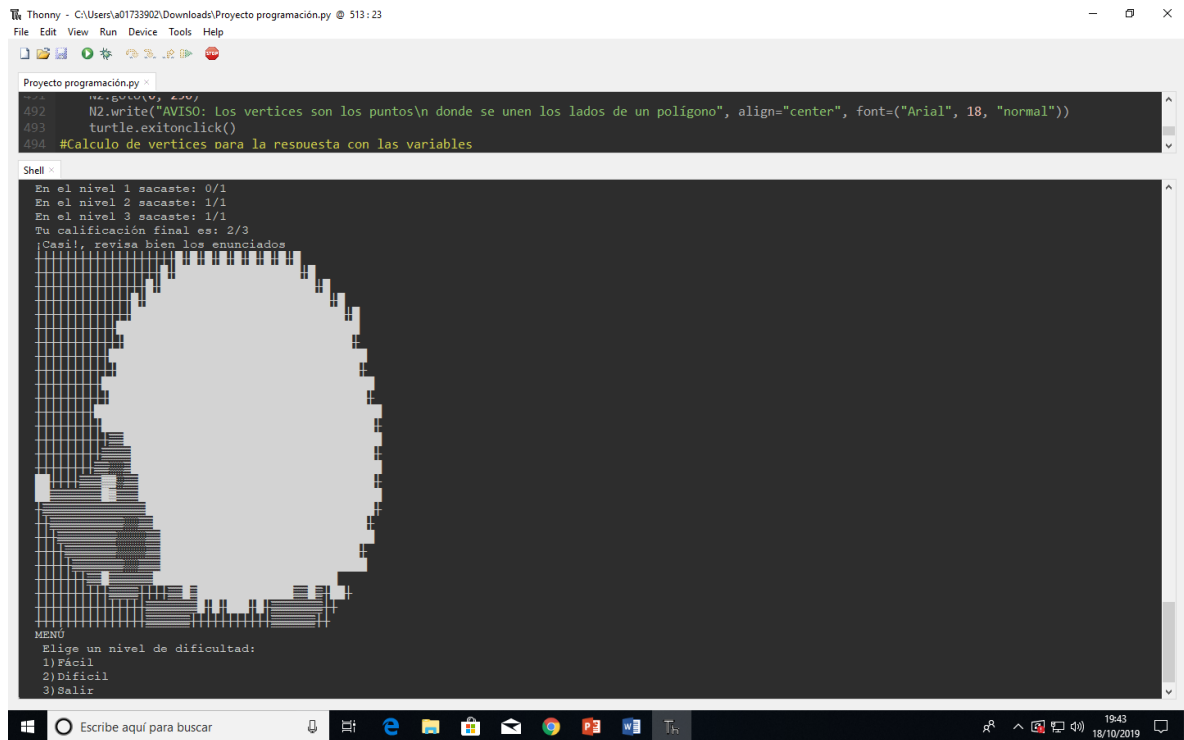
Para los niveles difíciles, las preguntas irán desde problemas de perímetro con incógnitas, hasta visualización de figuras tridimensionales, partiendo de una imagen bidimensional. Estas respuestas solo requerirán ingresar un número, pero el procedimiento implica el uso de matemáticas y razonamiento más avanzado.

El programa hace uso de la librería de tortugas para representar en pantalla lo que el problema quiere decir, así como de retroalimentación para incitarlos a seguir mejorando.

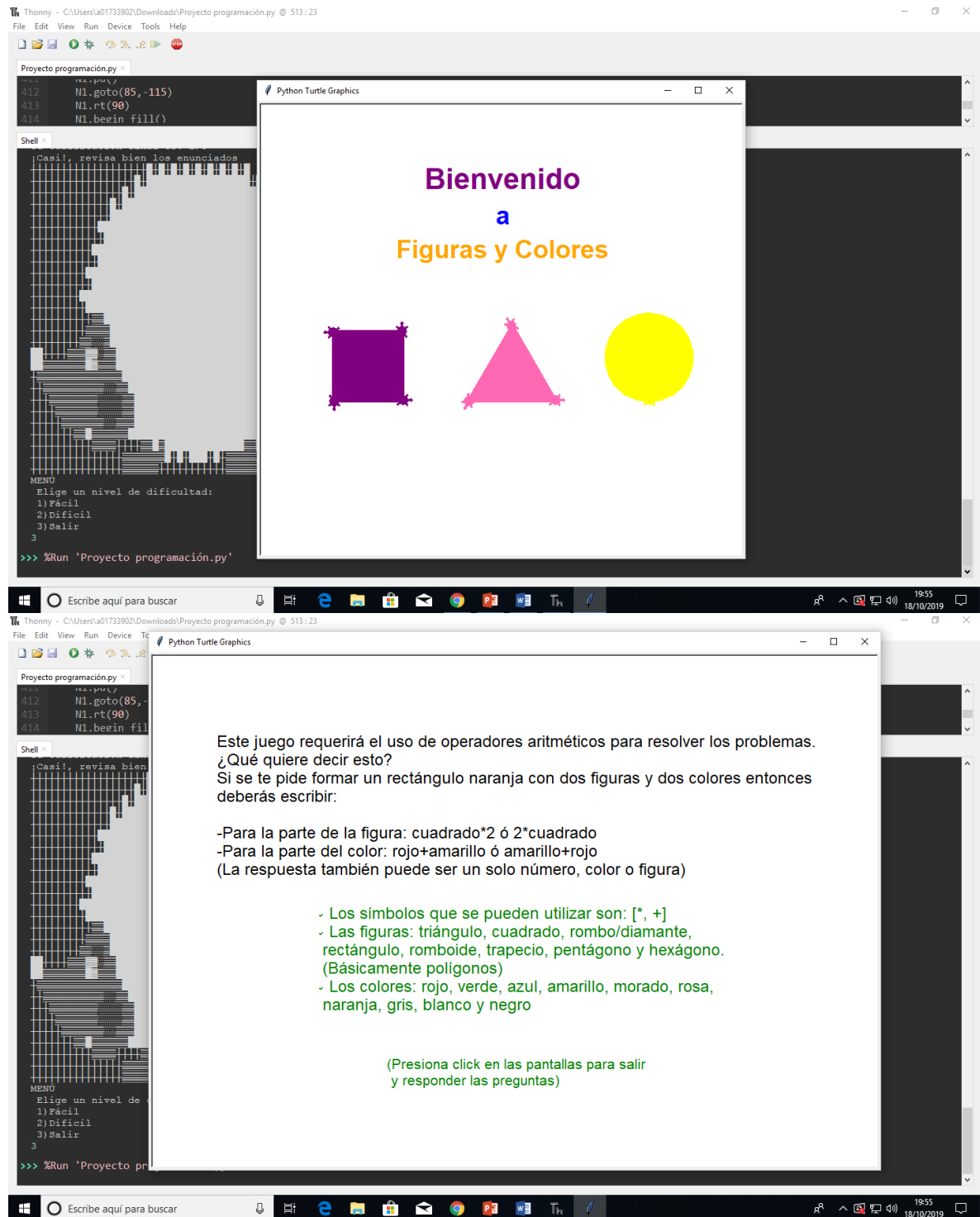
Impresión de pantalla de intentos:

