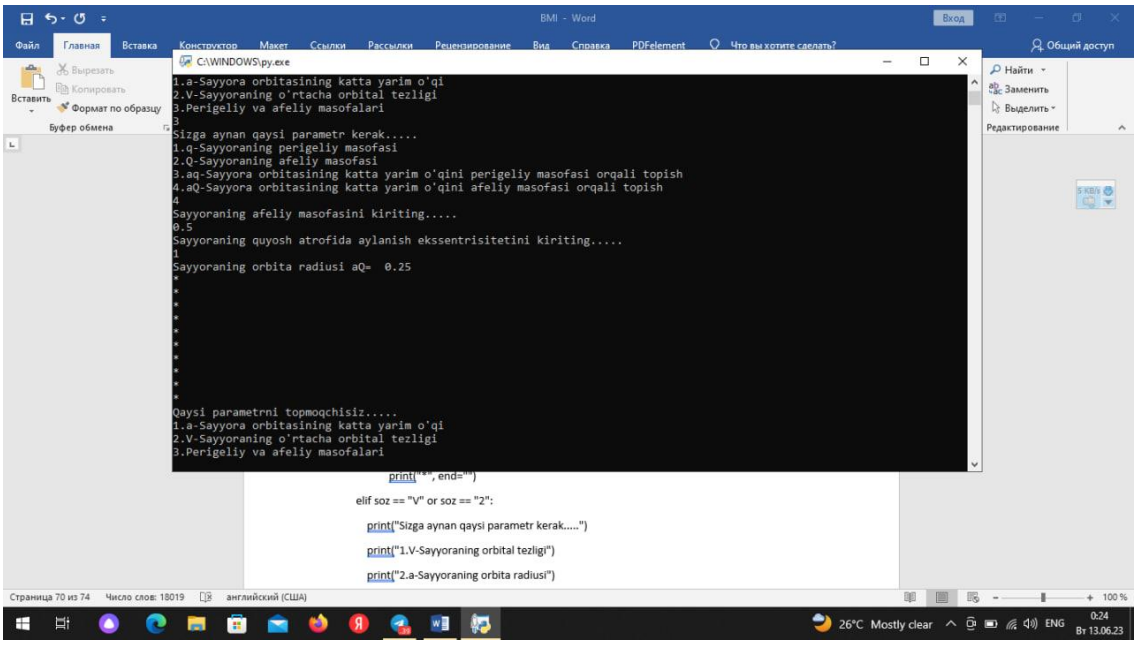
JOURNAL OF UNIVERSAL SCIENCE RESEARCH

ISSN (E): 2181-4570

PYTON DASTURIDA ASTRONOMIYADAN ANIMATSIYA YARATISH

Haydarova Dilorom- Navoiy davlat pedagogika instituti magistranti
Ilmiy rahbar: Sayfullayeva Gulhayo- Navoiy davlat pedagogika instituti dotsenti




```
elif sozi == "aq" or sozi == "3":
print("Sayyoraning perigeliy masofasini kiriting.....")
q = float(input())
print("Sayyoraning quyosh atrofida aylanish ekssentrisitetini kiriting.....")
e = float(input())
print("Sayyoraning orbita radiusi aq= ", q / (1 - e))
for i in range(10):
print("*", end="")
elif sozi == "aQ" or sozi == "4":
print("Sayyoraning afeliy masofasini kiriting.....")
Q = float(input())
print("Sayyoraning quyosh atrofida aylanish ekssentrisitetini kiriting.....")
e = float(input())
print("Sayyoraning orbita radiusi aQ= ", Q / (1 + e))
for i in range(10):
```

