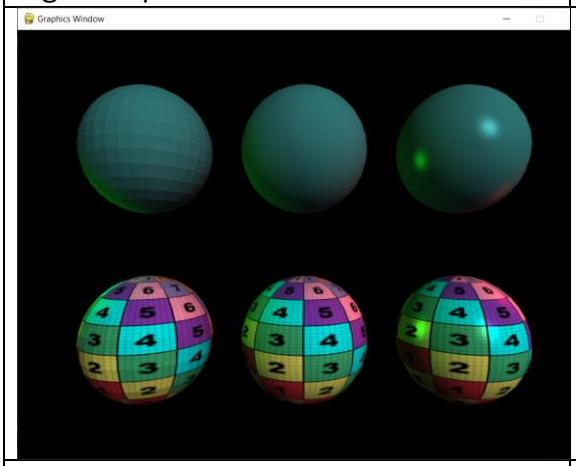
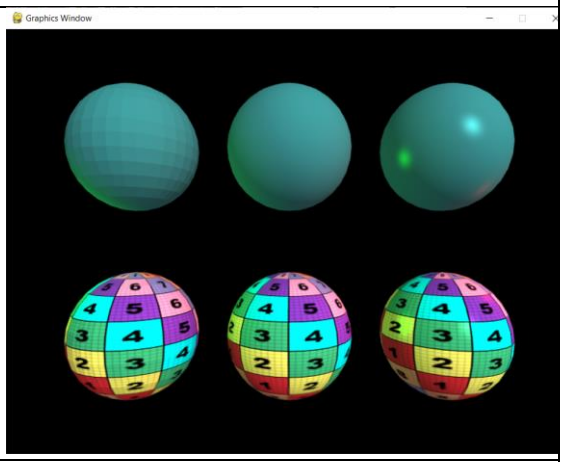
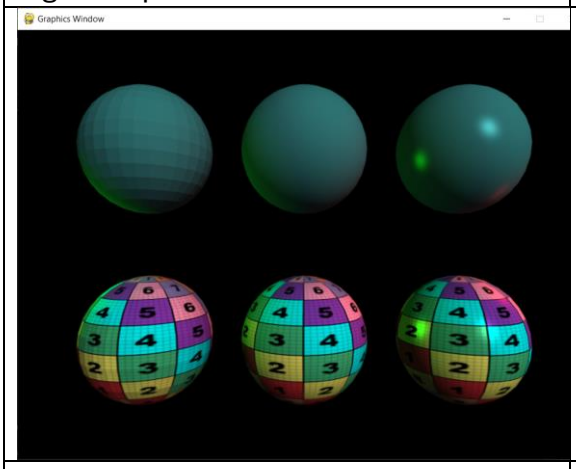
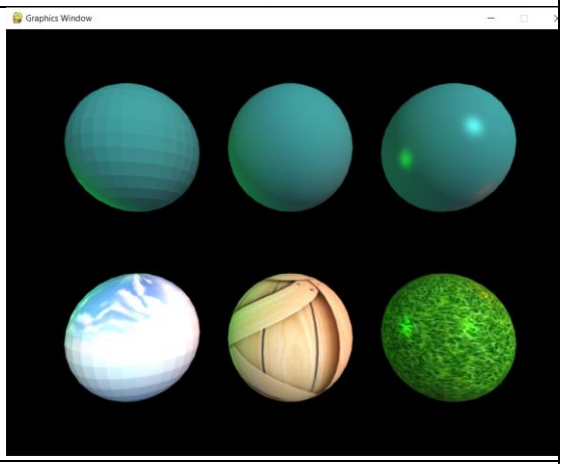
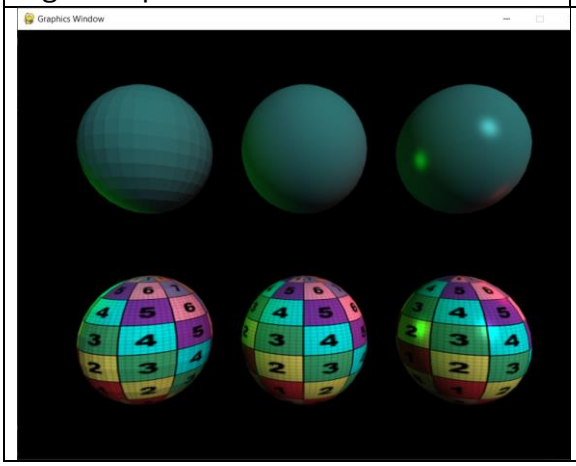
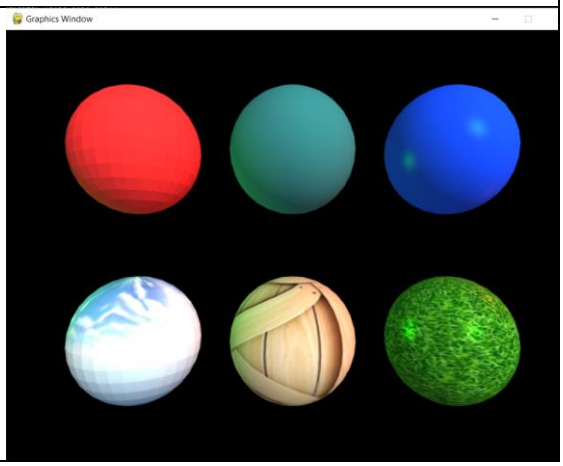
Miguel Silva 80072

