



arnes 

Integracija Pythona v Blenderju za obogatitve kontekstualne vsebine medijskega izdelka

Davorin Babič, Srednja medijska in grafična šola Ljubljana
Mreža znanja 2021, od 23. do 30. novembra

Izhodiščna ideja

“Code is not purely abstract and mathematical; it has significant social, political, and **aesthetic** dimensions.”

– Montfort Nick et al. `10 PRINT CHR$(205.5+RND(1)); : GOTO 10`



Vsebinski koncept integracije

- Ponavljanje in izpopolnjevanje temeljnih **programerskih konceptov** in **programiranje** v izbranem besedilnem programskem jeziku z ustvarjalnimi vajami in majhnimi projekti.
- Razvijanje **ustvarjalnih pristopov** in **sodobnih oblikovalskih praks**, ki jih medijski tehniki lahko uporabljajo v svojem učnem procesu — **statično**, **dinamično** in **interaktivno** generativno snovanje preprostih prototipov in izdelkov v izbrani platformi.

Ustvarjalne metode

V procesu učenja ustvarjanja z integracijo obeh omenjenih orodij sloni delo dijakov na **treh ustvarjalnih metodah**:

- **generativni pristopi** v ustvarjanju vsebin v modeliranju kot tudi v fizičnih izdelkih,
- **kreativno kodiranje** in **avtomatizacija** v procesu konceptualne zamisli in
- **hibridni pristopi** k ustvarjanju in oblikovanju na osnovi besedilne programske kode v izbranem programskem jeziku.

Izpeljava integracije

Cilj učne situacije:

- združiti teorijo in metodologijo iz računalništva z načeli **3D modeliranja**, **geometrijske estetike** in **teorije vizualnih medijev**,
- **kreativno prakso**,
- pedagoškimi pristopi iz **likovne** in **grafične umetnosti** ter **stereometrije**.

Izpeljava integracije – etape

- e1: razvijati svoj **umetniški** in **oblikovalski besednjak** skupaj z uvajanjem v razvojnem okolju in **programsko sintakso**.
- e2: nadgradnji ustvarjalnega besednjaka skozi vaje osnovne **programerske spretnosti**.
- e3: raziskovanje razlik med **digitalno generativno postopkovno vizualizacijo** z uporabo **pseudonaključnih števil** in ponavljajočih se naključnih dogajanj ter klasično zasnovanih ustvarjalnih pristopov v tovrstni vizualizaciji.

Izpeljava integracije

- e4: temeljni umetniški in oblikovalski koncepti, nanašajoči se na dojemanje informacij, pridobljenih iz svetlostnih razlik na slikovni ploskvi digitalnega modela in njegove teksture — delo s **šumom**.
- e5: uvedba načel ustvarjanja 3D animacij z namenom ustvariti prepričljivejše animacije. Uvedeni so nekateri naprednejši programerski koncepti, kot so **objekti** in **razredi**.
- e6: vizualizacija **algoritmičnega procesa**, tako da lahko dijaki vidijo, kaj lahko ustvari ustrezno napisana programska koda.

