

Python easy game engine 2D and 3D lerning German

ursina game engine for easy lerning!

Python3

Info

Um ursina zu lernen solltest du dich mit Python gut aus kennen.

Diese Programme benutzen alle kein `class`.

Du brauchst ein `python-env` und ein Debian oder Ubuntu PC.

Info zum Starten des Programms aus dem Terminal

Debian/Ubuntu

python-env

Um aus ein Ordner raus zugehen musste diesen command benutzen:

```
cd ..
```

Um in ein Ordner rein zugehen verwendest du diesen commmand:

```
cd Der_ordner_wo_rein_willst
```

PROF

Um dein Programm zu runen verwändest du diesen command:

```
python3 dateiname.py
```

Terminator ein terminal king

Der **Terminator** ein verbessertes und moderneres terminal!

Terminator Installation

Beim terminal installieren:

Terminal command:

```
sudo apt update  
sudo apt install terminator
```

Link:

[GitHub - gnome-terminator](#)

Drücke auf code und im kleinen kasten steht download zip drück darauf.

Gehe auf dowloads drücke auf das zip rechts click drücke jetzt auf mit anderen anwendungen öffnen
öffne das programm mit deiner Software.

Ursina installation

Debian/Ubuntu

python-env

Terminal command:

```
pip install panda3d
```

Terminal command:

```
pip install ursina
```

Terminal command:

```
pip install ursina packages
```

PROF

Fehler?

Falls du kein Problem bei der Installation hattest kannst du diesen Teil überspringen!

Debian/Ubuntu

python-env

Terminal command:

```
sudo apt install pip
```

Terminal command:

```
sudo apt upgrade pip
```

Terminal command:

```
pip install panda3d
```

Terminal command:

```
pip install ursina
```

Terminal command:

```
pip install ursina packages
```

Ursina's Geschichte

Die **Ursina** Engine wurde ab ca. **2019** vom Entwickler **@pokepetter** *ein norwegischer Entwickler* ins Leben gerufen. Ziel war es, eine einfache und leicht zugängliche **3D-Game-Engine** und auch **2D-Game-Engine** in **Python** zu schaffen. Die Engine sollte auch für Einsteiger geeignet sein und die Erstellung von **3D/2D-Games** mit minimalem Code ermöglichen.

Mit ursina 2D games

Ursina importieren!

```
from ursina import *
```

So importierst du ursina. Jetzt willst du natürlich wissen wie man ein bildschirm auf machen kann!

```
from ursina import Ursina

app = Ursina()

app.run()
```

Das Game ist noch nicht 2D um das zu machen muss man was kleines importieren und hinzufügen!

```

from ursina import Ursina
from ursina import camera

app = Ursina()
camera.orthographic = True
camera.fov = 10

app.run()

```

Das wichtigste rolle bei **2D/3D-Games** spielt **def update** aber zu erst der player fange wir mal an!

```

from ursina import Ursina
from ursina import camera
from ursina import Entity

app = Ursina()
camera.orthographic = True
camera.fov = 10
player = Entity(model='cube', color=color.black, scale=(2.5), position=(0, -4), collider='box')

app.run()

```

Das ist der player jetzt müssen wir seine bewungen mit in **def update** rein machen!

```

from ursina import Ursina
from ursina import camera
from ursina import Entity
from ursina import held_keys
from ursina import time

app = Ursina()
camera.orthographic = True
camera.fov = 10
player = Entity(model='cube', color=color.black, scale=(2.5), position=(0, -4), collider='box')

def update():
    if held_keys['a']:
        player.x -= 15 * time.dt
    if held_keys['d']:
        player.x += 15 * time.dt

```

```
app.run()
```

Mit ursina Minecraft

Fangen wir mit einen Sky an!

```
from ursina import Ursina
from ursina import Sky

app = Ursina()
Sky()

app.run()
```

Jetzt hast du schon ein Tages Himmml.Wenn du ein Sonnen untergang haben willst musst du noch etwas kleines hinzufügen was ich machen werde!

```
from ursina import Ursina
from ursina import Sky

app = Ursina()
Sky(texture="sky_sunset")

app.run()
```

Du hast jetzt eine Sonnen untergang texture.Doch du brauchst noch ein player den du leicht importieren kannst.

```
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import Ursina
from ursina import Sky

app = Ursina()
player = FirstPersonController
Sky(texture='sky_sunset')

app.run()
```

Nun hast du auch den player importiert.Du brauchst die bewegungen nicht zu definieren nachmlich die wurden auch importiert.