P10-1

Lighted spheres	P10-1 A
	
Lighted spheres	P10-1 B
	
Lighted spheres	P10-1 C
	

Explicação das alterações:

1. (a) Aumento da luz ambiente
    - No código original, a luz ambiente era definida como AmbientLight(color=[0.1, 0.1, 0.1]).
    - Na alteração, passou a ser AmbientLight(color=[0.5, 0.5, 0.5]), o que significa que a intensidade da luz ambiente foi aumentada de 0.1 para 0.5 (5 vezes mais luz ambiente).

```
1. ambient_light = AmbientLight(color=[0.5, 0.5, 0.5])
```

2. (b) Mudança das texturas na fila de baixo
    - Originalmente, as três esferas da fila de baixo utilizavam todas a textura grid.jpg.
    - Na alteração, cada uma das esferas da parte inferior ganhou uma textura diferente: “sky.jpg”, “crate.jpg” e “grass.jpg”. Assim, cada material (FlatMaterial, LambertMaterial, e PhongMaterial) deixou de usar a grid.jpg e passou a usar uma das novas texturas.

```
1.          # lighted materials with a texture
2.          textured_flat_material = FlatMaterial(
3.              texture=Texture("images/sky.jpg"),
4.              number_of_light_sources=4
5.          )
6.          textured_lambert_material = LambertMaterial(
7.              texture=Texture("images/crate.jpg"),
8.              number_of_light_sources=4
9.          )
10.         textured_phong_material = PhongMaterial(
11.             texture=Texture("images/grass.jpg"),
12.             number_of_light_sources=4
13.         )
```

3. (c) Mudança na cor dos materiais esquerdo e direito da fila de cima.

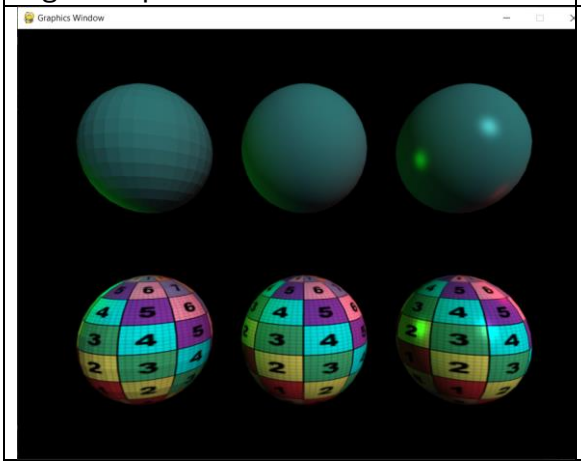
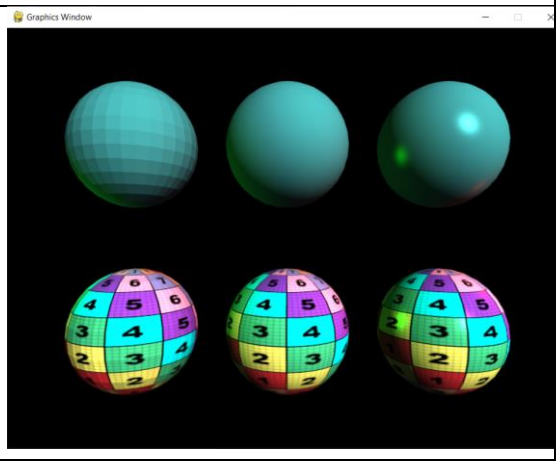
    Foi modificado foi o valor atribuído a baseColor em dois dos materiais:
    - Para o *FlatMaterial*, a propriedade foi alterada para [1.0, 0.2, 0.2], o que gera um tom de vermelho na esfera correspondente (esfera superior esquerda).
    - Para o *PhongMaterial*, a propriedade passou a ser [0.1, 0.3, 1.0], o que resulta num tom de azul na esfera correspondente (esfera superior direita).

