

Computer Graphics PA1

Report

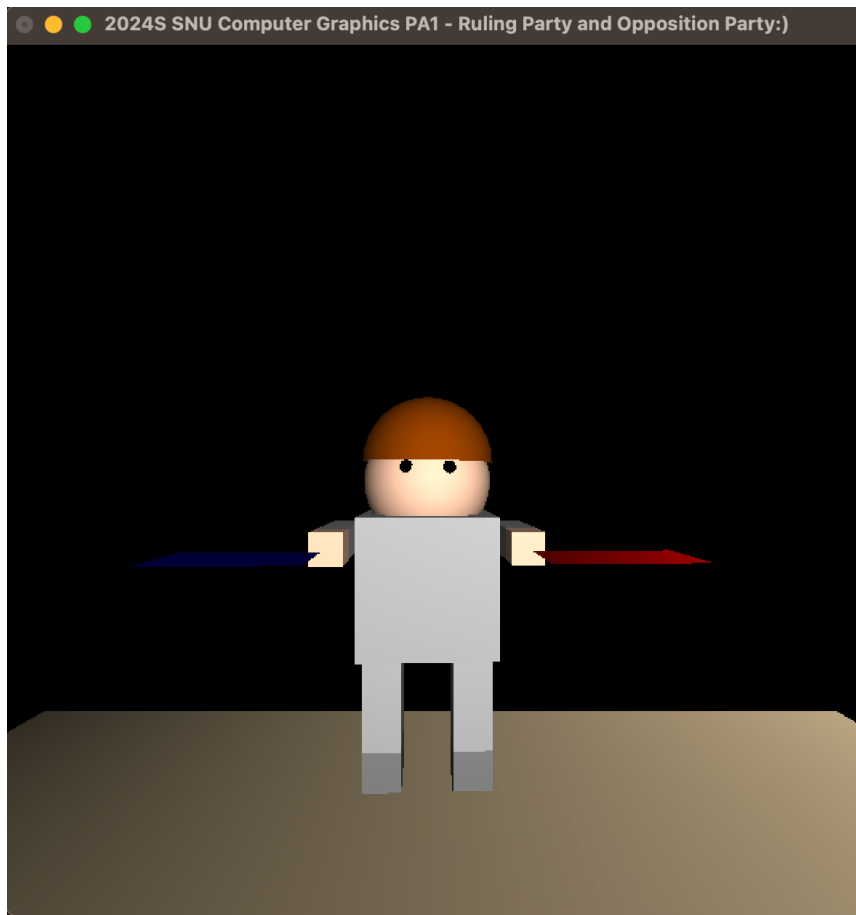
2021-90029

전기정보공학부 최대현

Description

- 작품명: Ruling Party and Opposition Party (여당과 야당)
- 3-level hierarchy를 가지고 있는 사람 (유권자)를 구현하였다.
- 모델의 animation 또한 3-level hierarchy를 갖는다.
- PyOpenGL, GLUT를 이용하여 구현하였고, 일부 기능 (음악 재생 등)에는 pygame을 이용하였다.
- 별도의 shading은 구현하지 않았고, Primitive (Sphere, Cube, 2D Rectangle)에 색을 입혀 모든 구성요소를 구현하였다.

Structure



Hierarchical structure 의 overview 는 다음과 같다.

```
Voter
- Head
  - Hair
  - left_eye
  - right_eye
- Body
  - left_arm
    - ArmNode
      - ArmNode
        - FlagNode
  - right_arm
    - ArmNode
      - ArmNode
        - FlagNode
  - left_leg
    - LegNode
      - LegNode
  - left_leg
    - LegNode
      - LegNode
```