Jetzt kommen wir zum schwierigen teil die Blöcke die in ursina cube gennant werden.

```
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import Ursina
from ursina import Button
from ursina import Sky
from ursina import scene

app = Ursina()
player = FirstPersonController
boxes = []
Sky(texture='sky_sunset')

def add_box(position, color):
    boxes.append(
        Button(
            parent=scene, #ursina muss wissen wo man die box sieht
            model='cube',
            origin=0.5,
            color=color,
            position=position,
        )
    )

app.run()
```

Jetzt sieht der code nicht so bunt aus aber bei Vs code oder Neovim schon. Hier habe ich eine erklärungs Zeile geschriben die mit # anfängt die brauchst du nicht. Gerade fällst du immer noch also muss unter dir eine plate sein die wir jetzt machen

PROF

```
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import Ursina
from ursina import Button
from ursina import Sky
from ursina import color
from ursina import scene

app = Ursina()
player = FirstPersonController
boxes = []
Sky(texture='sky_sunset')
player.position = 3, 20, 3

def add_box(position, color):
    boxes.append(
```

```

        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position
        )
    )

for u in range(20):
    for ui in range(20):
        add_box(position=(u, 0, ui), color=color.green)

app.run()

```

Es kann sein das du nach den fall in einer box bags du musst nur 2/3 mal auf Space drücken, das sind nur 33 Zeile.

Nun fehlen zwei sachen plazieren und abauen hier braucht man zwei sachen `def update():` und `def input(key):`. Fangen wir mit update das werden wir nicht benötigen, trotzdem will ich das euch erklären `def update():` ist kein normaler def, was du in update rein schreibst wider holt sich vortlaufent bis das programm gestoppt wird wie eine `while True` schleife und input ist das gleiche in kommplizierter. Weil wir update nicht benötigen müssen wir nicht viel darein schreiben.

```

from ursina import Sky
from ursina import Ursina
from ursina import Button
from ursina import color
from ursina import destroy
from ursina import scene
from ursina import mouse
from ursina.prefabs.first_person_controller import FirstPersonController

app = Ursina()
player = FirstPersonController()
player.position = 2, 20, 2
Sky(texture='sky_sunset')

def add_box(position, color):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position
        )
    )

```

```

    )

for u in range(20):
    for ui in range(20):
        add_box(position=(u, 0, ui), color=color.green)

def update():
    pass

def input(key):
    global a, d, ddd, cv, t
    for box in boxes:
        if box.hovered:
            if key == "right mouse down":
                add_box(box.position + mouse.normal, color=color.gray)
            if key == "left mouse down":
                boxes.remove(box)
                destroy(box)

app.run()

```

Jetzt hast du Schwierigkeiten beim Zumachen des Programmes, aber natürlich habe ich auch dafür eine Lösung.

```

from ursina import Sky
from ursina import Ursina
from ursina import Button
from ursina import color
from ursina import destroy
from ursina import scene
from ursina import mouse
from ursina import held_keys
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import application

app = Ursina()
player = FirstPersonController()
player.position = 2, 20, 2
Sky(texture='sky_sunset')

def add_box(position, color):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,

```



```

        position=position
    )
)

for u in range(20):
    for ui in range(20):
        add_box(position=(u, 0, ui), color=color.green)

def update():
    if held_keys['c']:
        application.quit()

def input(key):
    global a, d, ddd, cv, t
    for box in boxes:
        if box.hovered:
            if key == "right mouse down":
                add_box(box.position + mouse.normal, color=color.gray)
            if key == "left mouse down":
                boxes.remove(box)
                destroy(box)

app.run()

```

Wenn du c drückst schließt das programm sich von allein. Wenn du willst das man die farbe der box die man plaziert wechseln kann musst zwei listen definieren, dafür musst du text importieren das geht so:

```

from ursina import Text

```

So kannst du text importieren. Das programm sieht aktuell so aus:

```

from ursina import Sky
from ursina import Ursina
from ursina import Button
from ursina import color
from ursina import destroy
from ursina import scene
from ursina import mouse
from ursina import held_keys
from ursina import Text
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import application

app = Ursina()
player = FirstPersonController()
Sky(texture='sky_sunset')