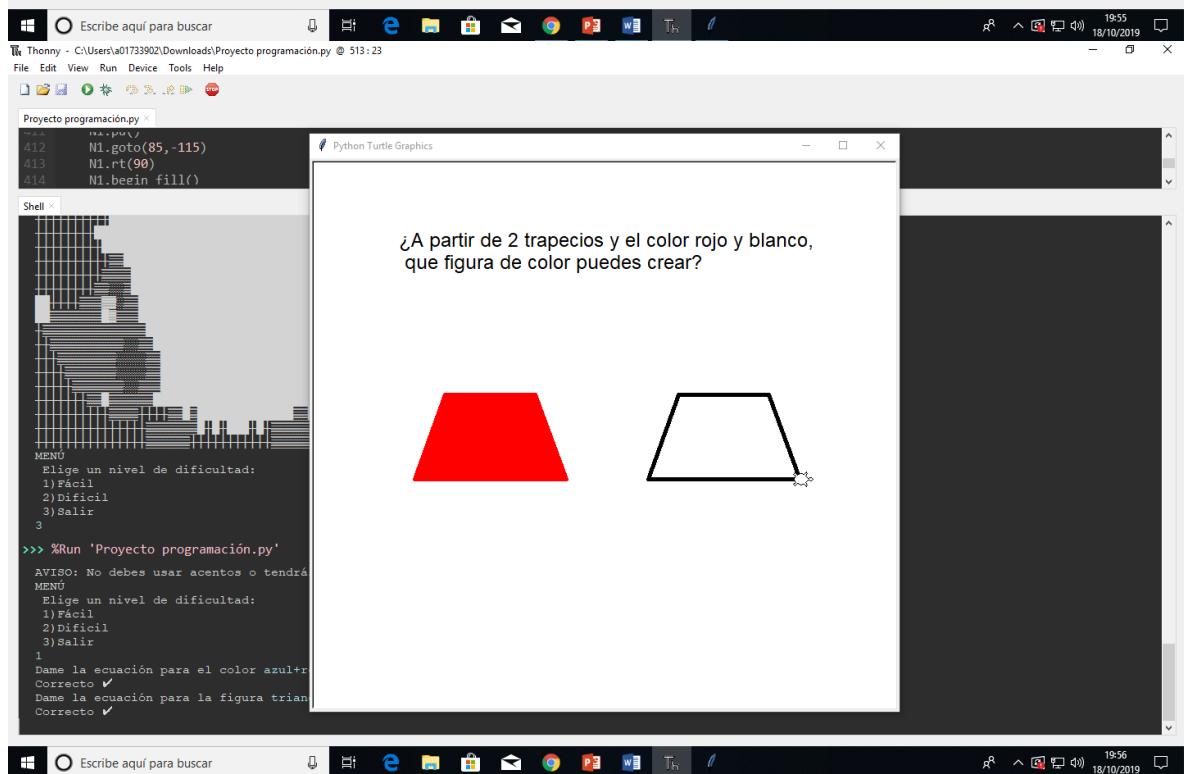
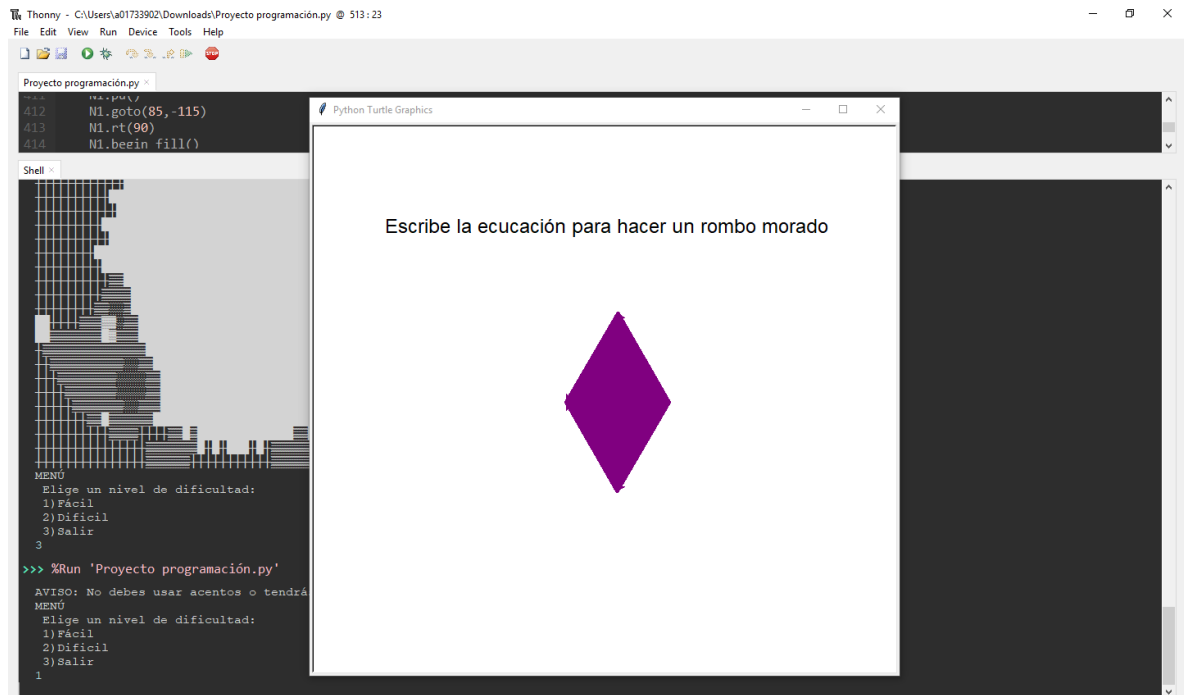
1.-

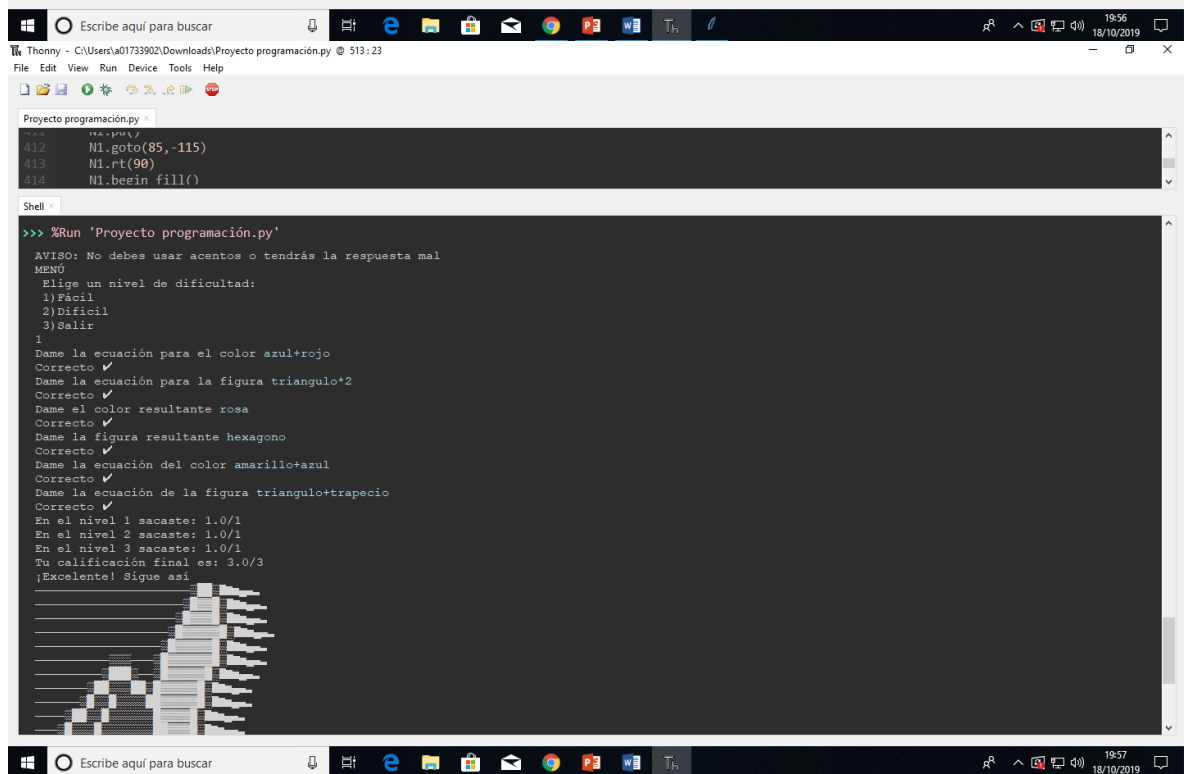
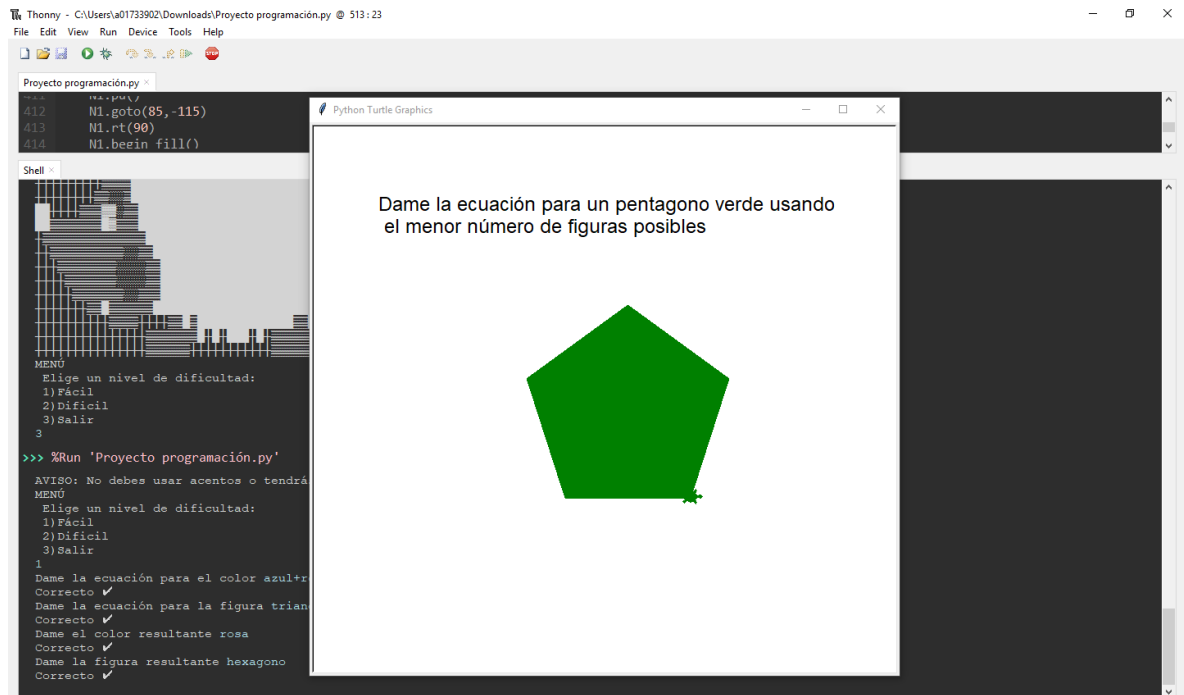