1. Head

```
class Head(Model):

    def __init__(self):
        self.roll = 0.0
        self.animation = HeadShaking(0.02)
        self.animation.ratio = 0.5

    def _draw(self):
        glColor(0.98, 0.81, 0.69)
        glRotate(self.roll, 0.0, 0.0, 1.0)
        glutSolidSphere(1.0, 32, 32)

        glPushMatrix()
        # define left eye
        glColor3f(0.0, 0.0, 0.0) # black color
        glTranslatef(-0.3, 0.2, 0.9)
        glutSolidSphere(0.1, 10, 10)
        glPopMatrix()

        # define right eye
        glPushMatrix()
        glTranslatef(0.3, 0.2, 0.9)
```

```

glutSolidSphere(0.1, 10, 10)
glPopMatrix()

## define hair
glColor3f(0.5, 0.25, 0.0) # brown
glPushMatrix()
glTranslatef(0.0, 0.3, 0.0)
## define clipping plane, to express hair (up-half)
clip_plane = (0.0, 0.2, 0.0, 0.0)
glEnable(GL_CLIP_PLANE0)
glClipPlane(GL_CLIP_PLANE0, clip_plane)
glutSolidSphere(1.03, 32, 32)
glDisable(GL_CLIP_PLANE0)
glPopMatrix()

```

- 살구색의 Sphere로 얼굴을 정의했고, 갈색의 Clipped Sphere로 머리카락, 검은색의 작은 Sphere로 눈을 정의했다.
- Head Shaking을 animation이라는 이름의 멤버 변수로 가진다.

2. Body

```

### Body
class Body(Model):
    def __init__(self):
        self.left_arm = Arm(-90, -1.5, (0.0, 0.0, 1.0)) # define left arm, begin with -90 degree.
        self.right_arm = Arm(-90, 1.5, (1.0, 0.0, 0.0)) # define right arm, begin with -90 degree.
        self.left_leg = Leg(0, -0.8)
        self.right_leg = Leg(0, 0.8)

    def _draw(self):
        glColor(0.7, 0.7, 0.7) # gray color
        glTranslate(0.0, -1.5, 0.0)
        glutSolidCube(2) # body is 2x2x2 solid cube.
        self.left_arm.render()
        self.right_arm.render()
        self.left_leg.render()
        self.right_leg.render()

    def update(self):
        self.left_arm.update()
        self.right_arm.update()

```

```
self.left_leg.update()
self.right_leg.update()
```

- 왼팔, 오른팔, 왼다리, 오른다리를 멤버 변수로 가진다.
- 몸통은 2x2x2의 Cube로 구현하였다.

3. Arm

```
class Arm(Model):
    def __init__(self, degree, displacement, color):
        self.degree = degree
        self.displacement = displacement
        self.node = ArmNode(ArmNode(FlagNode(color))) # last node is flag, colored RED or BLUE.

    def _draw(self):
        glTranslate(0.0, 0.5, 0.0)
        glTranslate(self.displacement, 0.0, -1.0)
        glRotate(self.degree, 1.0, 0.0, 0.0) # express each arm's movement
        self.node.render()

    def update(self):
        self.node.update()
```

- 현재 시점에서의 각도와 node를 멤버 변수로 가지는데, node는 2개의 ArmNode와 1개의 FlagNode로 구성되어 있다.
- 깃발을 올리고 내리는 동작을 구현하기 위해 현재 시점의 팔 각도를 멤버 변수로 가진다.

4. ArmNode

```
### Arm Node, Each arm consists of two ArmNode and one FlagNode.
class ArmNode(Model):

    def __init__(self, next_node=None):
        self.flag = None
        self.next_node = None
        if isinstance(next_node, FlagNode):
            self.flag = next_node
        else:
```

```

        self.next_node = next_node

    def _draw(self):
        if self.next_node: # If inner one (express arm with T-Shirts)
            glColor(0.7, 0.7, 0.7)
            glTranslate(0.0, 0.1, 0.0)
            glutSolidCube(0.5)
            glTranslate(0.0, -0.5, 0.0)
            glutSolidCube(0.5)
            glTranslate(0.0, -0.5, 0.0)
        elif self.flag: # If outer one (express hands)
            glColor(0.7, 0.7, 0.7)
            glTranslate(0.0, 0.1, 0.0)
            glutSolidCube(0.5)
            glTranslate(0.0, -0.5, 0.0)
            glColor(0.98, 0.81, 0.69)
            glutSolidCube(0.5)
            glTranslate(0.0, -0.5, 0.0)
        if self.next_node:
            self.next_node.render()
        elif self.flag:
            self.flag.render()

    def update(self):
        if self.next_node:
            self.next_node.update()