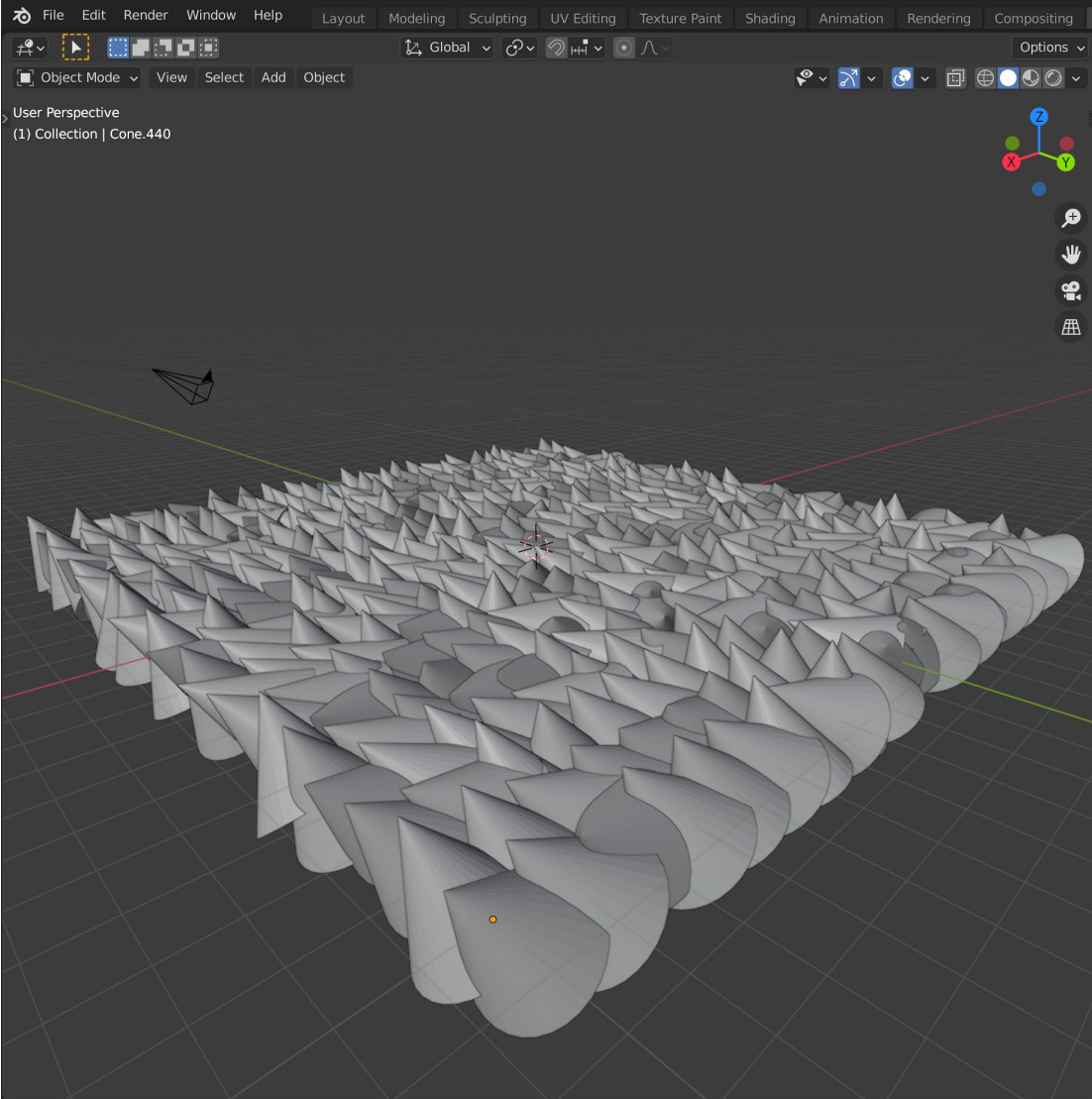
Pedagoško-didaktična ustreznost

Opiranje na pristope:

- strukturirani zaporedni vsebinski moduli,
- različne stopnje zahtevnosti reševanja problemov,
- navodila za izvedbo, ki jih je bilo enostavno razumeti,
- potrebno minimalno branje predstavljene teorije, uporaba zanimivih in ustreznih primerov,
- možnost takojšnje povratne informacije,
- učenje v ritmu z lastnim tempom in
- sprotno spremljanje dela dijakov, individualno delo.