```

```

list = [color.red, color.green, color.white, color.black,
        color.gray, color.blue, color.brown]
list_draw = ["red", "green", "white", "black", "gray", "blue", "brown"]
d = 0
a = list[d]
ddd = list_draw[d]
b_b = False
boxes = []
g = 1
b_b_text = Text(' ', origin=(0,7), scale=2)

def add_box(position, color):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position
        )
    )

for _ in range(2):
    g -= 1
    for x in range(51):
        for y in range(51):
            add_box( (x, g, y), color=color.green)

def update():
    if ((held_keys['l']) or (held_keys['r'])):
        ddd = list_draw[d]
        b_b_text.text = ddd
    if held_keys['c']:
        application.quit()

def input(key):
    global a, d
    for box in boxes:
        if box.hovered:
            if key == "right mouse down":
                add_box(box.position + mouse.normal, color=a)
            if key == "left mouse down":
                boxes.remove(box)
                destroy(box)
            if key == "l":
                if d == 0:
                    d = 6
                else:
                    d -= 1
            a = list[d]

```

```

        ddd = list_draw[d]
    if key == "r":
        if d == 6:
            d = 0
        else:
            d += 1
    a = list[d]
    ddd = list_draw[d]

app.run()

```

Jetzt brauchst du ein respawn point wo du mit einen Button hin kannst *zum beispiel* ich benutze `m` als Button. Jetzt fangen wir mal mit der andärung an.

```

from ursina import Sky
from ursina import Ursina
from ursina import Button
from ursina import color
from ursina import destroy
from ursina import scene
from ursina import mouse
from ursina import held_keys
from ursina import Text
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import application

app = Ursina()
player = FirstPersonController()
Sky(texture='sky_sunset')
list = [color.red, color.green, color.white, color.black,
        color.gray, color.blue, color.brown]
list_draw = ["red", "green", "white", "black", "gray", "blue", "brown"]
d = 0
a = list[d]
ddd = list_draw[d]
b_b = False
boxes = []
g = 1
b_b_text = Text('', origin=(0,7), scale=2)

def add_box(position, color):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,

```

```

        position=position
    )
)

for _ in range(2):
    g -= 1
    for x in range(51):
        for y in range(51):
            add_box( (x, g, y), color=color.green)

def update():
    if ((held_keys['l']) or (held_keys['r'])):
        ddd = list_draw[d]
        b_b_text.text = ddd
    if held_keys['c']:
        application.quit()
    if held_keys['m']:
        player.position = 15, 100, 15

def input(key):
    global a, d
    for box in boxes:
        if box.hovered:
            if key == "right mouse down":
                add_box(box.position + mouse.normal, color=a)
            if key == "left mouse down":
                boxes.remove(box)
                destroy(box)
            if key == "l":
                if d == 0:
                    d = 6
                else:
                    d -= 1
                a = list[d]
                ddd = list_draw[d]
            if key == "r":
                if d == 6:
                    d = 0
                else:
                    d += 1
                a = list[d]
                ddd = list_draw[d]

app.run()

```

PROF

Jetzt kanns sein das du nach dem sturz in einem Block bags um raus aus dem Block zu kommen musst du 2/3 mal space drücken. Jetzt kannst du das ganze mit texturen machen!

Here is programm with textures:

```
from ursina import Sky
from ursina import Ursina
from ursina import Button
from ursina import color
from ursina import Text
from ursina import held_keys
from ursina import destroy
from ursina import scene
from ursina import mouse
from ursina.prefabs.first_person_controller import FirstPersonController
from ursina import application

app = Ursina()
player = FirstPersonController()
player.position = 2, 20, 2
Sky(texture='sky_sunset')
list = ['der.jpeg', 'retro.png', 'cry.png', 'obs.png', 'glo.jpg',
        'bri.jpg', 'red.jpg', 'ingo.png', 'stein1.png', 'stein2.png',
        'tr1.png', 'buer.jpg', 'ft.png', 'rt.jpg',
        'tatata.jpg', 'ste.jpg', 'derr.jpg']
list_draw = ["dirt", "Retro Obsidian", "Crying Obsidian",
             "Obsidian", "glow stone", "Blackstone", "Redstone", "silver erz",
             "Stone", "Kobbel Stone", "Glas", "Zigel",
             "end stone", "wood", "Nether stone", "snow", "Sand"]
d = 0
t = 0
cv = [color.white, color.white66]
wer = []
a = list[d]
ddd = list_draw[d]
boxes = []
g = 1
b_b_text = Text('', origin=(0,7), scale=2)

def add_box(position, color, texture):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture=texture
        )
    )

for ioo in range(2):
    g = ioo + 1
```

```

for ou in range(51):
    x = ou + 1
    for hu in range(51):
        y = hu + 1
        add_box( (x, g, y), color=color.white, texture='ste.jpg')

def update():
    global d, ddd, list_draw
    if held_keys['c']:
        application.quit()
    if held_keys['m']:
        player.position = 0, 17, 0
    if ((held_keys['l']) or (held_keys['r'])):
        ddd = list_draw[d]
        b_b_text.text = ddd

def input(key):
    global a, d, ddd, cv, t
    for box in boxes:
        if box.hovered:
            if key == "right mouse down":
                if a == 'tr1.png':
                    t = 1
                else:
                    t = 0
            add_box(box.position + mouse.normal, color=cv[t],
texture=a)
            if key == "left mouse down":
                boxes.remove(box)
                destroy(box)
        if key == "l":
            if d == 0:
                d = len(list)-1
            else:
                d -= 1
            a = list[d]
        if key == "r":
            if d == len(list)-1:
                d = 0
            else:
                d += 1
            a = list[d]

app.run()