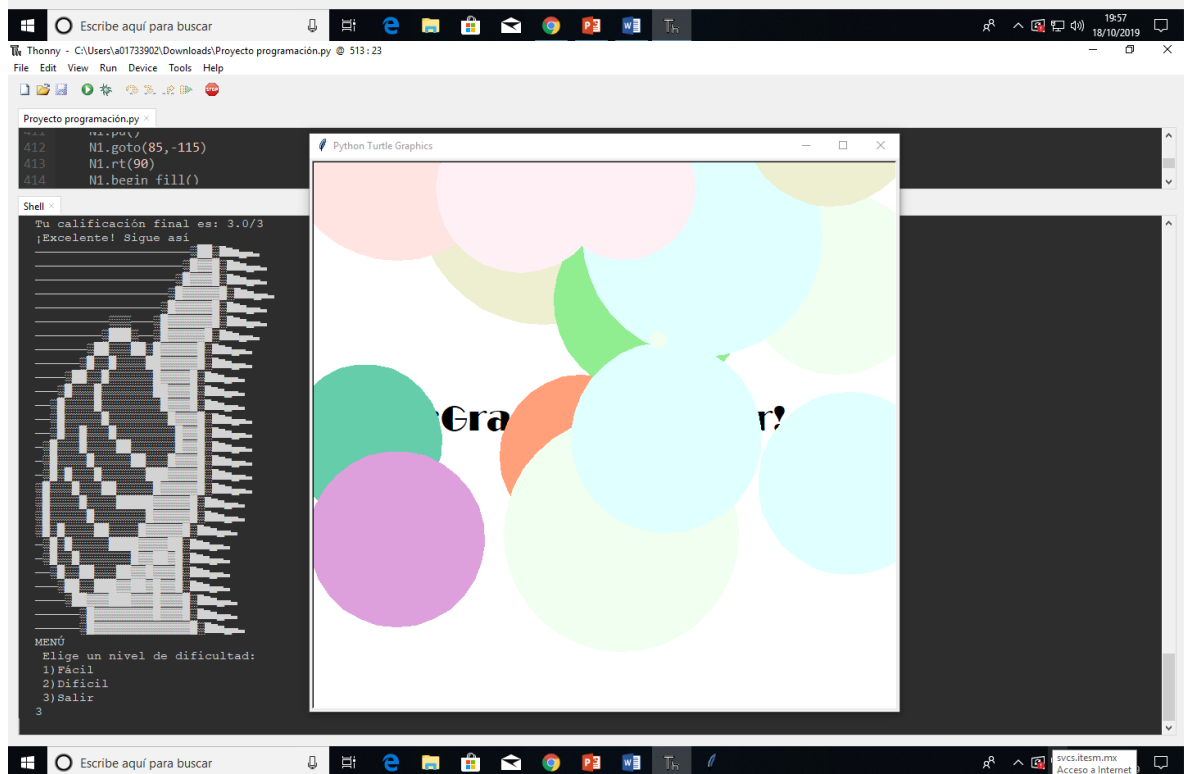
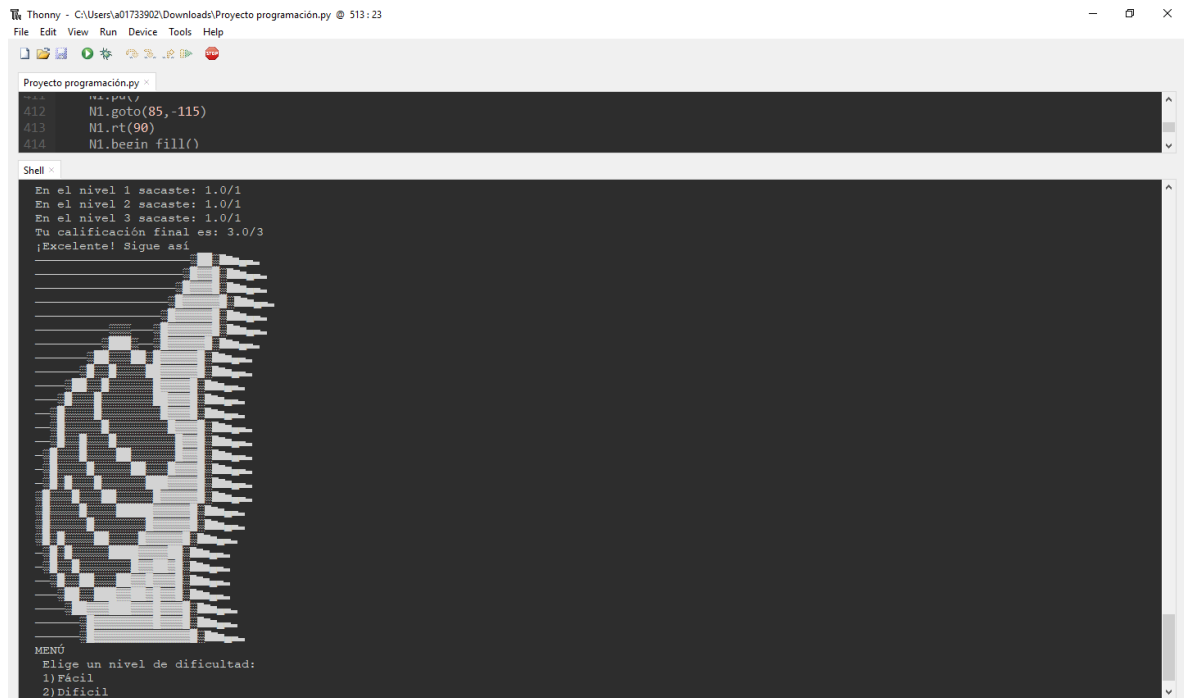