Programiranje digitalnih medijev

- Interdisciplinarno zasnovano področje, ki združuje **osnove računalništva** z **estetskimi načeli** in kreativnimi praksami iz umetnosti in oblikovanja v 3D prostoru.
- Dijaki se s pomočjo programskega jezika **Python** urijo v **računalniškem mišljenju**, povrh pa so ustvarjajo **digitalni portfelj** estetsko zasnovanih generativnih vizualizacij v platformi **Blender**, ki so vključevale **osnovne koncepte** in **konstrukte programiranja**, povezane z **likovno kompozicijo** in **načeli likovnega reda**.



```
1 import bpy
2 from random import *
3
4 def ustvari_stozce():
5     for x in range(-10, 11):
6         for y in range(-10, 11):
7             r1 = random()
8             r2 = random()
9             r3 = random()
10            bpy.ops.mesh.primitive_cone_add(location=(x, y, 0), rotation=(r1, r2, r3))
11
12 def ustvari_unijo():
13     base_cone = bpy.data.objects[0]
14     bpy.context.scene.objects.active = base_cone
15     for i, cone in enumerate(bpy.data.objects):
16         if i == 0:
17             continue
18         boolean = base_cone.modifiers.new('ConeBoolean', 'BOOLEAN')
19         boolean.object = cone
20         boolean.operation = 'UNION'
21         bpy.ops.object.modifier_apply(apply_as='DATA', modifier='ConeBoolean')
22         bpy.context.scene.objects.unlink(cone)
23
24 if __name__ == "__main__":
25     ustvari_stozce()
26     ustvari_unijo()
27
28
```

```
>>> |
589), scale=(1, 1, 1))
✓ bpy.ops.mesh.primitive_cone_add(enter_editmode=False, location=(10, 9, 0), rotation=(0.363993, 0.301123, 0.245174), scale=(1, 1, 1))
✓ bpy.ops.mesh.primitive_cone_add(enter_editmode=False, location=(10, 10, 0), rotation=(0.27493, 0.629749, 0.409968), scale=(1, 1, 1))
✓ bpy.ops.text.run_script()
✗ Python script failed, check the message in the system console
```

Scene Collection

- Collection
 - Camera
 - Cone
 - Cone.001
 - Cone.002
 - Cone.003
 - Cone.004
 - Cone.005
 - Cone.006

Current File

- Brushes 74
- Cameras 3
- Collections 99
- Images 2
- Lights 1
- Line Styles 1
- Materials 2
- Meshes 99
- Objects 99
- Palettes