```

- 자신의 노드가 ArmNode인 경우 (팔의 안쪽 노드)와 자신의 노드가 FlagNode인 경우 (바깥쪽 노드)로 구분해서, 색상을 다르게 구현하였다.

5. FlagNode

```

class FlagNode(Model):

    def __init__(self, color):
        self.R, self.G, self.B = color

    def _draw(self):
        glColor3f(self.R, self.G, self.B)
        width, height = 2.0, 1.0 # width and height of flag

        glBegin(GL_QUADS) # start to draw

```

```

if self.R != 0:
    glVertex3f(0, 0, 0)
    glVertex3f(0, -height, 0)
    glVertex3f(width, -height, 0)
    glVertex3f(width, 0, 0)
elif self.B != 0:
    glVertex3f(0, 0, 0)
    glVertex3f(0, -height, 0)
    glVertex3f(-width, -height, 0)
    glVertex3f(-width, 0, 0)
glEnd() # end to draw

```

- glVertex3f를 이용하여 2차원 평면의 직사각형으로 구현했고, 색상에 따라 위치가 다르게 (왼팔과 오른팔에 각각 빨간색, 파란색 깃발) 구현했다.

6. Leg

```

### Leg
class Leg(Model):
    def __init__(self, degree, displacement):
        self.degree = degree
        self.displacement = displacement
        self.node = LegNode(LegNode())

    def _draw(self):
        glTranslate(0.0, -1.5, -1.0)
        glTranslate(self.displacement, 0.0, 0.0)
        glRotate(self.degree, 0.0, 0.0, 1.0)
        self.node.render()

    def update(self):
        self.node.update()

```

- Arm과 마찬가지로 node를 가지고 있고, node는 2개의 LegNode로 구성되어 있다.
- 점프 시 다리를 벌렸다 모으는 동작을 구현하기 위해 현재 시점의 다리 각도를 멤버 변수로 가진다.

7. LegNode

```

#### Leg Node, Each leg consists of two LegNodes.
class LegNode(Model):
    def __init__(self, next_node=None):
        self.next_node = next_node

    def _draw(self):
        if self.next_node: # If inner one, expresses leg with pants.
            glColor(0.7, 0.7, 0.7)
            glTranslate(0.0, 0.0, 0.0)
            glutSolidCube(0.7)
            glTranslate(0.0, -0.7, 0.0)
            glutSolidCube(0.7)
            glTranslate(0.0, -0.7, 0.0)

        else: # If outer one
            glColor(0.7, 0.7, 0.7)
            glTranslate(0.0, 0.0, 0.0)
            glutSolidCube(0.7)
            glTranslate(0.0, -0.7, 0.0)
            glColor(0.5, 0.5, 0.5) # expresses shoes
            glutSolidCube(0.7)
            glTranslate(0.0, -0.7, 0.0)

        if self.next_node:
            self.next_node.render()

    def update(self):
        if self.next_node:
            self.next_node.update()

```

- 자신의 노드가 LegNode인 경우 (다리의 안쪽 노드)와 자신의 노드가 없는 경우 (다리의 바깥쪽 노드)로 구분해서, 색상을 다르게 구현하였다.