2.-

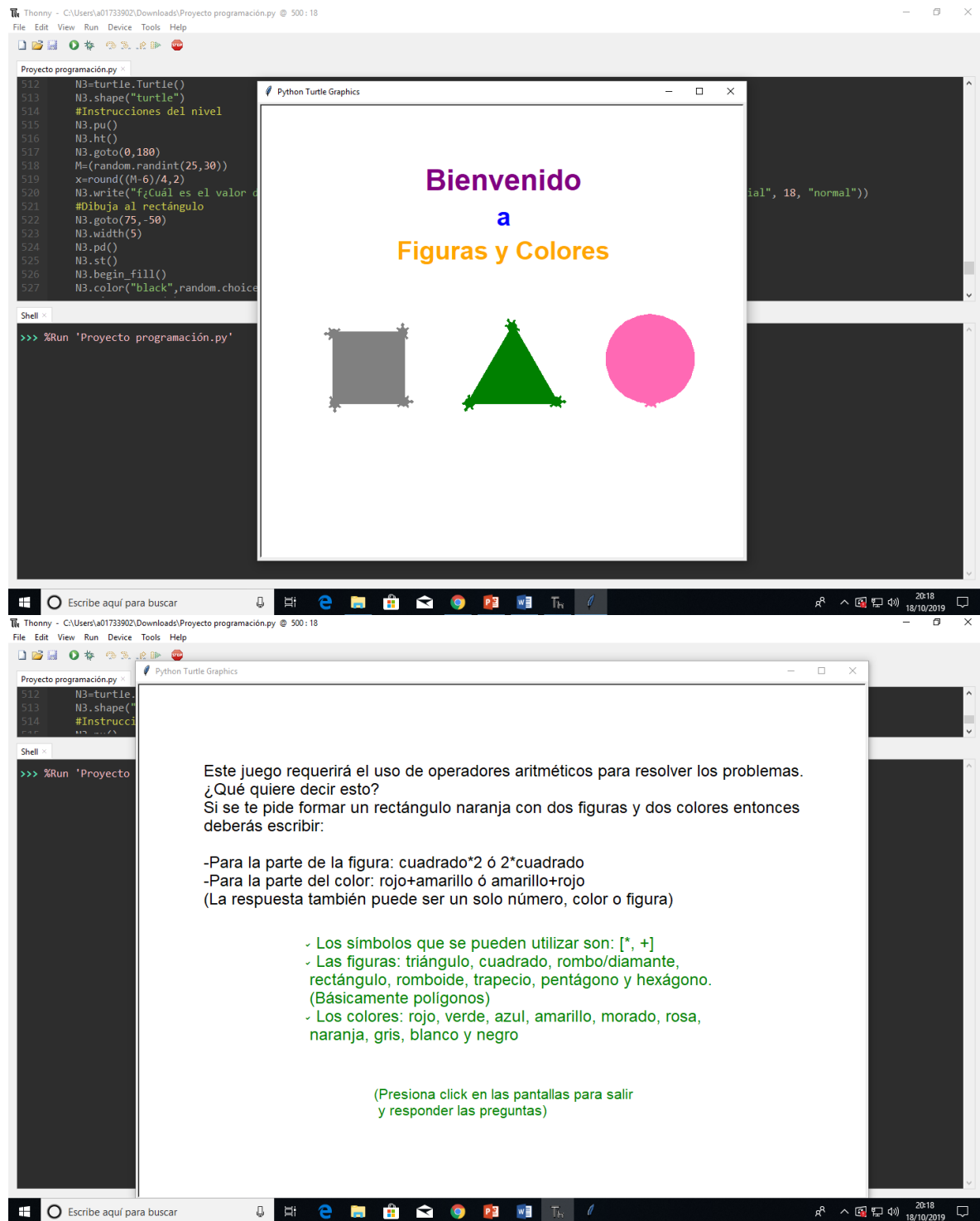


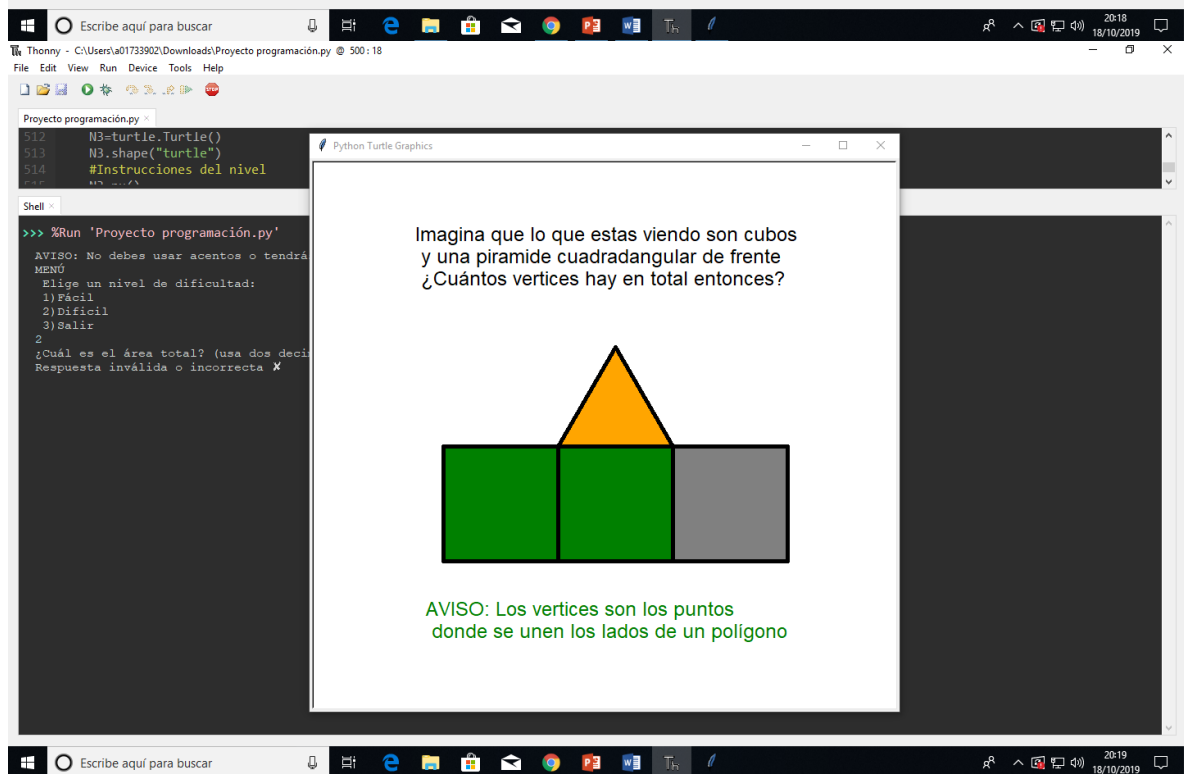
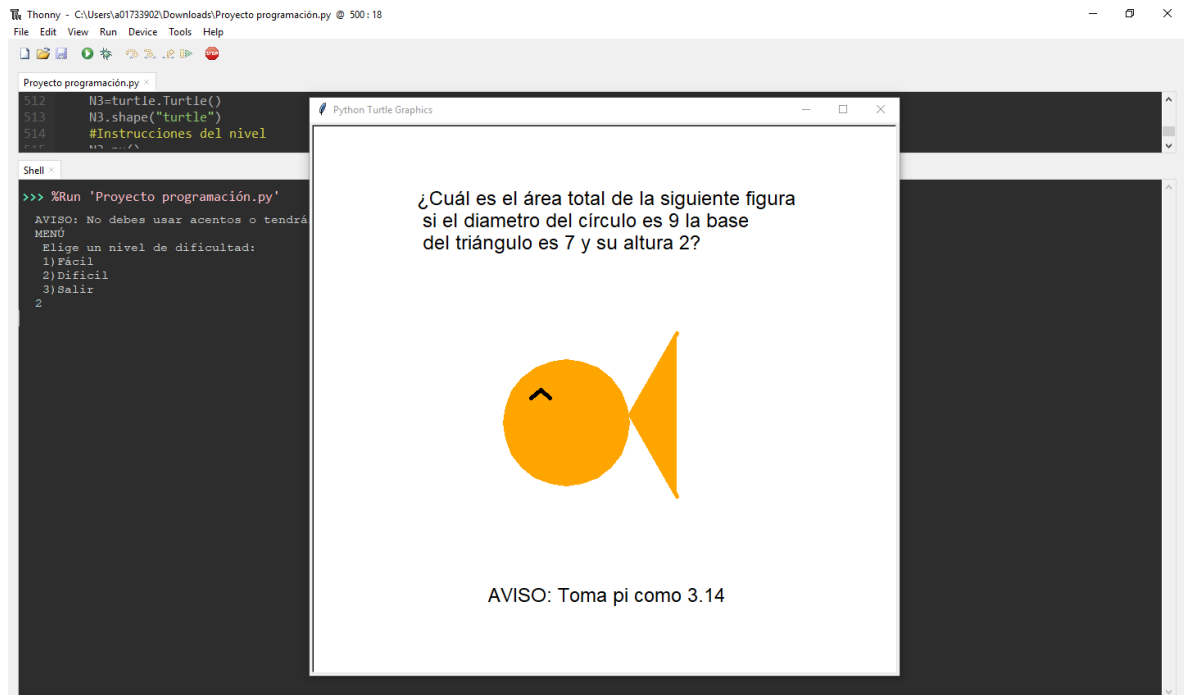