Scene

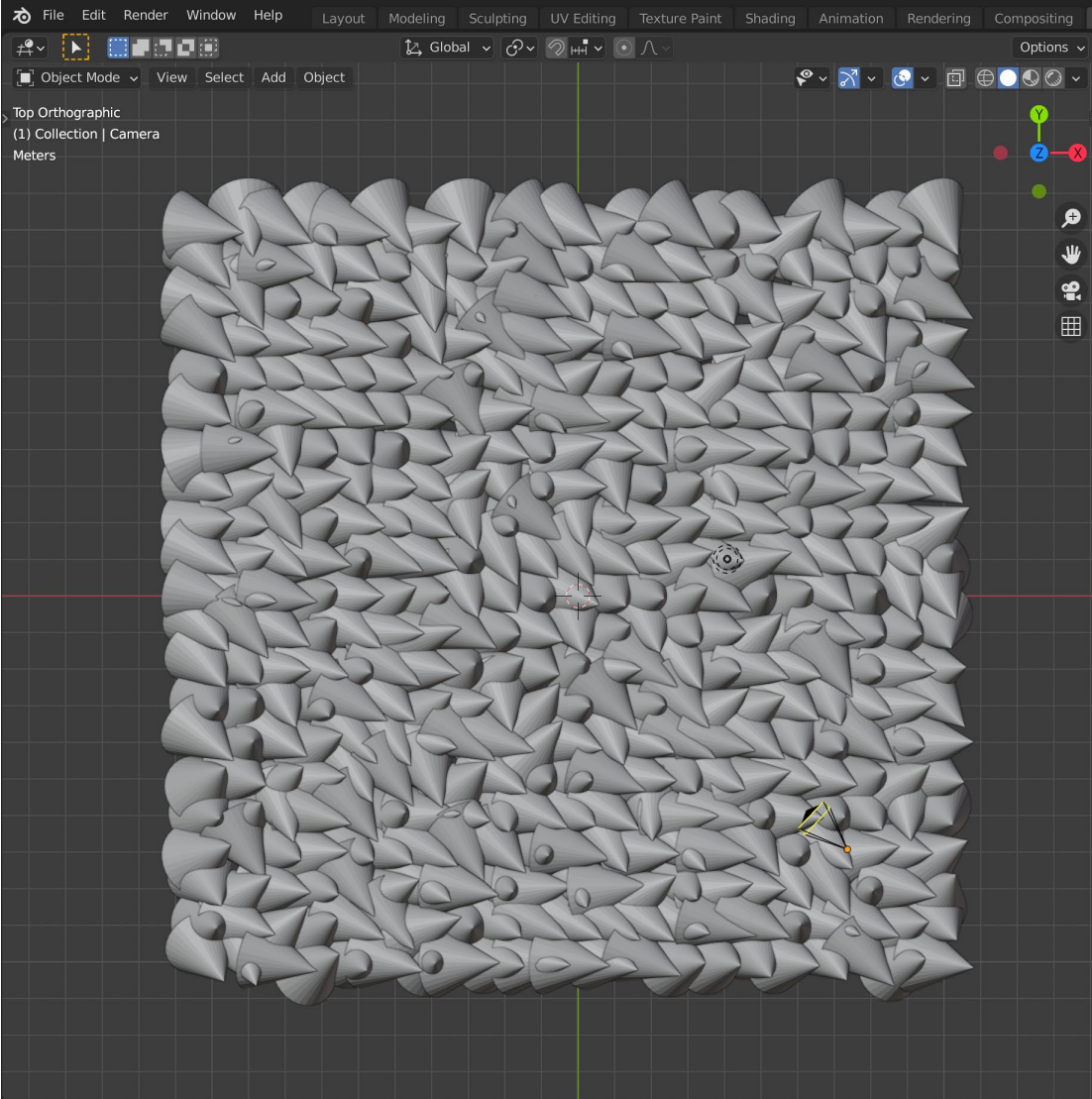
Render Engine: Eevee

Sampling

Render	64
Viewport	16

Viewport Denoising

- Ambient Occlusion
- Bloom
- Depth of Field
- Subsurface Scattering
- Screen Space Reflections
- Motion Blur
- Volumetrics
- Performance
- Hair
- Shadows
- Indirect Lighting
- Film
- Simplify
- Grease Pencil
- Freestyle
- Color Management



```
1 import bpy
2 from random import *
3
4 def ustvari_stozce():
5     for x in range(-10, 11):
6         for y in range(-10, 11):
7             r1 = random()
8             r2 = random()
9             r3 = random()
10            bpy.ops.mesh.primitive_cone_add(location=(x, y, 0), rotation=(r1, r2, r3))
11
12 def ustvari_unijo():
13     base_cone = bpy.data.objects[0]
14     bpy.context.scene.objects.active = base_cone
15     for i, cone in enumerate(bpy.data.objects):
16         if i == 0:
17             continue
18         boolean = base_cone.modifiers.new('ConeBoolean', 'BOOLEAN')
19         boolean.object = cone
20         boolean.operation = 'UNION'
21         bpy.ops.object.modifier_apply(apply_as='DATA', modifier='ConeBoolean')
22         bpy.context.scene.objects.unlink(cone)
23
24 if __name__ == "__main__":
25     ustvari_stozce()
26     ustvari_unijo()
27
28
```

```
>>>
bpy.context.space_data.shading.type = 'RENDERED'
bpy.context.space_data.shading.type = 'SOLID'
bpy.context.space_data.shading.type = 'WIREFRAME'
bpy.context.space_data.shading.type = 'SOLID'
bpy.context.space_data.params.filename = "cones1.blend"
Saved "cones1.blend"
```

Scene Collection

- Collection
 - Camera
 - Cone
 - Cone.001
 - Cone.002
 - Cone.003
 - Cone.004
 - Cone.005
 - Cone.006

Current File

- Brushes 74
- Cameras 3
- Collections 99
- Images 2
- Lights 1
- Line Styles 1
- Materials 2
- Meshes 99
- Objects 99
- Palettes