8. Voter

```

#### Model, Human having 2 different (Blue and Red) flags.
class Voter(Model):
    def __init__(self, pos, scale):
        self.pos_x, self.pos_y, self.pos_z = pos
        self.scale_x, self.scale_y, self.scale_z = scale
        self.jump = 0.0
        self.head = Head()

```

```

self.body = Body()

def _draw(self):
    glTranslate(self.pos_x, self.pos_y, self.pos_z)
    glTranslate(0.0, self.jump, 0)
    glScale(self.scale_x, self.scale_y, self.scale_z)
    self.head.render()
    self.body.render()

def update(self):
    self.head.update()
    self.body.update()

```

- 처음 렌더링 시 필요한 (x, y, z) 위치와 scale factor을 가지고 있다.
- Jump 시 모델의 높이를 조작해야 하므로 현재 시점의 위치를 멤버 변수로 갖는다.
- Head 객체와 Body 객체를 initialize한 뒤 멤버 변수로 갖는다.

Animation

- 모든 조작은 키보드 입력을 받아 수행되며, 키보드 입력을 누르지 않으면 모델은 움직이지 않는다.

Model's motion

- 모델은 왼손에 빨간색 깃발을, 오른손에 파란색 깃발을 들고 있다.
- 키보드로 조작할 수 있는 모델의 움직임은 3가지이다. 움직임은 다음과 같다.
 1. 's'키를 누르면 모델은 오른팔을 들어올리고, 왼팔을 내린다. 결과적으로, 모델은 빨간색 깃발을 위로 올리게 된다.
 2. 'w'키를 누르면 모델은 왼팔을 들어올리고, 오른팔을 내린다. 결과적으로, 모델은 파란색 깃발을 위로 올리게 된다.
 3. 'j'키를 누르면 모델은 점프를 한다. 이 때 모델은 가장 높이 있을 때 양 다리를 각각 30도까지 벌렸다가 착지할 때 다리가 완전히 모아지도록 움직인다.

Viewpoint change

- 키보드로 카메라의 viewpoint를 조작(회전)할 수 있다.
 1. 'a'키를 누르면 카메라의 viewpoint가 시계방향으로 회전한다.
 2. 'd'키를 누르면 카메라의 viewpoint가 반시계방향으로 회전한다.

Shaking Head

- 키보드의 키를 누를 때마다 display 함수가 호출되는데, 이 함수 내에서 model.render()를 호출하고, 이 때마다 모델이 머리를 좌우로 흔드는 모션을 반복한다.

Artistic Point (Extra Implementation)

```
### Function to play music, related with model's arm position
def play_music(file_path):
    global current_music_file
    if current_music_file == file_path and pygame.mixer.music.get_busy():
        return
    pygame.mixer.music.stop()
    pygame.mixer.music.load(file_path)
    pygame.mixer.music.play()
    current_music_file = file_path
```

- 모델이 빨간색 깃발을 올리고 있을 때는 국민의힘 정당 국회의원 후보의 CM송이 pygame.music.start() 함수를 호출함으로써 재생된다.
- 모델이 파란색 깃발을 올리고 있을 때는 더불어민주당 정당 국회의원 후보의 CM송이 pygame.music.start() 함수를 호출함으로써 재생된다.
- 정치적 중립을 지키기 위하여 처음 프로그램이 시작할 때 두 팔의 각도는 똑같이, 90도를 유지하고 있고, 이후에도 두 팔의 각도가 90도가 되는 시점에서는 재생되고 있던 음악을 정지한다.
- 실제 모션과 비슷하게 모델링하기 위해서, jump할 때 모델의 위치를 역학에서의 중력 가속도-변위 공식을 약간 수정하여 모델링하였다.

