

Projekta darbs – Interaktīvas vides izveide ar objektu ievietošanu.

Uzdevuma mērķis:

Strādājot pārī izveidot interaktīvu 3D vidi par izvēlēto tēmu ar pašu veidotiem objektiem un organizēt versiju kontroli, izmantojot GitHub. Šis darbs nedrīkst būt saistīts vai pārveidots no iepriekšējiem darbiem, nepieciešams izveidot radošu vidi veidojot jaunus 3D objektus, šoreiz objektu apjoms nav norādīts, bet tas nenozīmē, ka tie nav jāveido, varat importēt arī modeļus (max 2, kas netiks vērtēti) no interneta norādot saiti, kur tos dabūjāt, visas nepieciešamās formas, objektus veidojat paši, vairāk parasti vizard objekti nederēs - bumbiņas, konusi, cilindri.

Uzdevuma soļi:

1. Pašu veidoto objektu izveide

1. 3D objektu modelēšana:

- Izveidojiet 3D objektus, izmantojot Blender, Tinkercad vai citu 3D modelēšanas programmatūru.
- Eksportējiet objektus piemērotā formātā, piemēram, .osgb vai .obj.

2. Objektu importēšana Vizard 7 vidē:

- Pārlicinieties, ka objektu faili ir pieejami Vizard 7 projektā.
- Importējiet modeļus projektā, izmantojot viz.add() funkciju.

```
import viz  
viz.go()
```

```
# Importējiet un novietojiet pašu veidotu objektu  
customObject = viz.add('path/to/your/object.osgb')  
customObject.setPosition([0, 0, 5])
```

2. Vides izveide un interaktivitātes pievienošana

1. Novietojiet 3D objektus vidē:

- Novietojiet objektus, izmantojot setPosition() un citus pozicionēšanas rīkus.

```
customObject.setPosition([0, 0, 5])  
anotherObject.setPosition([2, 0, 5])
```

2. Pievienojiet interaktivitāti:

- Izveidojiet funkcijas, lai lietotāji varētu mijiedarboties ar objektiem, piemēram, tos pārvietot vai mainīt krāsu.

```
import vizact

def onPick(e):
    if e.object.valid():
        e.object.color(viz.RED)
        move = vizact.moveTo([0, 2, 0], time=1)
        e.object.addAction(move)

viz.callback(viz.PICK_EVENT, onPick)
viz.mouse(viz.OFF)

vizact.onkeydown('p', viz.mouse.pick)
```

3. Kameru kontrole:

- Pievienojiet iespēju kontrolēt kameru, lai lietotāji varētu apskatīt vidi no dažādiem leņķiem.

```
def updateCamera():
    viz.MainView.move([0, 0, 0.1])

vizact.ontimer(0, updateCamera)
```

3. Versiju kontroles organizēšana ar GitHub

1. Inicializējiet Git repo:

- Atveriet komandrindu un inicializējiet Git repozitoriju projekta mapē.

```
git init
```

2. Pievienojiet failus Git:

- Pievienojiet projekta failus un veiciet pirmo *commit*.

```
git add .
git commit -m "Initial commit"
```

3. Izveidojiet GitHub repozitoriju:

- Izveidojiet jaunu repozitoriju GitHub un pievienojiet tā attālo URL savam lokālajam *repo*.

```
git remote add origin https://github.com/username/repository.git
git push -u origin master
```

4. Versiju kontrole un sadarbība:

- Abi pāra dalībnieki var strādāt ar projektu, veicot izmaiņas savos *branches* un izmantojot *pull requests*, lai integrētu izmaiņas.

```
git checkout -b feature-branch
```

```
# veiciet izmaiņas  
git add .  
git commit -m "Added new feature"  
git push origin feature-branch
```

5. Pull requests un apvienošana:

- Veiciet *pull requests* GitHub, lai apvienotu izmaiņas galvenajā *branches*.
- Pārskatiet un apstipriniet izmaiņas pirms apvienošanas.

4. Testēšana un uzlabošana

- Testējiet izveidoto vidi un pārlicinieties, ka visi objekti ir pareizi novietoti un interaktivitāte darbojas.
- Pielāgojiet un uzlabojiet vidi, pievienojot papildu funkcionalitāti vai vizuālos elementus.

Uzdevuma nodošana:

Vērtēšanas Kritēriji

1. 3D Objektu Kvalitāte (20 punkti)

- **Objektu oriģinalitāte un dizains** (10 punkti): Pašu veidoto objektu unikalitāte un radošums.
- **Tehniskā kvalitāte un detalizācija** (10 punkti): Objektu precizitāte, modelēšanas prasmes un izmantotās tekstūras kvalitāte.

2. Integrācija Vizard 7 Vidē (20 punkti)

- **Objektu pareiza importēšana** (10 punkti): Objektu korekta importēšana un attēlošana Vizard 7 vidē.
- **Objektu novietojums un pozicionēšana** (10 punkti): Objektu novietojums 3D telpā, ievērojot proporcijas un mērogus.

3. Interaktivitāte (20 punkti)

- **Lietotāju mijiedarbība ar objektiem** (10 punkti): Spēja izvēlēties, pārvietot, mainīt krāsu vai veikt citas darbības ar objektiem.
- **Kameru kontrole** (10 punkti): Efektīva kameras kontrole un iespēja apskatīt vidi no dažādiem leņķiem.

4. GitHub Versiju Kontrole (20 punkti)

- **Projekta repozitorija struktūra un dokumentācija** (10 punkti): Sakārtots repozitorijs, skaidra mapju un failu struktūra, un detalizēta dokumentācija (README, instalācijas un lietošanas instrukcijas).
- **Commit vēsture un pull requesti** (10 punkti): Skaits un kvalitāte veiktajiem commits, efektīva pull requestu un apvienošanas prakse.

5. Projekta Prezentācija (10 punkti)

- **Projekta demonstrācija** (10 punkti): Projekta funkcionalitātes un galveno iezīmju demonstrācija, uzsverot paveikto darbu.

6. Sadarbība un Komunikācija (10 punkti)

- **Komandas darbs un sadarbība** (5 punkti): Abu pāra dalībnieku vienlīdzīga iesaiste un sadarbība projektā.
- **Komunikācijas kvalitāte** (5 punkti): Efektīva komunikācija starp dalībniekiem un problēmu risināšana.

Kopējais punktu skaits: 100 punkti

Papildus punkti (līdz 10 punkti)

- **Papildus funkcionalitāte vai uzlabojumi**: Projekta demonstrēšana nodrošināta 3D alā.