Scene

Render Engine: Eevee

Sampling

Render	64
Viewport	16

Viewport Denoising

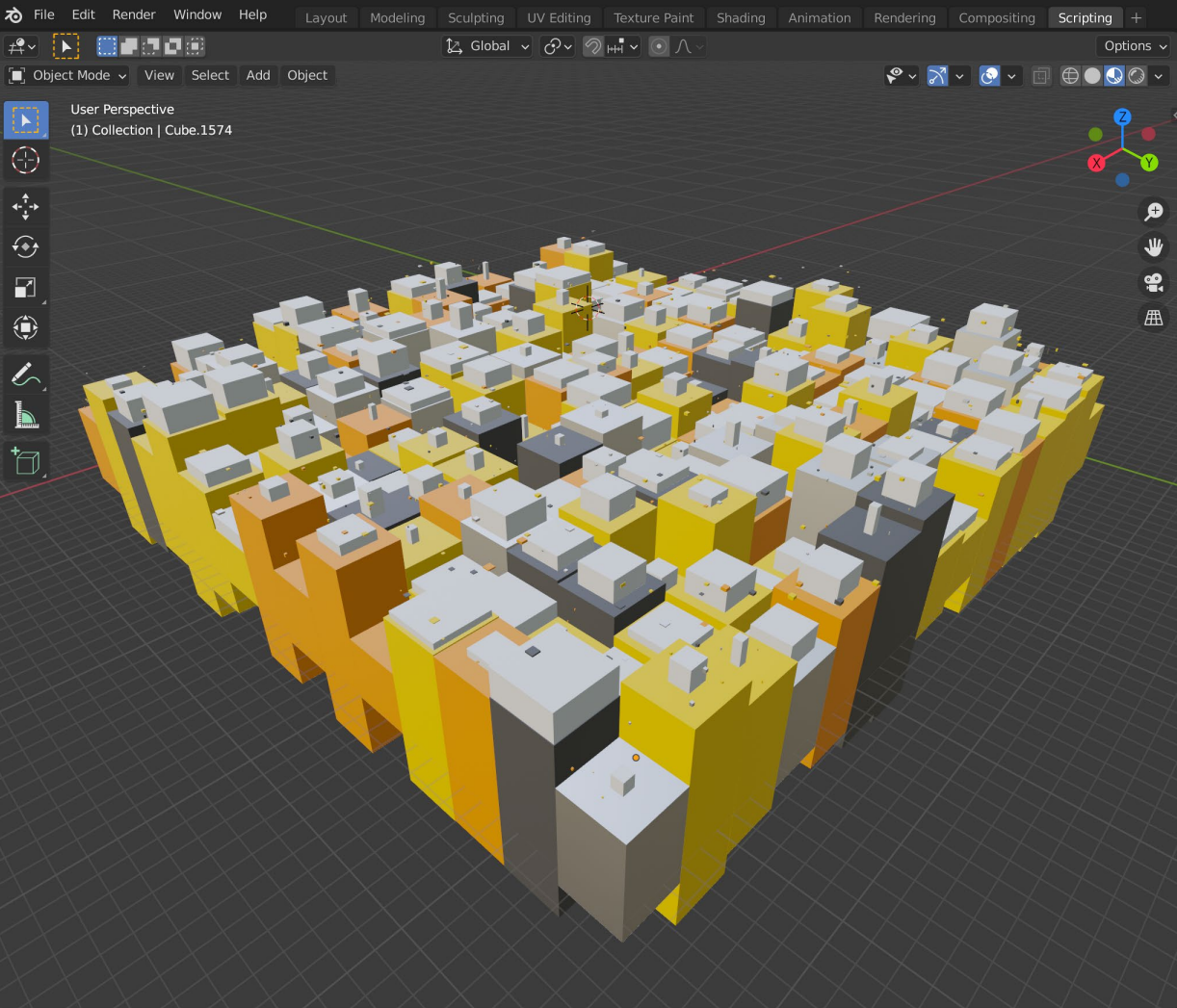
- Ambient Occlusion
- Bloom
- Depth of Field
- Subsurface Scattering
- Screen Space Reflections
- Motion Blur
- Volumetrics
- Performance
 - Hair
 - Shadows
 - Indirect Lighting
 - Film
 - Simplify
 - Grease Pencil
 - Freestyle
 - Color Management

```
import bpy
from random import *

def ustvari_stozce():
    for x in range(-10, 11):
        for y in range(-10, 11):
            r1 = random()
            r2 = random()
            r3 = random()
            bpy.ops.mesh.primitive_cone_add(location=(x, y, 0), rotation=(r1, r2, r3))

def ustvari_unijo():
    base_cone = bpy.data.objects[0]
    bpy.context.scene.objects.active = base_cone
    for i, cone in enumerate(bpy.data.objects):
        if i == 0:
            continue
        boolean = base_cone.modifiers.new('ConeBoolean', 'BOOLEAN')
        boolean.object = cone
        boolean.operation = 'UNION'
        bpy.ops.object.modifier_apply(apply_as='DATA', modifier='ConeBoolean')
        bpy.context.scene.objects.unlink(cone)

if __name__ == "__main__":
    ustvari_stozce()
    ustvari_unijo()
```



```
bpy.context.space_data.show_region_toolbar = False
bpy.context.space_data.show_region_toolbar = True
bpy.context.space_data.show_region_ui = False
```

```
1 import bpy
2 import random
3
4 stevilo_kock = 15
5 velikost_kocke = 3
6
7 x = velikost_kocke/2
8 y = velikost_kocke/2
9 z = velikost_kocke/2
10
11 materials_list =
12 ['Material.001', 'Material.002', 'Material.002', 'Material.003', 'Material.
13 004']
14
15 start_z = 1
16 end_z = 4
17
18 x_floating = 0
19 y_floating = 0
20
21 print("test")
22 #bpy.ops.mesh.primitive_cube_add(size=velikost_kocke ,
23   enter_editmode=False, align='WORLD', location=(8,8,2.5),
24   scale=(1, 1, 1.5))
25
26 #bpy.ops.mesh.primitive_cube_add(size=velikost_kocke ,
27   enter_editmode=False, align='WORLD', location=(8,8,2.5),
28   scale=(1, 1, 2))
29