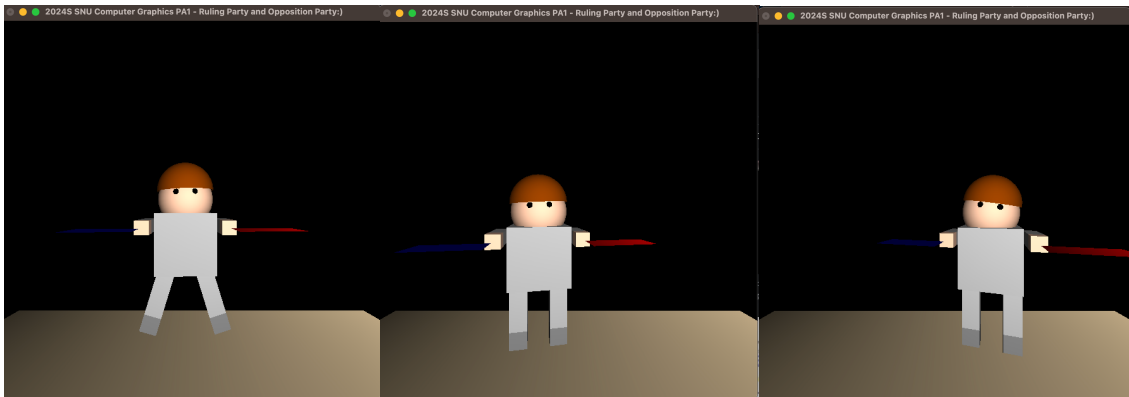
$$h = 5t - \frac{1}{2}gt^2$$

```
### Function to update the model's state
def update_jump():
    global robot_jump, y_pos, leg_degree, jump_start_time
    if robot_jump:
        t = time.time() - jump_start_time # calculate elapsed time
        y_pos = 0 + 5*t - 0.5*9.81*t**2 # calculate current position with gravity
        acceleration
        # Adjust degree based on jump phase
```

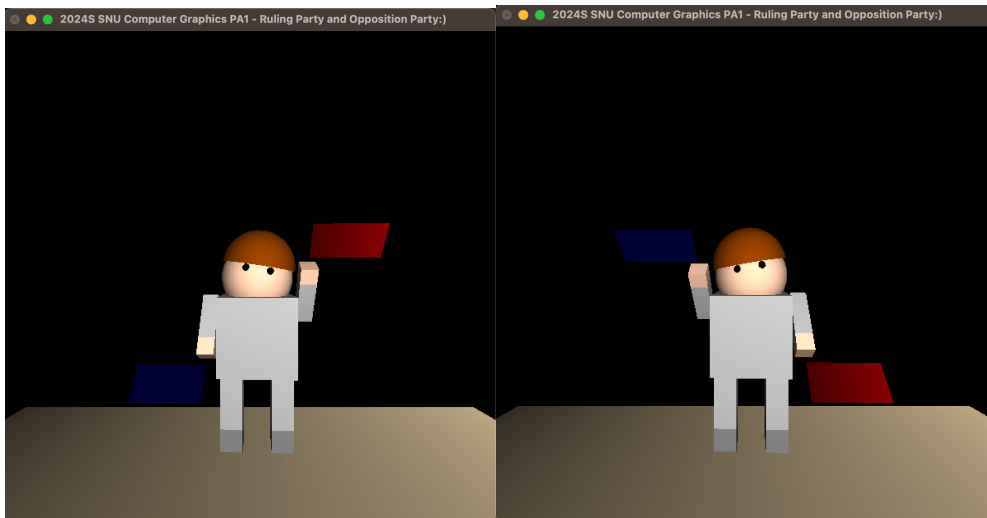
Instruction

- run.py가 있는 디렉토리로 이동하여 'python3 run.py' 를 입력하여 프로그램을 실행

행할 수 있다.



- Jump / Viewpoint를 change했을 때의 모습. ($j / a / d$)
- Viewpoint의 회전 각도에는 제한이 없다. (계속 누르면 원위치로 되 돌아온다.)
- Jump는 중력가속도-변위 공식으로 높이를 모델링하여, 시간이 지나면 땅에 착지한다.



- 왼팔 / 오른팔을 올려 깃발을 들었을 때의 모습. 실제로, 각 깃발에 맞는 음악도 재생된다. (w / s)
- 왼팔과 오른팔의 각도는 상하, 지면과 이루는 각도를 90도까지로 제한한다.