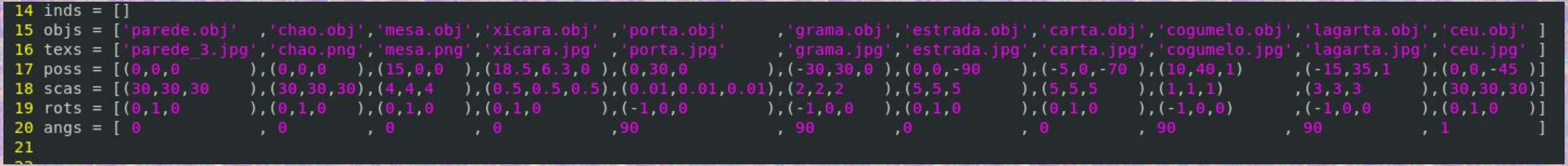
Trabalho Final

Bruno Fernandes Moreira 11218712
Francisco de Freitas Pedrosa 11215699
Savio Duarte Fontes 10737251
Thales Willian Dalvi da Silva 11219196

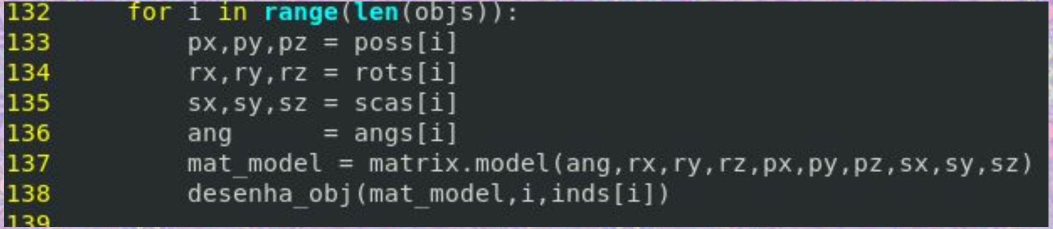
Tema - Alice no País das Maravilhas

Tema - Outras Ideias

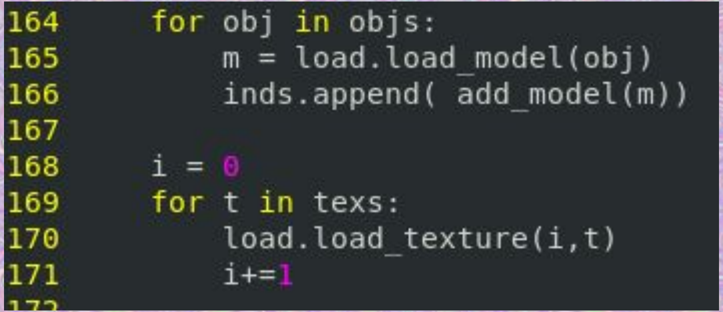
Objetos



```
inds = []
objs = ['parede.obj'  ,'chao.obj','mesa.obj','xicara.obj' ,'porta.obj'     ,'grama.obj','estrada.obj','carta.obj','cogumelo.obj','lagarta.obj','ceu.obj' ]
texs = ['parede_3.jpg','chao.png','mesa.png','xicara.jpg' ,'porta.jpg'     ,'grama.jpg','estrada.jpg','carta.jpg','cogumelo.jpg','lagarta.jpg','ceu.jpg' ]
poss = [(0,0,0       ),(0,0,0   ),(15,0,0  ),(18.5,6.3,0 ),(0,30,0        ),(-30,30,0 ),(0,0,-90    ),(-5,0,-70 ),(10,40,1)     ,(-15,35,1   ),(0,0,-45 )]
scas = [(30,30,30    ),(30,30,30),(4,4,4   ),(0.5,0.5,0.5),(0.01,0.01,0.01),(2,2,2    ),(5,5,5      ),(5,5,5    ),(1,1,1)       ,(3,3,3      ),(30,30,30)]
rots = [(0,1,0       ),(0,1,0   ),(0,1,0   ),(0,1,0      ),(-1,0,0        ),(-1,0,0   ),(0,1,0      ),(0,1,0    ),(-1,0,0)      ,(-1,0,0     ),(0,1,0   )]
angs = [ 0            , 0        , 0        , 0           ,90              , 90        ,0           , 0         , 90           , 90          , 1         ]