30 #bpy.ops.mesh.primitive_cube_add(size=velikost_kocke ,
31   enter_editmode=False, align='WORLD', location=(8,8,2.5),
32   scale=(1, 1, 3))
33
34 for n in range(0, stevilo_kock):
35     x_floating = 0
36     x = velikost_kocke/2
37     for i in range(0, stevilo_kock):
38         bpy.ops.mesh.primitive_cube_add(size=velikost_kocke ,
39           enter_editmode=False, align='WORLD', location=(x,y,z),
40           scale=(1, 1, 1))
41
42         random_variable_z_scale = random.uniform(start_z, end_z)
43         random_variable_xy_scale = random.uniform(0.1, 0.99)
44         random_variable_z_scale_medium = random.uniform(0.1, 0.5)
45
46         bpy.ops.transform.resize(value=(1, 1,
47           random_variable_z_scale), orient_type='GLOBAL', orient_matrix=((1, 0,
48           0), (0, 1, 0), (0, 0, 1)), orient_matrix_type='GLOBAL',
49           constraint_axis=(True, True, True), mirror=False,
50           use_proportional_edit=False, proportional_edit_falloff='SMOOTH',
51           proportional_size=1, use_proportional_connected=False,
52           use_proportional_projected=False, release_confirm=True)
53
54         item = bpy.context.object
55
56         item.data.materials.append(bpy.data.materials[random.choice(materials_list)])
57
58         bpy.ops.mesh.primitive_cube_add(size=velikost_kocke ,
59           enter_editmode=False, align='WORLD',
60           location=(x,y,((random_variable_z_scale*velikost_kocke)/2+2*z),
61           scale=(1, 1, 1))
62
63         bpy.ops.transform.resize(value=(random_variable_xy_scale,
64           random_variable_xy_scale, random_variable_z_scale_medium),
65           orient_type='GLOBAL', orient_matrix=((1, 0, 0), (0, 1, 0), (0, 0, 1)),
66           orient_matrix_type='GLOBAL', constraint_axis=(True, True, True),
67           mirror=False, use_proportional_edit=False,
68           proportional_edit_falloff='SMOOTH', proportional_size=1,
69           use_proportional_connected=False, use_proportional_projected=False,
70           release_confirm=True)
```