```

PROF

Die pngs/jpgs/jpegs musst du dir selber aus dem Internet raussuchen ein Beispiel:

minecraft Blockname texture

So kannst du dir die images rausuchen und du musst meine image namen zu deinen tauschen. Dieses programm hat 2 Monate gedauert also unterschätze es nicht. Man kann Grass nicht anwenden, weil er mehr als 1 image verfügt.

Jetzt erstelle ein neue python datei ich habe diese datei **Lucky_block_battle.py** genannt aber du kannst sie auch anders nennen nämlich jetzt definieren wir **add_Lucky**. Das andere programm kannst du mit copy(Ctrl+C) und paste(Ctrl+V) in dieses programm copieren. Nun müssen wir das programm ein bisschen verändern!

```
from ursina import Sky
from ursina import Ursina
from ursina import Text
from ursina import color
from ursina import Button
from ursina import destroy
from ursina import mouse
from ursina import scene
from ursina.prefabs.first_person_controller import FirstPersonController
from random import randint

app = Ursina()
player = FirstPersonController()
eur = 0
liste = ['luck.png', 'wat.jpg']
io = liste[randint(0, 1)]
Sky(texture='sky_sunset')
list = ['retro.png', 'cry.png', 'obs.png', 'glo.jpg',
        'bri.jpg', 'red.jpg', 'ingo.png']

noob = ['stein1.png', 'stein2.png', 'tr1.png',
        'buer.jpg', 'der.jpeg', 'ft.png',
        'rt.jpg', 'tatata.jpg', 'ste.jpg', 'derr.jpg']

d = 0
aa = 0
bb = 0
tvee = " "
ui = 0
cc = 0
lucky = []
z = 0
mixes = []
aaaa = 0
t = 0
n = 0
a = 0
ur = 0
cv = [color.white, color.white66]
wer = []
boxes = []
g = 1
d_text = Text('', origin=(0, -10), scale=1)
```

```
b_text = Text('', origin=(0, -11), scale=1)
player.position = 8, 100, 0
```

```
def add_box(position, color, texture):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture=texture
        )
    )
```

```
def add_mix(position, texture):
    mixes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color.white,
            position=position,
            texture=texture
        )
    )
```

```
def add_lucky(position, color, texture):
    lucky.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture=texture
        )
    )
```

```
def add_lu(position, color):
    wer.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture='lu.png'
        )
    )
```