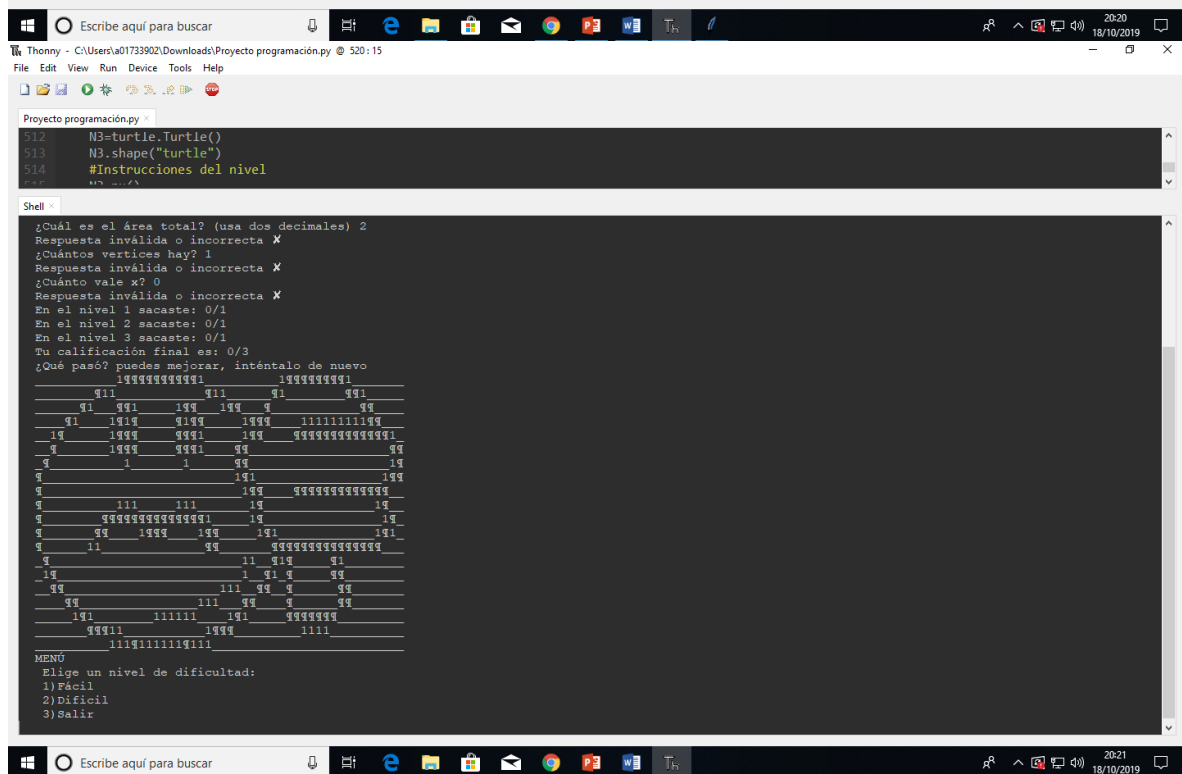
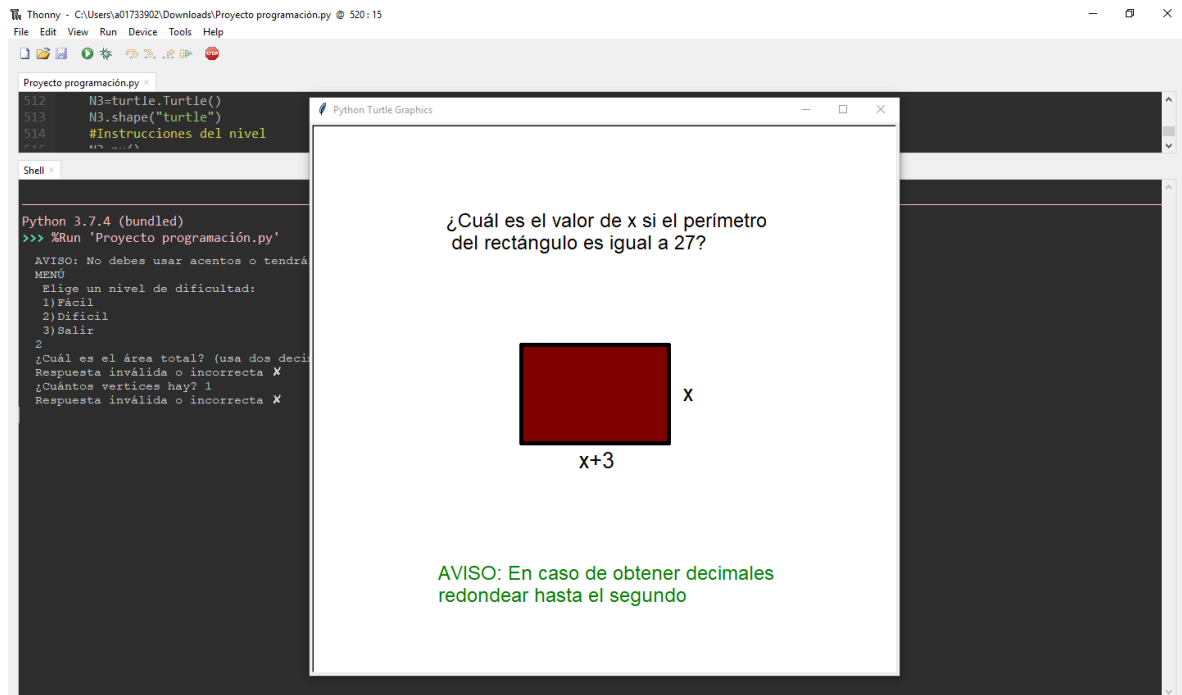