Blender 2.92.0 interface showing the right-hand side panels:

- Find & Replace:** Search for 'size_of_cube', 'Find Next', 'Replace', 'Replace All'. Filter: 'velikost_kocke'. Options: Case, Wrap, All.
- Properties:** Margin: 80, Font Size: 12, Tab Width: 4, Indentati...: Spaces.
- Scene Collection:** Collection (checked), Cube (checked), Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.00, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01, Cube.01.
- Current File:** Dimensions: Resolut...: 1920 px, Y: 1080 px, %: 100%, Aspect X: 1.000, Y: 1.000, Render...: checked, Crop to...: unchecked, Frame...: 1, End: 250, Step: 1, Frame...: 24 fps.
- Time Remapping:** Stereoscopic: unchecked, Output: /tmp/.
- File Properties:** Saving: File Exte...: checked, Cache R...: unchecked, File For...: PNG, Color: BW RG RG, Color D...: 8 16, Compr...: 15%, Image...: Overwrite: checked, Placeholder: unchecked, Metadata: unchecked, Post Processing: unchecked.

Text: Internal

```
import bpy
import random

stevilo_kock = 15
velikost_kocke = 3

x = velikost_kocke/2
y = velikost_kocke/2
z = velikost_kocke/2

materials_list = ['Material.001','Material.002','Material.002','Material.003','Material.004']

start_z = 1
end_z = 4

x_floating = 0
y_floating = 0

for n in range(0,stevilo_kock):
    x_floating = 0
    x = velikost_kocke/2
    for i in range(0,stevilo_kock):
        bpy.ops.mesh.primitive_cube_add(size=velikost_kocke , enter_editmode=False, align='WORLD', location=(x,y,z), scale=(1, 1, 1))

        random_variable_z_scale = random.uniform(start_z, end_z)
        random_variable_x
        y_scale = random.uniform(0.1, 0.99)
        random_variable_z_scale_medium = random.uniform(0.1, 0.5)

        bpy.ops.transform.resize(value=(1, 1, random_variable_z_scale), orient_type='GLOBAL', orient_matrix=((1, 0, 0), (0, 1, 0), (0, 0, 1)), orient_matrix_type='GLOBAL', constraint_axis=(True, True, True), mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1, use_proportional_connected=False, use_proportional_projected=False, release_confirm=True)

        item = bpy.context.object
```

<p>Osnovni programerski koncepti</p>	<p>Tehnike kreativnega kodiranja</p>
<p>Osnove programiranja v izbranem programskem jeziku / okolju</p>	<p>Vgrajeni osnovni geometrijski objekti (geometrijski primitivi), koordinatni sistem, načela kreativnega kodiranja</p>
<p>Spremenljivke, podatkovni tipi, izrazi, stavki</p>	<p>Modeliranje v ustreznem merilu, sorazmernost razdalj med geometrijskimi primitivi</p>
<p>Krmilne strukture in vejitve</p>	<p>Uporaba preprostih iteracij za izrisovanje velikega števila različnih objektov in geometrijskih oblik, uporaba pogojnih stavkov v preprostih simulacijah, rekurzija</p>
<p>Modularnost: pisanje funkcij, z vsebovanimi krmilnimi strukturami in generatorji naključnih števil</p>	<p>Izrisovanje parametrično podanih objektov:lega, velikost, barva obrobe in polnila itd., naključno ustvarjene kompozicije v digitalnem 3D prostoru</p>
<p>Seznami in nizi</p>	<p>Kompleksnejše strukturne likovne kompozicije z mnogo različnimi objekti, shranjenimi v seznamih</p>
<p>Uporaba matematičnih konceptov: kotne funkcije, geometrijske transformacije v ravnini in prostoru</p>	<p>Kompleksnejše abstraktno / geometrijsko oblikovanje, strukturne likovne kompozicije pravilnih teles, ustvarjalna estetika</p>

Rezultati

- Za usvajanje **računalniškega mišljenja** dijakom pripravimo **podlago** za didaktične aktivnosti, prek katerih usvajajo osnovne in naprednejše programerske koncepte.
- Vsebinske in tehnične prilagodljivosti znotraj posameznih učnih enot dijakom omogočajo, da svoje ustvarjalne spretnosti usmerjajo v sestavljanje tako **preprostih** kot **zapletenih računalniških programov** v ustvarjanju digitalne generativne postopkovne vizualizacije.

Rezultati

- Predstavitev **besedilnega programiranja** in računalniškega mišljenja skozi učenje predstavljenega programskega jezika.
- Izpostavljanje scenarijev, kje in kdaj je mogoče **kodiranje** uporabiti za **skiciranje konceptualnih idej** in **razvoja delujočih prototipov** programske kode, ki se osredotočajo na **izrazne medije**.
- Spodbujanje k **razumevanju etike** odprte kode z uporabo kode, ki je na voljo na spletu, ter s sprostivijo in dokumentiranjem svojega projekta z istim pristopom.

davorin.babic@smgs.si

arnes 