)

```
n = -21
ccc = -6
for _ in range(20):
    n += 1
    ccc = -6
    for _ in range(27):
        ccc += 1
        add_box(position=(ccc, 0, n), color=color.white,
texture='retro.png')
```

```
n = -1
for _ in range(150):
    n += 2
    add_lu(position=(15, 1, n), color=color.white)
```

```
n = -1
for _ in range(150):
    n += 2
    a = ['mix.png', 'fant.png']
    add_mix(position=(8, 1, n), texture=a[randint(0, 1)])
```

```
n = -1
for _ in range(150):
    n += 2
    io = liste[randint(0, 1)]
    add_lucky(position=(1, 1, n), color=color.white, texture=io)
```

```
for u in range(300):
    for i in range(3):
        add_box(position=(i+14, 0, u), color=color.white,
texture='derr.jpg')
```

```
for u in range(300):
    for i in range(3):
        zu = randint(0, len(noob)-1)
        tv = noob[zu]
        add_box(position=(i+7, 0, u), color=color.white, texture=tv)
```

```
for u in range(300):
    for i in range(3):
        add_box(position=(i, 0, u), color=color.white,
texture='ste.jpg')
```

```
def update():
    pass
```

```

tv = None

def input(key):
    global a, d, ddd, cv, t, eur, d_text, ur, b_text, noob, tvee, bb,
cc, aa, ui, tv
    for box in boxes:
        if box.hovered:
            if key == "right mouse down" and eur != 0:
                eur -= 1
                a = randint(0, len(list)-1)
                o = list[a]
                t = 1 if o == 'tr1.png' else 0
                add_box(box.position + mouse.normal, color=cv[t],
texture=o)
                d_text.text = str(f'box:{eur+ui}')
            elif key == "right mouse down" and ui != 0:
                ui -= 1
                tzeer = randint(0, 2)
                if tzeer == 2:
                    a = randint(0, len(list)-1)
                    tvee = list[a]
                else:
                    zu = randint(0, len(noob)-1)
                    tvee = noob[zu]
                t = 1 if tvee == 'tr1.png' else 0
                add_box(box.position + mouse.normal, color=cv[t],
texture=tvee)
                d_text.text = str(f'box:{eur+ui}')
            elif key == "left mouse down":
                boxes.remove(box)
                destroy(box)
            if key == "right mouse down" and ur != 0:
                ur -= 1
                a = randint(0, len(noob)-1)
                o = noob[a]
                t = 1 if o == 'tr1.png' else 0
                rtr = randint(1, 20)
                if not rtr == 1:
                    add_box(box.position + mouse.normal, color=cv[t],
texture=o)
                else:
                    a = ['mix.png', 'fant.png']
                    add_mix(box.position + mouse.normal,
texture=a[randint(0, 1)])
                    b_text.text = str(f'noob box:{ur}')

app.run()

```

Die **Lucky Blöcke** kann man noch nicht ab bauen und bei dein programm musst du die *pngs/jpegs/jpgs* nach deinen bildern nennen bei ursina kann man nur die *jpg, jpeg* oder *png* Bilder formate benutzen. Auf **Lucky Blöcken** Blöcke plazieren und **Lucky Blöcke** ab bauen und dafür **Blöcke** bekommen das alles musst du programmieren farsuch es zu erst allein und vergleiche es mit diesem programm!

Lösung:

```
from ursina import Sky
from ursina import Ursina
from ursina import Text
from ursina import color
from ursina import Button
from ursina import destroy
from ursina import mouse
from ursina import scene
from ursina.prefabs.first_person_controller import FirstPersonController
from random import randint

app = Ursina()
player = FirstPersonController()
eur = 0
liste = ['luck.png', 'wat.jpg']
io = liste[randint(0, 1)]
Sky(texture='sky_sunset')
list = ['retro.png', 'cry.png', 'obs.png', 'glo.jpg', 'bri.jpg',
'red.jpg', 'ingo.png']

noob = ['stein1.png', 'stein2.png', 'tr1.png', 'buer.jpg', 'der.jpeg',
'ft.png', 'rt.jpg', 'tatata.jpg', 'ste.jpg', 'derr.jpg']
d = 0
aa = 0
bb = 0
tvee = " "
ui = 0
cc = 0
lucky = []
z = 0
mixes = []
aaaa = 0
t = 0
n = 0
a = 0
ur = 0
cv = [color.white, color.white66]
wer = []
boxes = []
g = 1
d_text = Text('', origin=(0, -10), scale=1)
b_text = Text('', origin=(0, -11), scale=1)
player.position = 8, 100, 0
```

```
def add_box(position, color, texture):
    boxes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture=texture
        )
    )
```

```
def add_mix(position, texture):
    mixes.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color.white,
            position=position,
            texture=texture
        )
    )
```

```
def add_lucky(position, color, texture):
    lucky.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture=texture
        )
    )
```

```
def add_lu(position, color):
    wer.append(
        Button(
            parent=scene,
            model='cube',
            origin=0.5,
            color=color,
            position=position,
            texture='lu.png'
        )
    )
```

```

n = -21
ccc = -6
for _ in range(20):
    n += 1
    ccc = -6
    for _ in range(27):
        ccc += 1
        add_box(position=(ccc, 0, n), color=color.white,
texture='retro.png')

n = -1
for _ in range(150):
    n += 2
    add_lu(position=(15, 1, n), color=color.white)

n = -1
for _ in range(150):
    n += 2
    a = ['mix.png', 'fant.png']
    add_mix(position=(8, 1, n), texture=a[randint(0, 1)])

n = -1
for _ in range(150):
    n += 2
    io = liste[randint(0, 1)]
    add_lucky(position=(1, 1, n), color=color.white, texture=io)

for u in range(300):
    for i in range(3):
        add_box(position=(i+14, 0, u), color=color.white,
texture='derr.jpg')

for u in range(300):
    for i in range(3):
        zu = randint(0, len(noob)-1)
        tv = noob[zu]
        add_box(position=(i+7, 0, u), color=color.white, texture=tv)

for u in range(300):
    for i in range(3):
        add_box(position=(i, 0, u), color=color.white,
texture='ste.jpg')

def update():
    pass

tv = None