O *LambertMaterial* manteve o valor [0.2, 0.5, 0.5], de forma a que a esfera do meio se mantivesse com a mesma cor. Estas alterações fazem com que, na fila

superior, a bola da esquerda seja vermelha e a da direita azul, conforme pretendido.

```
      # lighted materials with a color
        flat_material = FlatMaterial(
            property_dict={"baseColor": [1.0, 0.2, 0.2]},  # Tom vermelho
            number_of_light_sources=4
        )
        lambert_material = LambertMaterial(
            property_dict={"baseColor": [0.2, 0.5, 0.5]}, # Igual
            number_of_light_sources=4
        )
        phong_material = PhongMaterial(
            property_dict={"baseColor": [0.1, 0.3, 1.0]},  # Tom azul
            number_of_light_sources=4
        )
```

P10-2

Lighted spheres	P10-2
	

```
1. directional_light = DirectionalLight(color=[1.5, 1.5, 1.5], direction=[-1, -1, -2])
```

Aumentei a intensidade da luz direcional ao alterar os seus valores de cor de [0.8, 0.8, 0.8] para [1.5, 1.5, 1.5].

O que acontece:

1. A luz direcional fica significativamente mais brilhante, pois a sua intensidade foi aumentada para quase o dobro (de 0.8 para 1.5).
2. Este aumento de intensidade afeta todos os objetos na cena.
3. As esferas aparecem muito mais brilhantes no lado voltado para a luz direcional (que provém da direção [-1, -1, -2]).
4. As esferas com material Phong exibem realces especulares mais pronunciados (os pontos brilhantes que simulam a reflexão).

P10-3

Importei a biblioteca math,
Defini as variáveis para a rotação:

```
1.    # Camera rotation variables
2.         self.rotation_angle = 0
3.         self.rotation_speed = 0.5  # radians per second
4.         self.orbit_radius = 8      # distance from center
```

- rotation_angle da track ao ângulo de rotação atual
- rotation_speed controla quão rápido a camera orbita
- orbit_radius define a distância do centro

E no método update fiz:

```
 1. # Update rotation angle
 2. self.rotation_angle += self.rotation_speed * self.delta_time
 3. # Calculate new camera position
 4. x = self.orbit_radius * math.sin(self.rotation_angle)
 5. z = self.orbit_radius * math.cos(self.rotation_angle)
 6. # Update camera position
 7. self.rig.set_position([x, 0, z])
 8. # Make camera look at center
 9. self.rig.look_at([0, 0, 0])
10. # Render the scene
```

Em que:

- Atualizo o ângulo baseado no tempo que passou
- Calculo *x* e *z* utilizando respetivamente o seno e coseno para criar o movimento circular
- Defino a posição da camera para as novas coordenadas
- E depois a camera aponta para o centro

P10-4

1. Adicionado uma nova luz direcional azul:

```
1. self.blue_directional_light = DirectionalLight(color=[0, 0, 0.8], direction=[1, 0, 0])
2. self.scene.add(self.blue_directional_light)
```

2. Aumentado o número de fontes de luz de 3 para 4 em todos os construtores de materiais:

```
1. flat_material = FlatMaterial(
2. property_dict={"baseColor": [0.2, 0.5, 0.5]},
3. number_of_light_sources=4
4. )
5.
```

(O mesmo se aplica ao lambert_material e phong_material)

3. Criado um helper para a luz direcional azul:

```
1. blue_directional_light_helper = DirectionalLightHelper(self.blue_directional_light)
```

4. Posicionado a luz azul na parte inferior da janela:

```
1. self.blue_directional_light.set_position([0, -2, 0])
2. self.blue_directional_light.add(blue_directional_light_helper)
3.
```

5. Adicionado movimento oscilatório para a luz azul:

```
1. self.blue_directional_light.set_direction([1, -math.sin(0.5 * self.time), 0])
```

Isto faz com que a luz azul oscile em fase oposta à luz direcional original.