```

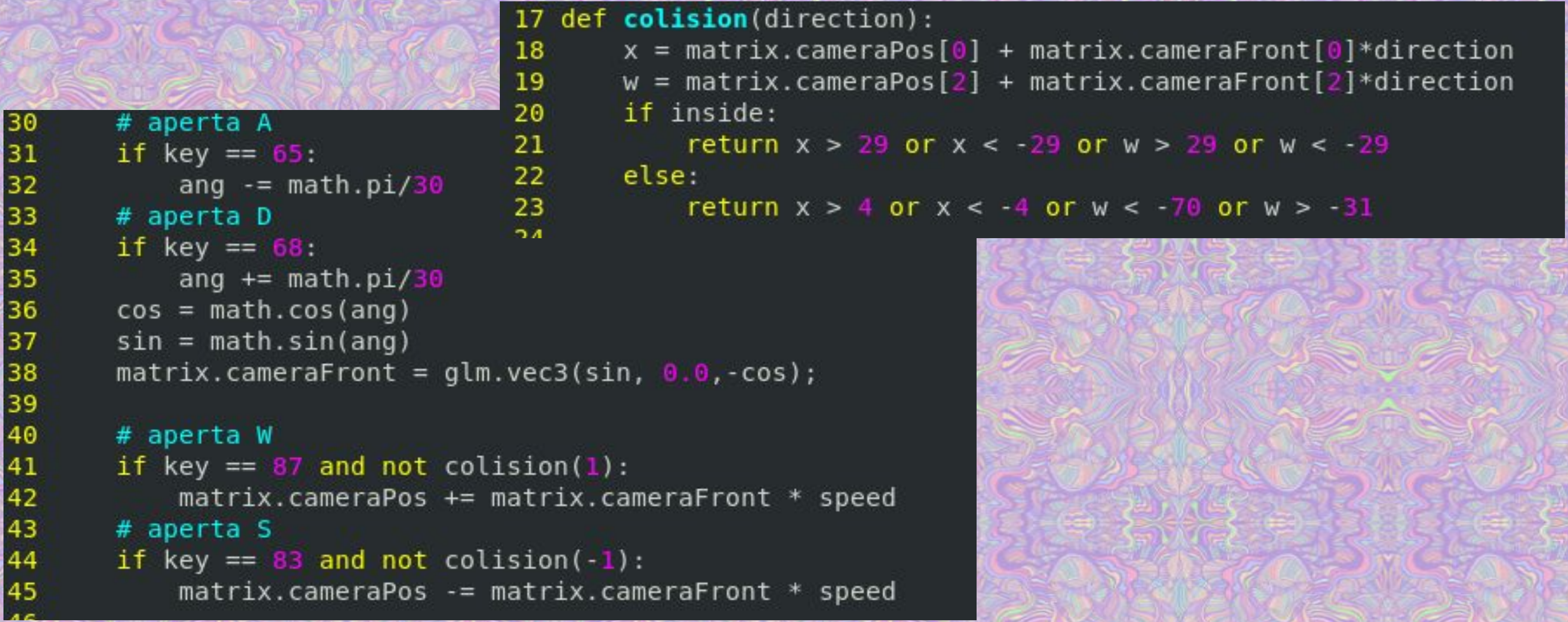


```
    for i in range(len(objs)):
        px,py,pz = poss[i]
        rx,ry,rz = rots[i]
        sx,sy,sz = scas[i]
        ang      = angs[i]
        mat_model = matrix.model(ang,rx,ry,rz,px,py,pz,sx,sy,sz)
        desenha_obj(mat_model,i,inds[i])
```



```
    for obj in objs:
        m = load.load_model(obj)
        inds.append( add_model(m))

    i = 0
    for t in texs:
        load.load_texture(i,t)
        i+=1
```

