

NAMA : KRESHNA PUTRA ADI WICAKSANA
NIM / KELAS : V3920032 / TI-D
MATKUL : GAME DEVELOPMENT

TASK AND EVENT LANDING

1. Task

- a. Pada code dibawah ini termasuk kedalam task karena pada programnya ini dijalankan secara langsung dan default sebagai keterangan

```

1 self.title = addTitle("Panda3D: Tutorial - Bump Mapping")
2 self.inst1 = addInstructions(0.06, "Press ESC to exit")
3 self.inst2 = addInstructions(0.12, "Move mouse to rotate camera")
4 self.inst3 = addInstructions(0.18, "Left mouse button: Move forwards")
5 self.inst4 = addInstructions(0.24, "Right mouse button: Move backwards")
6 self.inst5 = addInstructions(0.30, "Enter: Turn bump maps Off")
7
8
9
10 self.room = loader.loadModel("models/abstractroom")
11 self.room.reparentTo(render)

```

- b. Kemudian ada task lagi yang digunakan untuk menjalankan sebuah teks dan kemudian akan dijalankan saat frame dijalankan secara default langsung dijalankan

[illegible]

- c. Dibawah ini adalah masuk ke dalam task karena digunakan untuk render sebuah model, light, dan posisi yang kemudian langsung dijalankan saat frame ini di running secara default

```
1 self.lightpivot = render.attachNewNode("lightpivot")
2 self.lightpivot.setPos(0, 0, 25)
3 self.lightpivot.hprInterval(10, LPoint3(360, 0, 0)).loop()
4 plight = PointLight('plight')
5 plight.setColor((1, 1, 1, 1))
6 plight.setAttenuation(LVector3(0.7, 0.05, 0))
7 plnp = self.lightpivot.attachNewNode(plight)
8 plnp.setPos(45, 0, 0)
9 self.room.setLight(plnp)
10
11
12 alight = AmbientLight('alight')
13 alight.setColor((0.2, 0.2, 0.2, 1))
14 alnp = render.attachNewNode(alight)
15 self.room.setLight(alnp)
16
17
18 sphere = loader.loadModel("models/icosphere")
19 sphere.reparentTo(plnp)
```

- d. Pada code dibawah ini adalah task juga karena ia langsung dijalankan saat frame di running atau defaultnya dijalankan

```
1 def setMouseBtn(self, btn, value):
2     self.mousebtn[btn] = value
3
4 def rotateLight(self, offset):
5     self.lightpivot.setH(self.lightpivot.getH() + offset * 20)
6
7 def rotateCam(self, offset):
8     self.heading = self.heading - offset * 10
```

- e. Kemudian yang terakhir ini juga masuk ke task karena digunakan untuk menambahkan tugas

```
1 taskMgr.add(self.controlCamera, "camera-task")
```

2. Event Handling

- a. Pada bagian dibawah ini termasuk kedalam Event Handling karena program dibawah ini akan menjalankan perintah yang sudah ditentukan ketika mendapatkan sebuah inputannya untuk menjalankannya

```
1 self.accept("escape", sys.exit, [0])
2 self.accept("mouse1", self.setMouseBtn, [0, 1])
3 self.accept("mouse1-up", self.setMouseBtn, [0, 0])
4 self.accept("mouse2", self.setMouseBtn, [1, 1])
5 self.accept("mouse2-up", self.setMouseBtn, [1, 0])
6 self.accept("mouse3", self.setMouseBtn, [2, 1])
7 self.accept("mouse3-up", self.setMouseBtn, [2, 0])
8 self.accept("enter", self.toggleShader)
9 self.accept("j", self.rotateLight, [-1])
10 self.accept("k", self.rotateLight, [1])
11 self.accept("arrow_left", self.rotateCam, [-1])
12 self.accept("arrow_right", self.rotateCam, [1])
```

- b. Kemudian pada bagian dibawah ini kedalam Event Handling yang dimana bagian dibawah ini menjalankan suatu perintah yang sudah ditentukan saat mendapatkan sebuah inputan untuk menjalankannya

```
1 def toggleShader(self):
2     self.inst5.destroy()
3     if (self.shaderenable):
4         self.inst5 = addInstructions(0.30, "Enter: Turn bump maps On")
5         self.shaderenable = 0
6         self.room.setShaderOff()
7     else:
8         self.inst5 = addInstructions(0.30, "Enter: Turn bump maps Off")
9         self.shaderenable = 1
10        self.room.setShaderAuto()
```

- c. Yang terakhir ini juga termasuk Event Handling karena dalam hal ini digunakan untuk menjalankan sebuah function atau fungsi yang pergerakannya dari hal yang sudah dideklarasikan sebelumnya

```
1  def controlCamera(self, task):
2
3      md = self.win.getPointer(0)
4      x = md.getX()
5      y = md.getY()
6      if self.win.movePointer(0, 100, 100):
7          self.heading = self.heading - (x - 100) * 0.2
8          self.pitch = self.pitch - (y - 100) * 0.2
9      if self.pitch < -45:
10         self.pitch = -45
11     if self.pitch > 45:
12         self.pitch = 45
13     self.camera.setHpr(self.heading, self.pitch, 0)
14     dir = self.camera.getMat().getRow3(1)
15     elapsed = task.time - self.last
16     if self.last == 0:
17         elapsed = 0
18     if self.mousebtn[0]:
19         self.focus = self.focus + dir * elapsed * 30
20     if self.mousebtn[1] or self.mousebtn[2]:
21         self.focus = self.focus - dir * elapsed * 30
22     self.camera.setPos(self.focus - (dir * 5))
23     if self.camera.getX() < -59.0:
24         self.camera.setX(-59)
25     if self.camera.getX() > 59.0:
26         self.camera.setX(59)
27     if self.camera.getY() < -59.0:
28         self.camera.setY(-59)
29     if self.camera.getY() > 59.0:
30         self.camera.setY(59)
31     if self.camera.getZ() < 5.0:
32         self.camera.setZ(5)
33     if self.camera.getZ() > 45.0:
34         self.camera.setZ(45)
```

3. Penjelasan Alur Task dan Event Handling

Pada Task dan Event Handling ini yang dimana berdasarkan taskmgr, Task ini akan dijalankan secara menyeluruh saat program yang dibuat ini dijalankan secara otomatis yang berdasarkan pada class dan inisialisasi pada code tersebut tanpa terkecuali. Sedangkan Event Handling sendiri akan dijalankan oleh program jika ada konfirmasi dari pengguna yang cocok dengan program event yang sudah disetting sebelumnya dan juga jika program ini sesuai maka akan menjalankan suatu code.