ISSN (E): 2181-4570

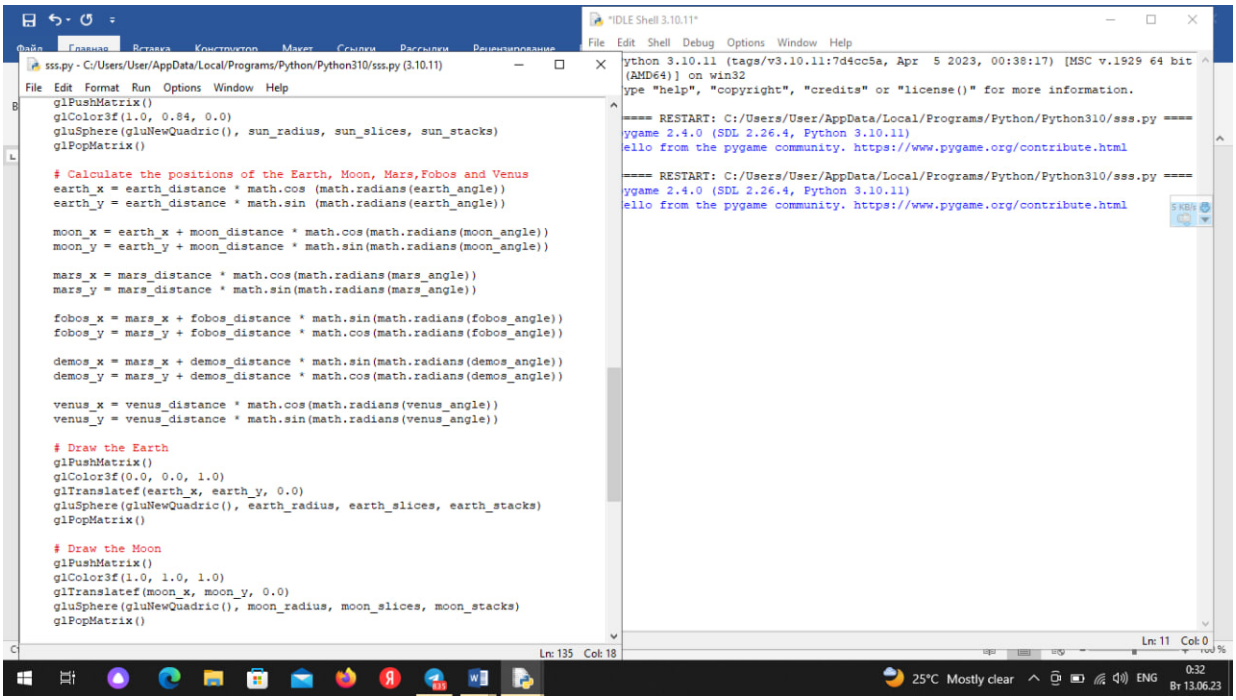
```
print("*", end="")
```

Animatsiya kodi:

```
import sys
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *
import math

# Initialize Pygame and OpenGL
pygame.init()
width, height = 1000, 700
pygame.display.set_mode((width, height), DOUBLEBUF | OPENGL)

# Set up the projection matrix
glMatrixMode(GL_PROJECTION)
gluPerspective(45, (width / height), 0.1, 50.0)

# Set up the modelview matrix
glMatrixMode(GL_MODELVIEW)
gluLookAt(0, -20, 10, 0, 0, 0, 0, 0, 1)

# Enable depth testing
glEnable(GL_DEPTH_TEST)
```

Rasm-5:

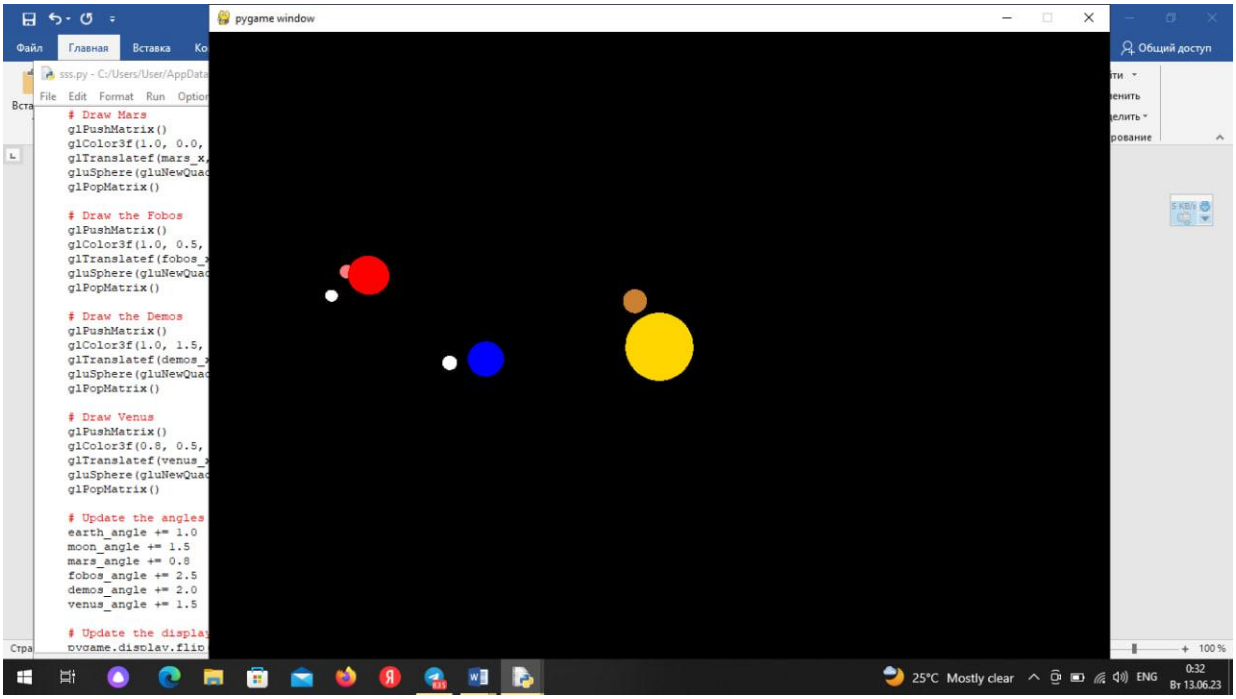
ISSN (E): 2181-4570



# Set up the Sun
sun_radius = 1.0
sun_slices = 30
sun_stacks = 30

Rasm-6:

ISSN (E): 2181-4570

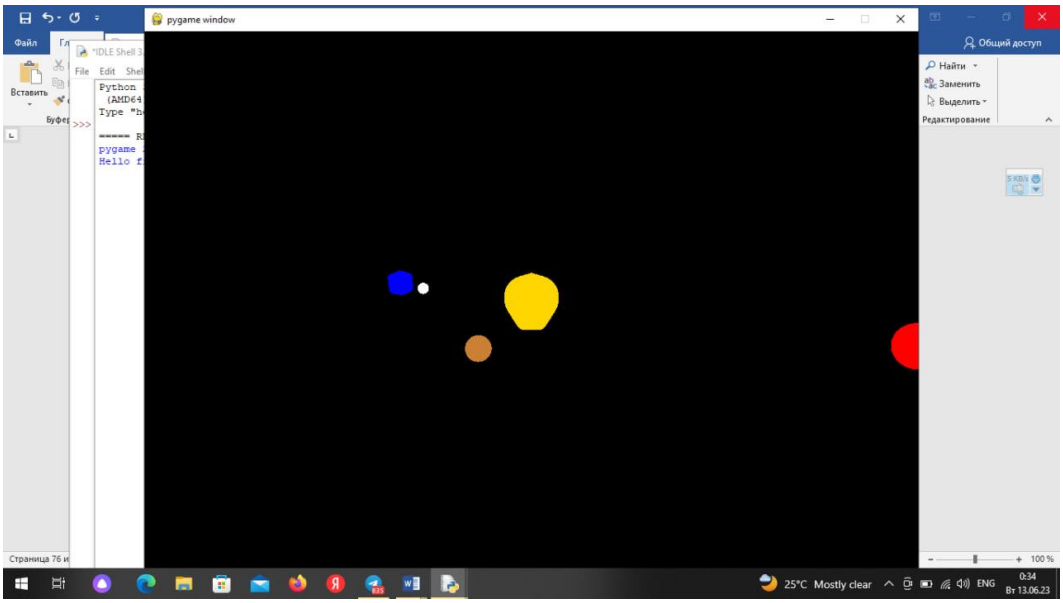



```
# Set up the Earth
earth_radius = 0.5
earth_distance = 5.0
earth_slices = 30
earth_stacks = 30
earth_angle = 0.0

# Set up the Moon
moon_radius = 0.2
moon_distance = 1.0
moon_slices = 30
moon_stacks = 30
moon_angle = 90.0

# Set up Mars
mars_radius = 0.7
mars_distance = 12.0
mars_slices = 30
mars_stacks = 30
```

ISSN (E): 2181-4570

mars_angle = 0.0

Rasm-7:



# Set up the Fobos
fobos_radius = 0.25
fobos_distance = 1.0
fobos_slices = 30
fobos_stacks = 30
fobos_angle = 0.0

# Set up the Demos
demos_radius = 0.2
demos_distance = 2.0
demos_slices = 30
demos_stacks = 30
demos_angle = 180.0

# Set up Venus
venus_radius = 0.4
venus_distance = 3.5
venus_slices = 30

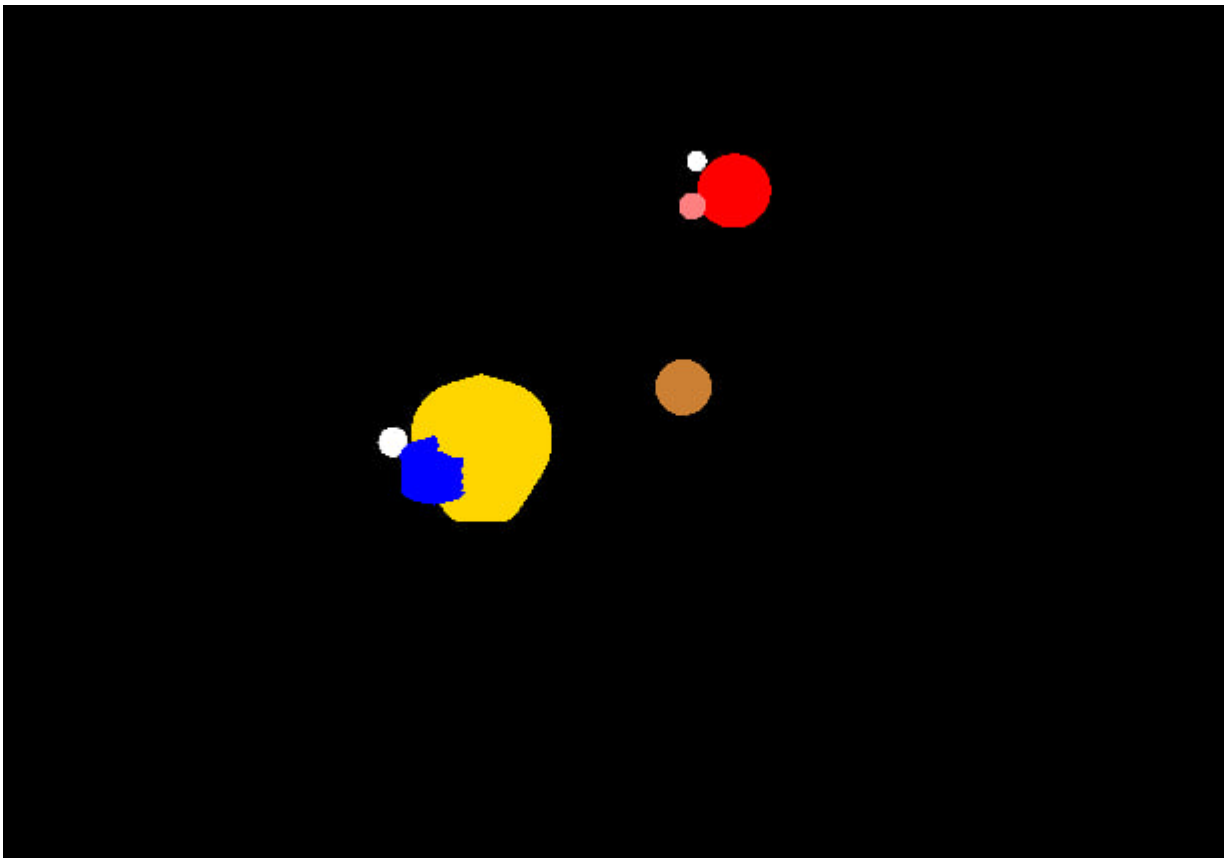
ISSN (E): 2181-4570

```
venus_stacks = 30
venus_angle = 0.0

# Main animation loop
while True:
for event in pygame.event.get():
if event.type == pygame.QUIT:
pygame.quit()
sys.exit()

# Clear the screen and depth buffer
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

# Draw the Sun
glPushMatrix()
glColor3f(1.0, 0.84, 0.0)
gluSphere(gluNewQuadric(), sun_radius, sun_slices, sun_stacks)
glPopMatrix()

# Calculate the positions of the Earth, Moon, Mars,Fobos and Venus
earth_x = earth_distance * math.cos (math.radians(earth_angle))
earth_y = earth_distance * math.sin (math.radians(earth_angle))

moon_x = earth_x + moon_distance * math.cos(math.radians(moon_angle))
moon_y = earth_y + moon_distance * math.sin(math.radians(moon_angle))

mars_x = mars_distance * math.cos(math.radians(mars_angle))
mars_y = mars_distance * math.sin(math.radians(mars_angle))

fobos_x = mars_x + fobos_distance * math.sin(math.radians(fobos_angle))
fobos_y = mars_y + fobos_distance * math.cos(math.radians(fobos_angle))

demos_x = mars_x + demos_distance * math.sin(math.radians(demos_angle))
```

ISSN (E): 2181-4570

```
demos_y = mars_y + demos_distance * math.cos(math.radians(demos_angle))
```

Rasm-8:

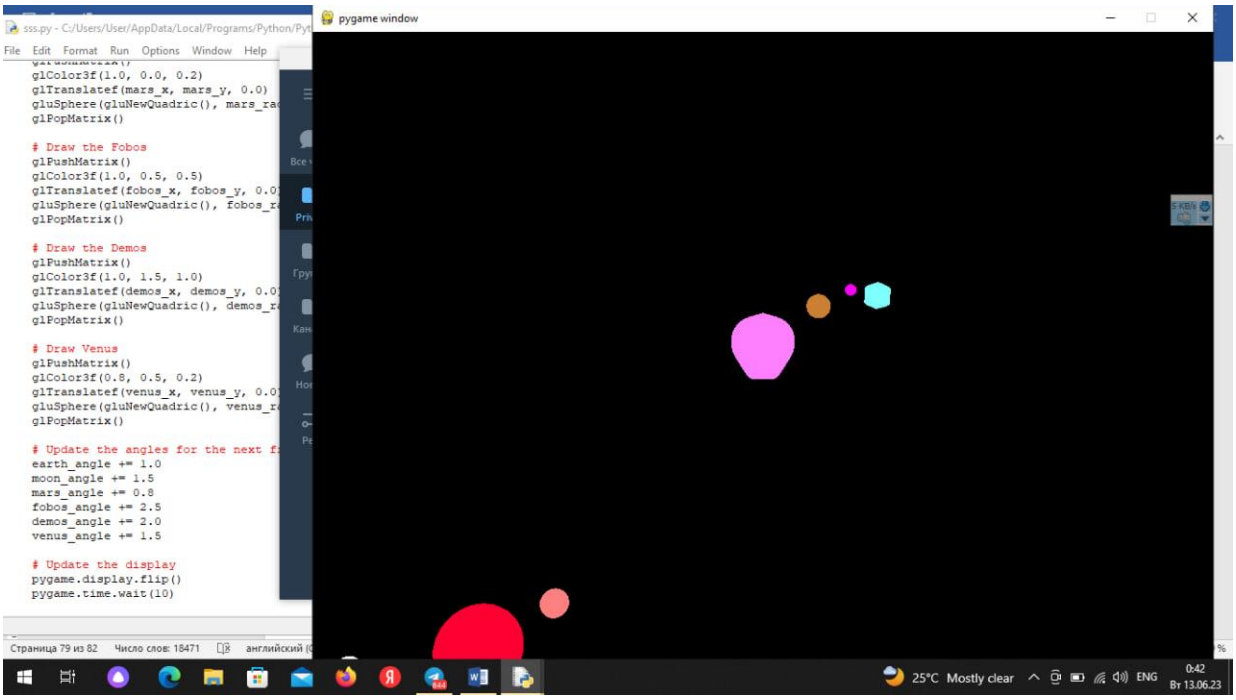


```
venus_x = venus_distance * math.cos(math.radians(venus_angle))
venus_y = venus_distance * math.sin(math.radians(venus_angle))

# Draw the Earth
glPushMatrix()
glColor3f(0.0, 0.0, 1.0)
glTranslatef(earth_x, earth_y, 0.0)
gluSphere(gluNewQuadric(), earth_radius, earth_slices, earth_stacks)
glPopMatrix()

# Draw the Moon
glPushMatrix()
```

ISSN (E): 2181-4570

```
glColor3f(1.0, 1.0, 1.0)
glTranslatef(moon_x, moon_y, 0.0)
gluSphere(gluNewQuadric(), moon_radius, moon_slices, moon_stacks)
glPopMatrix()

# Draw Mars
glPushMatrix()
glColor3f(1.0, 0.0, 0.0)
glTranslatef(mars_x, mars_y, 0.0)
gluSphere(gluNewQuadric(), mars_radius, mars_slices, mars_stacks)
glPopMatrix()
```

Rasm-9:




```
# Draw the Fobos
glPushMatrix()
glColor3f(1.0, 0.5, 0.5)
glTranslatef(fobos_x, fobos_y, 0.0)
gluSphere(gluNewQuadric(), fobos_radius, fobos_slices, fobos_stacks)
glPopMatrix()
```

ISSN (E): 2181-4570

```
# Draw the Demos
glPushMatrix()
glColor3f(1.0, 1.5, 1.0)
glTranslatef(demos_x, demos_y, 0.0)
gluSphere(gluNewQuadric(), demos_radius, demos_slices, demos_stacks)
glPopMatrix()

# Draw Venus
glPushMatrix()
glColor3f(0.8, 0.5, 0.2)
glTranslatef(venus_x, venus_y, 0.0)
gluSphere(gluNewQuadric(), venus_radius, venus_slices, venus_stacks)
glPopMatrix()

# Update the angles for the next frame
earth_angle += 1.0
moon_angle += 1.5
mars_angle += 0.8
fobos_angle += 2.5
demos_angle += 2.0
venus_angle += 1.5

# Update the display
pygame.display.flip()
pygame.time.wait(10)
```