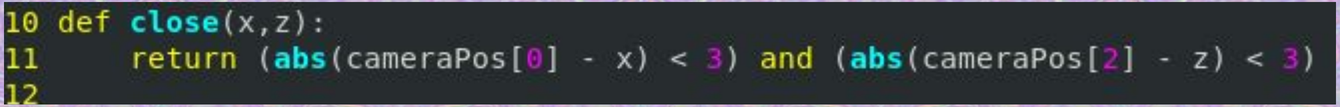
Movimentação



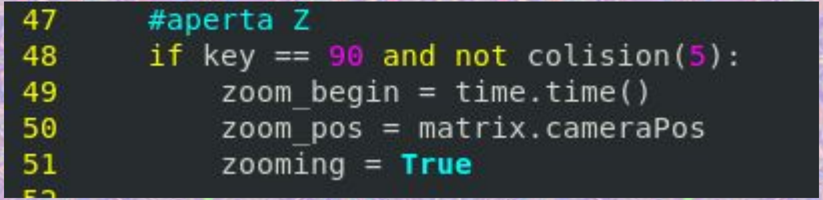
```python
def colision(direction):
    x = matrix.cameraPos[0] + matrix.cameraFront[0]*direction
    w = matrix.cameraPos[2] + matrix.cameraFront[2]*direction
    if inside:
        return x > 29 or x < -29 or w > 29 or w < -29
    else:
        return x > 4 or x < -4 or w < -70 or w > -31
```

```python
    # aperta A
    if key == 65:
        ang -= math.pi/30
    # aperta D
    if key == 68:
        ang += math.pi/30
    cos = math.cos(ang)
    sin = math.sin(ang)
    matrix.cameraFront = glm.vec3(sin, 0.0,-cos);

    # aperta W
    if key == 87 and not colision(1):
        matrix.cameraPos += matrix.cameraFront * speed
    # aperta S
    if key == 83 and not colision(-1):
        matrix.cameraPos -= matrix.cameraFront * speed
```

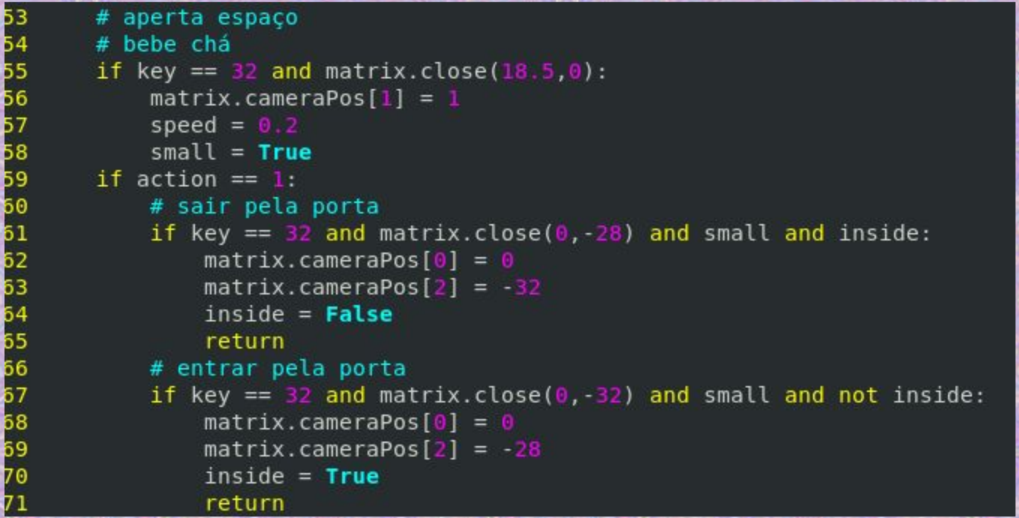
Eventos



```python
def close(x,z):
    return (abs(cameraPos[0] - x) < 3) and (abs(cameraPos[2] - z) < 3)

```

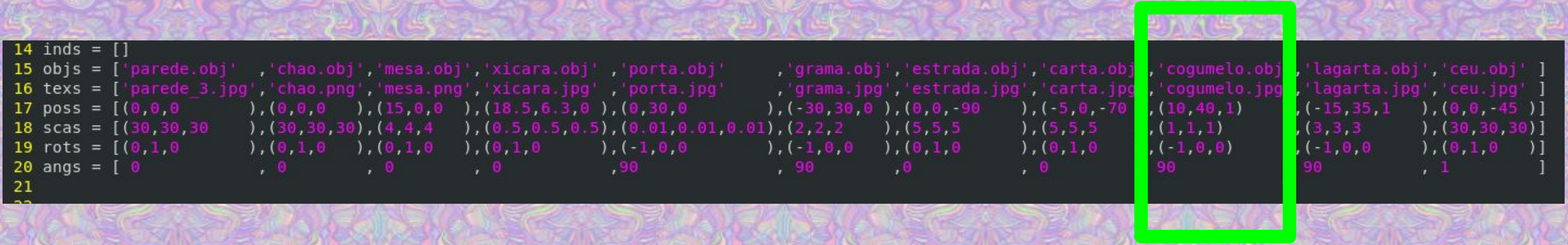


```python
    #aperta Z
    if key == 90 and not colision(5):
        zoom_begin = time.time()
        zoom_pos = matrix.cameraPos
        zooming = True
```