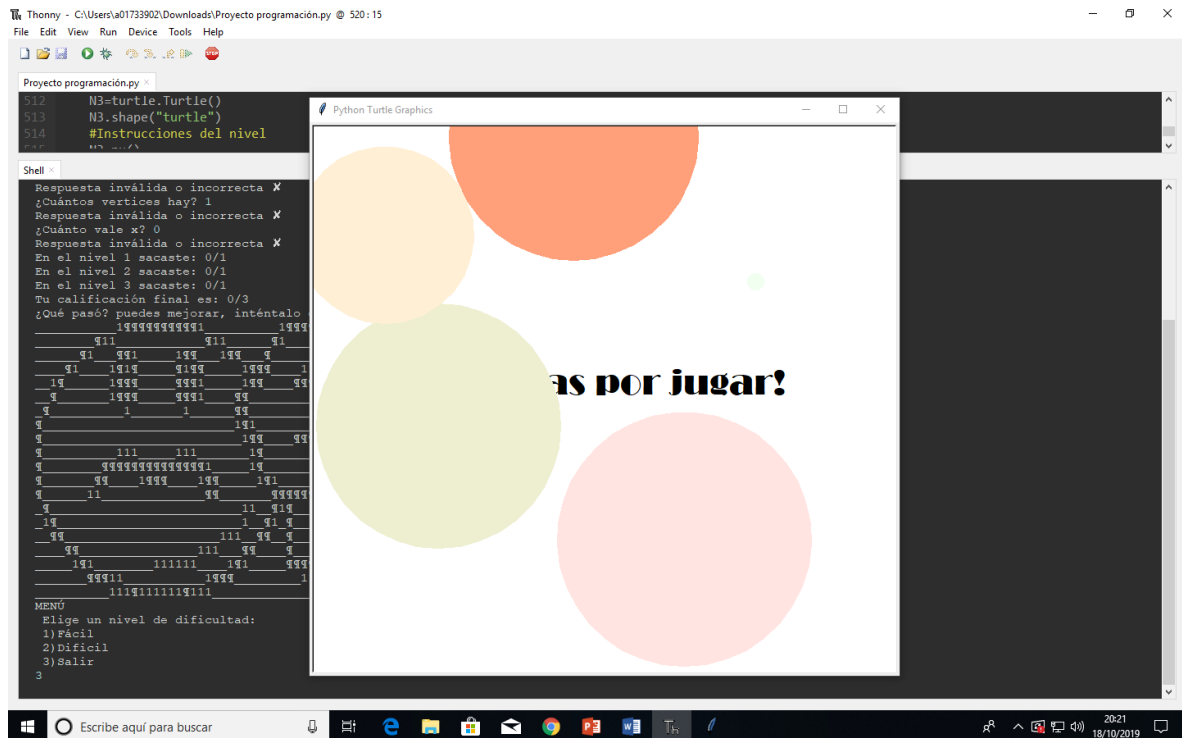


3.-









Código fuente del programa comentado

Primero, agrego las librerías que voy a usar:

```
import turtle
import random
from turtle import *
import time
import math
```

Estas dos líneas "resucitan" el ambiente de las tortugas después de que una ventana se cierra.

Funcionan reseteando dos banderas que impiden que un objeto de pantalla se cree de nuevo después de un `turtle.bye()`. Las utilizaré cada vez que tenga que abrir una ventana de tortuga de nuevo

```
#turtle.Turtle._screen = None: fuerza la creación de un objeto de pantalla Singleton
#turtle.TurtleScreen._RUNNING = True: Solo corre cuando se define un TurtleScreen()
```

```
#####
```

Aquí defino variables de colores y empiezo a escribir la trayectoria de las tortugas para hacer las figuras de la pantalla de inicio

```
#Metodos para las variables en las tortugas
```

```
A="Arial"
```

```
colors=["blue","red","purple","pink","yellow","orange","green","cyan","grey","hot pink","maroon"]
```

#Dibuja el titulo del juego en la pantalla

```
title=turtle.Turtle()
title.ht()
title.pu()
title.shape("turtle")
title.color("purple")
title.goto(0,200)
title.write("Bienvenido", align="center", font=(A, 32, "bold"))
title.color("blue")
title.goto(0,150)
title.write("a", align="center", font=(A, 28, "bold"))
title.color("orange")
title.goto(0,100)
title.write("Figuras y Colores", align="center", font=(A, 28, "bold"))
```

#Dibuja figuras abajo del titulo, empieza por el cuadrado hecho con un for

```
fig=turtle.Turtle(shape="turtle")
fig.pu()
fig.goto(-250,-100)
fig.width(7)
fig.color(random.choice(colors))
fig.begin_fill()
fig.pd()
for i in range(4):
    fig.forward(100)
    fig.stamp()
    fig.left(90)
fig.end_fill()
fig.pu()
```

#Termina el cuadrado y comienza el triángulo

```
fig.goto(75,-100)
fig.pd()
fig.color(random.choice(colors))
fig.begin_fill()
for i in range(3):
    fig.stamp()
    fig.left(120)
    fig.forward(125)
fig.end_fill()
```

#Termina el triángulo y hace un círculo

```
fig.pu()
fig.goto(215,-100)
fig.color(random.choice(colors))
fig.pd()
fig.begin_fill()
fig.circle(63)
fig.end_fill()
time.sleep(2)
```

```

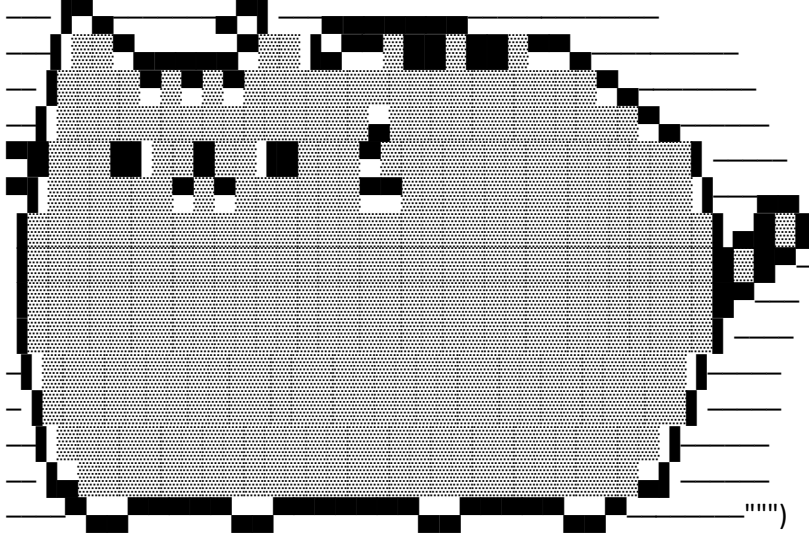
turtle.bye()
#Termina la pantalla de presentación
#####
#Defino la función para la pantalla de despedida
def despedida():
    turtle.Turtle._screen = None
    turtle.TurtleScreen._RUNNING = True
    Des=turtle.Turtle()
    Des.shape("turtle")
#Escribe mensaje de despedida
    Des.pu()
    Des.ht()
    Des.write("¡Gracias por jugar!", align="center", font=("Broadway", 32, "normal"))
    time.sleep(2)
    Des.pd()
    Des.st()
#Empiezo a poner colores y las características que tendrán los círculos decorativos para la
pantalla de despedida (radio, coordenadas aleatorias, grueso, etc)
    colors=["misty rose", "LightYellow2", "LightSalmon", "LightCyan", "lavender blush", "medium
aquamarine", "papaya whip", "plum", "honeydew", "light green"]
    Des.pensize(7)
    Des.shape("circle")
    Des.width(5)
    Des.speed(0)
    for i in range (60):
        Des.color(random.choice(colors))
        rand1=random.randrange(-320,320)
        rand2=random.randrange(-320,320)
        Des.up()
        Des.setpos((rand1, rand2))
        Des.pd()
        Des.begin_fill()
        Des.circle(random.randrange(75,160))
        Des.end_fill()
    turtle.bye()
#####
#Inicia pantalla de instrucciones
turtle.Turtle._screen = None
turtle.TurtleScreen._RUNNING = True
setup(.75,.85)
ins=turtle.Turtle()
Instrucciones=""Este juego requerirá el uso de operadores aritméticos para resolver los problemas.
¿Qué quiere decir esto?
Si se te pide formar un rectángulo naranja con dos figuras y dos colores entonces
deberás escribir:

```

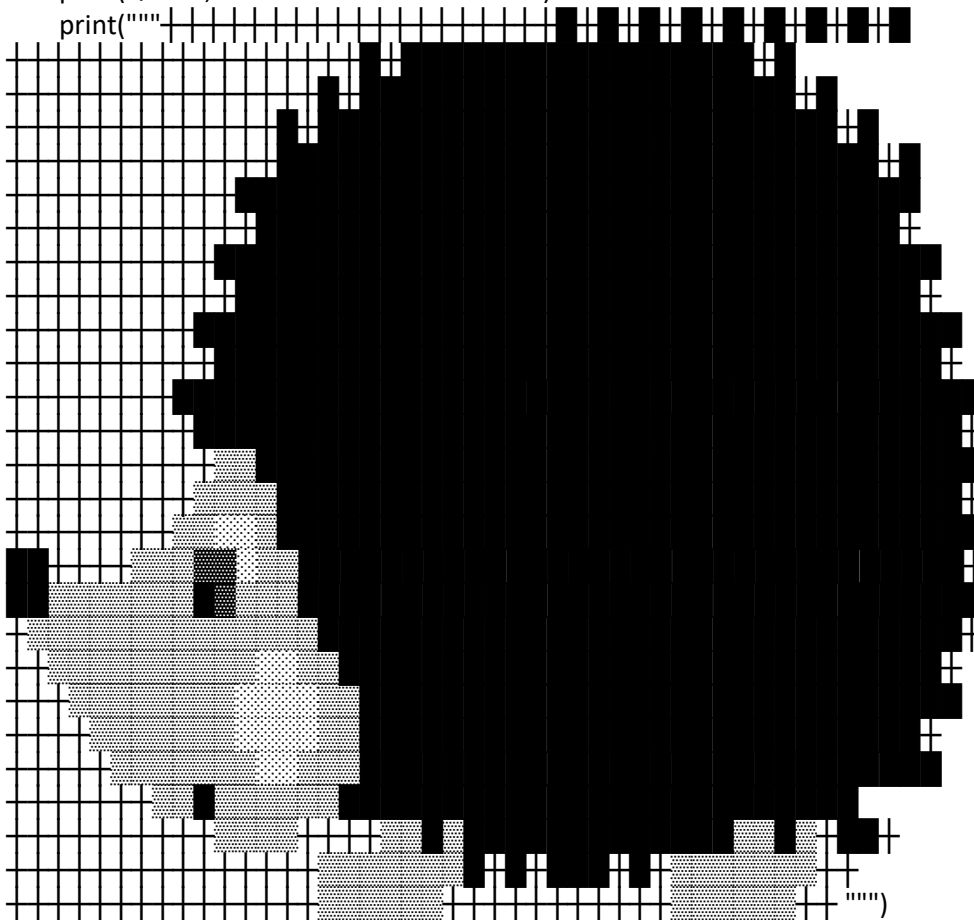
-Para la parte de la figura: cuadrado*2 ó 2*cuadrado

[illegible]

```
print("Tu puedes, piensa mejor en tu respuesta")  
print("""
```



```
elif cal>=2 and cal<3:  
    print("¡Casi!, revisa bien los enunciados")  
    print("""
```



```
elif cal==3:  
    print("¡Excelente! Sigue así")
```



```

    retro(cal)
#####
Empiezan los problemas como tal, que se encuentran en funciones llamadas por
el menú. Después de cada respuesta, se lleva un contador que aumenta sus
puntos, si es que contestaron bien.
#Se definen las funciones para el nivel fácil y difícil
def facil():
#Se establecen las variables para llevar el conteo del score
    scoref1=0
    scoref2=0
    scoref3=0
#Nivel 1 fácil
    turtle.Turtle._screen = None
    turtle.TurtleScreen._RUNNING = True
#Escribe las instrucciones del nivel
    N1=turtle.Turtle()
    N1.pu()
    N1.ht()
    N1.goto(0,200)
    N1.write("Escribe la ecuación para hacer un rombo morado", align="center", font=("Arial", 18,
"normal"))
#Dibuja un rombo morado, usando dos triángulos hechos con for
    N1.goto(75,0)
    N1.width(5)
    N1.pd()
    N1.st()
    N1.color("purple")
    N1.begin_fill()
    for i in range(3):
        N1.stamp()
        N1.left(120)
        N1.forward(125)
    N1.end_fill()
    N1.begin_fill()
    for i in range(3):
        N1.stamp()
        N1.rt(120)
        N1.forward(125)
    N1.end_fill()
    turtle.exitonclick()
#Termina la ventana y pide las respuestas
    Rc=input("Dame la ecuación para el color ").lower()
    if Rc=="azul+rojo" or Rc=="rojo+azul":
        print("Correcto ✓")
        scoref1+=0.5
    else:
        print("Respuesta inválida o incorrecta ✗")

```

```

Rf=input("Dame la ecuación para la figura ").lower()
if Rf=="2*triangulo" or Rf=="triangulo*2":
    print("Correcto ✓")
    scoref1+=0.5
else:
    print("Respuesta inválida o incorrecta ✗")

```

#Empieza nivel 2 fácil

```

turtle.Turtle._screen = None
turtle.TurtleScreen._RUNNING = True
N2=turtle.Turtle()
N2.shape("turtle")

```

#Escribe instrucciones del nivel

```

N2.pu()
N2.ht()
N2.goto(0,200)
N2.write("¿A partir de 2 trapecios y el color rojo y blanco,\n que figura de color puedes crear?",
align="center", font=("Arial", 18, "normal"))

```

#Dibuja un trapecio rojo

```

N2.goto(-50,-50)
N2.width(5)
N2.pd()
N2.st()
N2.color("red")
N2.begin_fill()
N2.lt(110)
N2.forward(110)
for i in range(2):
    N2.lt(70)
    N2.forward(110)
N2.lt(110)
N2.forward(185)
N2.end_fill()

```

#Dibuja un trapecio blanco

```

N2.pu()
N2.ht()
N2.goto(235,-50)
N2.pd()
N2.st()
N2.color("black","white")
N2.begin_fill()
N2.lt(110)
N2.forward(110)
for i in range(2):
    N2.lt(70)
    N2.forward(110)
N2.lt(110)

```

```

N2.forward(185)
N2.end_fill()
turtle.exitonclick()
Rc=input("Dame el color resultante ").lower()
if Rc=="rosa" or Rc=="rosado":
    print("Correcto ✓")
    scoref2+=0.5
else:
    print("Respuesta inválida o incorrecta ✗")
Rf=input("Dame la figura resultante ").lower()
if Rf=="romboide" or Rf=="hexagono":
    print("Correcto ✓")
    scoref2+=0.5
else:
    print("Respuesta inválida o incorrecta ✗")
time.sleep(1)

```

#Empieza nivel 3 fácil

```

turtle.Turtle._screen = None
turtle.TurtleScreen._RUNNING = True
N3=turtle.Turtle()
N3.shape("turtle")

```

#Dibuja un pentagono verde

```

N3.pu()
N3.ht()
N3.goto(0,200)
N3.write("Dame la ecuación para un pentagono verde usando\n el menor número de figuras
posibles", align="center", font=("Arial", 18, "normal"))
N3.goto(100,-115)
N3.width(5)
N3.pd()
N3.st()
N3.color("green")
N3.begin_fill()
for i in range(5):
    N3.lt(72)
    N3.forward(150)
N3.end_fill()
turtle.exitonclick()

```

#Termina la ventana y pide las respuestas

```

Rc=input("Dame la ecuación del color ").lower()
if Rc=="amarillo+azul" or Rc=="azul+amarillo":
    print("Correcto ✓")
    scoref3+=0.5
else:
    print("Respuesta inválida o incorrecta ✗")
Rf=input("Dame la ecuación de la figura ").lower()

```

```

if Rf=="trapecio+triangulo" or Rf=="triangulo+trapecio":
    print("Correcto ✓")
    scoref3+=0.5
else:
    print("Respuesta inválida o incorrecta ✗")
    scoref(scoref1,scoref2,scoref3)
    time.sleep(2)
#####
Empieza la función para los niveles difíciles, los difíciles cambian las variables
de las preguntas cada vez de forma aleatoria

