from direct.showbase.ShowBase import ShowBase

from direct.task import Task

from panda3d.core import WindowProperties, Vec3, LVector3f

class MyApp(ShowBase):

global horPos

global verPos

global forPos

horPos = float(0)

verPos = float(0)

forPos = float(0)

global forward

global backward

global right

global left

global up

global down

forward = False

backward = False

right = False

left = False

up = False

down = False

global moveSpeed

global mouseSensitivity

moveSpeed = float(100)

mouseSensitivity = float(50)

def \_\_init\_\_(self):

ShowBase.\_\_init\_\_(self)

base.disableMouse()

props = WindowProperties()

props.setCursorHidden(True)

props.setMouseMode(WindowProperties.M\_relative)

base.win.requestProperties(props)

#simplepbr.init()

self.accept("w", self.forwardDown)

self.accept("w-up", self.forwardUp)

self.accept("s", self.backwardDown)

self.accept("s-up", self.backwardUp)

self.accept("a", self.leftDown)

self.accept("a-up", self.leftUp)

self.accept("d", self.rightDown)

self.accept("d-up", self.rightUp)

self.accept("space", self.upDown)

self.accept("space-up", self.upUp)

self.accept("c", self.downDown)

self.accept("c-up", self.downUp)

self.taskMgr.add(self.moveCameraTask, "MoveCameraTask")

self.scene = self.loader.loadModel("models/environment")

self.scene.reparentTo(self.render)

self.scene.setScale(0.25, 0.25, 0.25)

self.scene.setPos(-8, 42, 0)

#base.messenger.toggleVerbose()

def moveCameraTask(self, task):

global horPos

global verPos

global forPos

deltaTime = globalClock.getDt()

if (forward):

forPos += moveSpeed \* deltaTime

if (backward):

forPos -= moveSpeed \* deltaTime

if (left):

horPos -= moveSpeed \* deltaTime

if (right):

horPos += moveSpeed \* deltaTime

if (down):

verPos -= moveSpeed \* deltaTime

if (up):

verPos += moveSpeed \* deltaTime

x = 0

y = 0

if base.mouseWatcherNode.hasMouse():

x = base.mouseWatcherNode.getMouseX()

y = base.mouseWatcherNode.getMouseY()

print("x: " + str(x))

print("y: " + str(y))

self.camera.setPos(horPos, forPos, verPos)

self.camera.setHpr(-x \* mouseSensitivity, y \* mouseSensitivity, 0)

print(-x\*mouseSensitivity)

v = self.camera.getRelativeVector(self.camera, Vec3.forward())

#print(v)

return Task.cont

def forwardDown(self):

global forward

forward = True

def forwardUp(self):

global forward

forward = False

def backwardDown(self):

global backward

backward = True

def backwardUp(self):

global backward

backward = False

def leftDown(self):

global left

left = True

def leftUp(self):

global left

left = False

def rightDown(self):

global right

right = True

def rightUp(self):

global right

right = False

def downDown(self):

global down

down = True

def downUp(self):

global down

down = False

def upDown(self):

global up

up = True

def upUp(self):

global up

up = False

app = MyApp()

app.run()