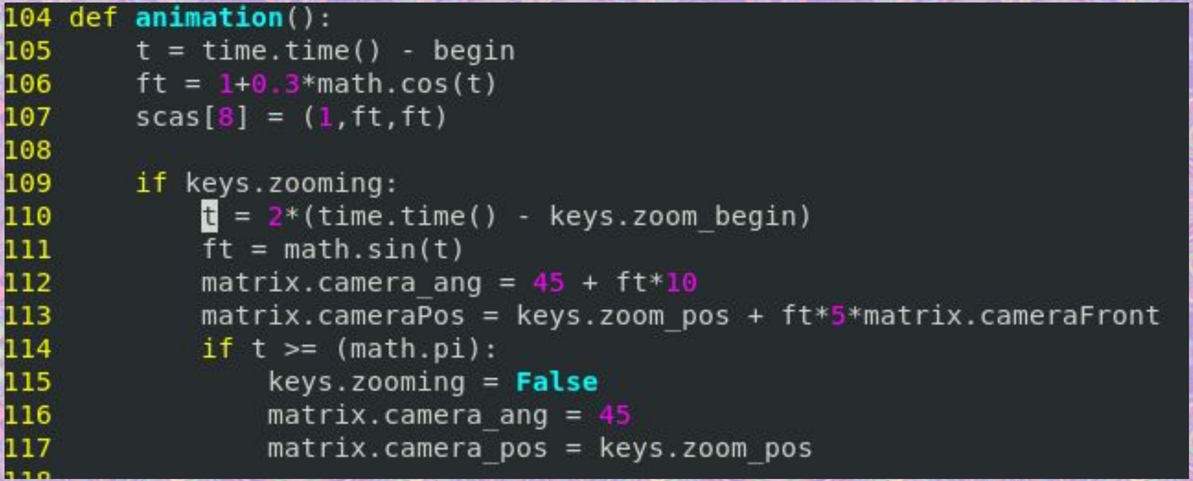


```python
    # aperta espaço
    # bebe chá
    if key == 32 and matrix.close(18.5,0):
        matrix.cameraPos[1] = 1
        speed = 0.2
        small = True
    if action == 1:
        # sair pela porta
        if key == 32 and matrix.close(0,-28) and small and inside:
            matrix.cameraPos[0] = 0
            matrix.cameraPos[2] = -32
            inside = False
            return
        # entrar pela porta
        if key == 32 and matrix.close(0,-32) and small and not inside:
            matrix.cameraPos[0] = 0
            matrix.cameraPos[2] = -28
            inside = True
            return
```

Animação



```
inds = []
objs = ['parede.obj'  ,'chao.obj','mesa.obj','xicara.obj' ,'porta.obj'      ,'grama.obj','estrada.obj','carta.obj ,'cogumelo.obj ,'lagarta.obj','ceu.obj' ]
texs = ['parede_3.jpg','chao.png','mesa.png','xicara.jpg' ,'porta.jpg'      ,'grama.jpg','estrada.jpg','carta.jpg ,'cogumelo.jpg ,'lagarta.jpg','ceu.jpg' ]
poss = [(0,0,0        ),(0,0,0   ),(15,0,0  ),(18.5,6.3,0 ),(0,30,0         ),(-30,30,0 ),(0,0,-90    ),(-5,0,-70  ,(10,40,1)    ,(-15,35,1   ),(0,0,-45 )]
scas = [(30,30,30     ),(30,30,30),(4,4,4   ),(0.5,0.5,0.5),(0.01,0.01,0.01),(2,2,2    ),(5,5,5      ),(5,5,5     ,(1,1,1)      ,(3,3,3      ),(30,30,30)]
rots = [(0,1,0        ),(0,1,0   ),(0,1,0   ),(0,1,0      ),(-1,0,0         ),(-1,0,0   ),(0,1,0      ),(0,1,0     ,(-1,0,0)     ,(-1,0,0     ),(0,1,0   )]
angs = [ 0            , 0        , 0        , 0           ,90               , 90        ,0            , 0           90            90           , 1         ]

```



```
def animation():
    t = time.time() - begin
    ft = 1+0.3*math.cos(t)
    scas[8] = (1,ft,ft)

    if keys.zooming:
        t = 2*(time.time() - keys.zoom_begin)
        ft = math.sin(t)
        matrix.camera_ang = 45 + ft*10
        matrix.cameraPos = keys.zoom_pos + ft*5*matrix.cameraFront
        if t >= (math.pi):
            keys.zooming = False
            matrix.camera_ang = 45
            matrix.camera_pos = keys.zoom_pos
```

Obrigado!