```

```

def dificil():
#Se establecen las variables para llevar el conteo del score
    scored1=0
    scored2=0
    scored3=0
#Empieza nivel 1 difícil
    turtle.Turtle._screen = None
    turtle.TurtleScreen._RUNNING = True
    N1=turtle.Turtle()
    N1.shape("turtle")
#Instrucciones y variables del nivel
    d=random.randint(8,10)
    b=random.randint(5,7)
    h=random.randint(2,4)
    N1.pu()
    N1.ht()
    N1.goto(0,180)
    N1.write(f"¿Cuál es el área total de la siguiente figura\n si el diametro del círculo es {d} la base\n
del triángulo es {b} y su altura {h}?", align="center", font=("Arial", 18, "normal"))
#Dibuja el cuerpo(círculo) del pez
    N1.goto(-50,-100)
    N1.width(5)
    N1.pd()
    N1.st()
    N1.color("orange")
    N1.begin_fill()
    N1.circle(75)
    N1.end_fill()
#Dibuja la cola(triángulo) del pez
    N1.pu()
    N1.goto(85,-115)
    N1.rt(90)
    N1.begin_fill()
    N1.pd()
    N1.rt(150)

```

```

N1.fd(115)
N1.rt(60)
N1.fd(115)
N1.end_fill()
#Dibuja el ojo del pez
N1.pu()
N1.ht()
N1.color("black")
N1.goto(-70,5)
N1.pd()
for i in range(2):
    N1.lt(80)
    N1.fd(15)
#Dibuja el texto de AVISO
N1.pu()
N1.goto(0,-250)
N1.write("AVISO: Toma pi como 3.14", align="center", font=("Arial", 18, "normal"))
turtle.exitonclick()
#Define los valores para las respuestas
r=d/2
a=round((3.14*(r**2)+(b*h)/2),2)
#Se le pregunta al usuario y responde la pregunta
Rf=input("¿Cuál es el área total? (usa dos decimales) ")
if Rf==str(a) or Rf==str(a-.01):
    print("Correcto ✓")
    scored1+=1
else:
    print("Respuesta inválida o incorrecta ✗")
    time.sleep(1)
#Nivel difícil 2
turtle.Turtle._screen = None # force recreation of singleton Screen object
turtle.TurtleScreen._RUNNING = True # only set upon TurtleScreen() definition
N2=turtle.Turtle()
N2.shape("turtle")
#Escribe las instrucciones del nivel
N2.pu()
N2.ht()
N2.goto(0,180)
M=["una piramide triangular","una piramide cuadrangular","un prisma triangular"]
index=random.randrange(len(M))
item=M[index]
N2.write(f"Imagina que lo que estas viendo son cubos\n y {item} de frente\n ¿Cuántos vertices
hay en total entonces?", align="center", font=("Arial", 18, "normal"))
#Dibuja los tres cubos(vistos de frente)
N2.goto(-200,-150)
N2.width(5)
N2.pd()

```



```

N2.st()
N2.begin_fill()
x=-200
for i in range(3):
    x+=140
    N2.pu()
    N2.begin_fill()
    N2.goto(x,-150)
    N2.pd()
    N2.color("black",random.sample(colors,1))
for i in range(4):
    N2.lt(90)
    N2.fd(140)
N2.end_fill()
N2.pu()
#Dibuja la piramide triangular(vista de frente)
N2.goto(-60,-10)
N2.pd()
N2.color("black",random.sample(colors,1))
N2.begin_fill()
for i in range(3):
    N2.fd(140)
    N2.lt(120)
N2.end_fill()
#Dibuja el texto de AVISO
N2.pu()
N2.ht()
N2.color("green")
N2.goto(0,-250)
N2.write("AVISO: Los vertices son los puntos\n donde se unen los lados de un polígono",
align="center", font=("Arial", 18, "normal"))
turtle.exitonclick()
#Calculo de vertices para la respuesta con las variables
if index==0:
    vertices=28
elif index==1:
    vertices=29
elif index==2:
    vertices=30
#Se le pregunta al usuario y responde la pregunta
Rf=input("¿Cuántos vertices hay? ")
if Rf==str(vertices):
    print("Correcto ✓")
    scored2+=1
else:
    print("Respuesta inválida o incorrecta ✗")
time.sleep(1)

```

#Empieza el nivel difícil 3

```
turtle.Turtle._screen = None
turtle.TurtleScreen._RUNNING = True
N3=turtle.Turtle()
N3.shape("turtle")
```

#Define las variables aleatorias para el problema y su resultado

```
M=(random.randint(25,30))
x=round((M-6)/4,2)
```

#Instrucciones del nivel

```
N3.pu()
N3.ht()
N3.goto(0,180)
N3.write(f"¿Cuál es el valor de x si el perímetro\n del rectángulo es igual a {M}?", align="center",
font=("Arial", 18, "normal"))
```

#Dibuja al rectángulo

```
N3.goto(75,-50)
N3.width(5)
N3.pd()
N3.st()
N3.begin_fill()
N3.color("black",random.choice(colors))
for i in range(2):
    N3.lt(90)
    N3.fd(120)
    N3.lt(90)
    N3.fd(180)
N3.end_fill()
N3.pu()
```

#Escribe los datos de los lados

```
N3.goto(-12,-87)
N3.color("black")
N3.write("x+3", align="center", font=("Arial", 20, "normal"))
N3.goto(100,-5)
N3.write("x", align="center", font=("Arial", 20, "normal"))
```

#Dibuja el texto de AVISO

```
N3.pu()
N3.ht()
N3.color("green")
N3.goto(0,-250)
N3.write("AVISO: En caso de obtener decimales redondear hasta\n el segundo. Si te da entero,
agrega .0 (e.g:8.0)", align="center", font=("Arial", 18, "normal"))
turtle.exitonclick()
```

#Se le pregunta al usuario y responde la pregunta

```
Rf=input("¿Cuánto vale x? ")
if Rf==str(x):
    print("Correcto ✓")
```

```
        scored3+=1
    else:
        print("Respuesta inválida o incorrecta X")
        scored(scored1,scored2,scored3)
        time.sleep(2)
#####
#Salta el menú, puedes elegir jugar indefinidamente, a menos que elijas
salir(3)
```

```
opcion=0
print("AVISO: No debes usar acentos o tendrás la respuesta mal")
while opcion!=3:
    #Aviso de reglas
    opcion=int(input("MENÚ\n Elige un nivel de dificultad:\n 1)Fácil\n 2)Difícil\n 3)Salir\n"))
    if opcion==1:
        facil()
    elif opcion==2:
        dificil()
    elif opcion==3:
        despedida()
    else:
        print('opción inválida')
```

Conclusiones:

Los desarrollos de programas computacionales tienen un enorme potencial social, tienen la capacidad de resolver problemas muy complejos en tiempos imposibles para seres humanos o, por el contrario, generar miles de problemas que nosotros podemos resolver, cosa que tiene uso, sobre todo a la hora de enseñar. Estos programas, además, permiten el uso de herramientas y acciones que normalmente no se podrían, ya sea con el uso de videos, audios, animaciones, retroalimentación inmediata y muchas otras más. Que no solo pueden facilitar el aprendizaje y darle una nueva dimensión, si no que de igual manera lo hace más divertido y entendible para niños, como es el caso del proyecto realizado. Un juego simplemente atrae más la atención de un niño y permite que pueda aprender sin que sienta que se trata de una tarea.

Encima de esto, un programa puede hacerla de profesor hasta cierto punto, y a través de retroalimentación y consejos, abre la posibilidad de un autoaprendizaje, que luego se puede mejorar con clases más personalizadas.

Instrucciones de uso:

Para interactuar con el programa es necesario hacer click en la ventana que sale una vez que esta complete su animación (Solo a partir de las instrucciones, la ventana de bienvenida es la única que pasa sola)

Una vez que se da click para seguir, el programa hace una pregunta que se tiene que responder tecleando en el Shell.

Hay que asegurarse de que el cursor para escribir esté a lado de la pregunta porque a veces no se pone automáticamente después de hacer click en una ventana.

Referencias consultadas:

cdlane. (2017). Re-open turtle after turtle.bye(). octubre 18, 2019, de stack overflow Sitio web: <https://stackoverflow.com/questions/44249534/re-open-turtle-after-turtle-bye>