```

```

def input(key):
    global a, d, ddd, cv, t, eur, d_text, ur, b_text, noob, tvee, bb,
cc, aa, ui, tv
    destroyed_lucks = []
    destroyed_lu = []
    destroyed_mix = []
    for luck in lucky:
        if luck.hovered:
            if key == "left mouse down":
                destroyed_lucks.append(luck)
                eur += 1
                d_text.text = str(f'box:{eur+ui}')
            elif key == "right mouse down" and eur != 0:
                eur -= 1
                a = randint(0, len(list)-1)
                o = list[a]
                t = 1 if o == 'tr1.png' else 0
                add_box(luck.position + mouse.normal, color=cv[t],
texture=o)
                d_text.text = str(f'box:{eur+ui}')

    for luck in destroyed_lucks:
        aa += 1
        lucky.remove(luck)
        destroy(luck)

    for mix in mixes:
        if mix.hovered:
            if key == "left mouse down":
                destroyed_mix.append(mix)
                ui += 1
                d_text.text = str(f'box:{eur+ui}')
            elif key == "right mouse down" and ui != 0:
                ui -= 1
                tzeer = randint(0, 2)
                if tzeer == 2:
                    a = randint(0, len(list)-1)
                    tvee = list[a]
                else:
                    zu = randint(0, len(noob)-1)
                    tvee = noob[zu]
                t = 1 if tvee == 'tr1.png' else 0
                add_box(mix.position + mouse.normal, color=cv[t],
texture=tvee)
                d_text.text = str(f'box:{eur+ui}')

    for mix in destroyed_mix:
        bb += 1
        mixes.remove(mix)
        destroy(mix)

    for lu in wer:

```

```

        if lu.hovered:
            if key == "left mouse down":
                destroyed_lu.append(lu)
                ur += 1
                b_text.text = str(f'noob box:{ur}')
            elif key == "right mouse down" and ur != 0:
                ur -= 1
                a = randint(0, len(noob)-1)
                o = noob[a]
                t = 1 if o == 'tr1.png' else 0
                rtr = randint(1, 20)
                if not rtr == 1:
                    add_box(lu.position + mouse.normal, color=cv[t],
texture=o)

            else:
                a = ['mix.png', 'fant.png']
                add_mix(lu.position + mouse.normal,
texture=a[randint(0, 1)])
                b_text.text = str(f'noob box:{ur}')

    for lu in destroyed_lu:
        cc += 1
        wer.remove(lu)
        destroy(lu)

    for box in boxes:
        if box.hovered:
            if key == "right mouse down" and eur != 0:
                eur -= 1
                a = randint(0, len(list)-1)
                o = list[a]
                t = 1 if o == 'tr1.png' else 0
                add_box(box.position + mouse.normal, color=cv[t],
texture=o)

                d_text.text = str(f'box:{eur+ui}')
            elif key == "right mouse down" and ui != 0:
                ui -= 1
                tzeer = randint(0, 2)
                if tzeer == 2:
                    a = randint(0, len(list)-1)
                    tvee = list[a]
                else:
                    zu = randint(0, len(noob)-1)
                    tvee = noob[zu]
                t = 1 if tvee == 'tr1.png' else 0
                add_box(box.position + mouse.normal, color=cv[t],
texture=tvee)

                d_text.text = str(f'box:{eur+ui}')
            elif key == "left mouse down":
                boxes.remove(box)
                destroy(box)
            if key == "right mouse down" and ur != 0:
                ur -= 1

```

```

a = randint(0, len(noob)-1)
o = noob[a]
t = 1 if o == 'tr1.png' else 0
rtr = randint(1, 20)
if not rtr == 1:
    add_box(box.position + mouse.normal, color=cv[t],
texture=o)
else:
    a = ['mix.png', 'fant.png']
    add_mix(box.position + mouse.normal,
texture=a[randint(0, 1)])
    b_text.text = str(f'noob box:{ur}')

app.run()

```

Jetzt hast du dein erstes **Lucky block minecraft** vollständig programmiert.