Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Systémová príručka

Príloha

2022 **Bc. Michal Bardzák**

Obsah

Zoznam obrázkov

1 Funkcia programu

Programový systém opísaný v tejto príručke je implementáciou postupov na transformáciu objektov reálneho sveta do virtuálnych prostredí.
Implementovaný softvér zahŕňa používateľské rozhrania, ktoré umožňujú používateľom vytvárať virtuálne prostredia založené na prostrediach z reálneho sveta a vykonávať na týchto prostrediach meranie dopadu výkonu týchto prostredí na výkon zariadení.

2 Analýza riešenia

Teoretický základ tohto riešenia bol popísaný v hlavnej časti diplomovej práce.

2.1 Časti navrhnutého systému

Navrhnutý systém transformácie objektov je rozdelený do dvoch hlavných častí.

Prvou časťou je doplnok k programu Blender, ktorý bol napísaný v programovacom jazyku Python a jeho úlohou je spracovať získané mapové údaje tak, aby výsledkom bola scéna ktorú je možné použiť vo webovej virtuálnej realite.

Druhú časť riešenia tvorí aplikácia vo webovom rámci pre rozšírenú realitu A-Frame. Táto aplikácia bola napísaná v značkovacom jazyku HTML a skriptovacom programovacom jazyku JavaScript. Účelom tejto aplikácie je zhodnotiť dopad vytvorených prostredí na výkon zariadení, na ktorých je táto aplikácia spustená.

2.2 Python

Pre implementáciu doplnku v programe Blender bol použitý jazyk Python. Python bol použitý z toho dôvodu že program Blender ponúka možnosť vytvárať v tomto jazyku skripty, ktoré môžu byť použité na automatizáciu rôznych činností v tomto programe.

2.3 Javascript

Jazyk JavaScript bol použitý na vytváranie komponentov pre scénu webovej virtuálnej reality. Komponenty v rozhraní A-Frame môžu byť pripojené k entitám za

účelom modifikovania ich vzhľadu, správania alebo funkcionality.

2.4 HTML

Značkovací jazyk HTML je v rozhraní A-Frame používaný rovnakým spôsobom ako pri obyčajných webových stránkach. Slúži hlavne na vytváranie statických elementov v scéne.

3 Popis funkcie kódu v doplnku k programu Blender

3.1 Vlastnosti

Táto trieda obsahuje kód kde sú uvedené vlastnosti ktoré chceme meniť. Tieto vlastnosti sú napríklad ceste exportovania a cesta k textúram, veľkosť projekcie kocky, výška o ktorú chceme budovu predĺžiť, počet segmentov do ktorých chceme združiť budovy v scéne a možnosť ktorá je vybraná pri výbere typu materiálu budov. Tieto vlastnosti sú uložené ako premenné.

```
1  class MyProperties(bpy.types.PropertyGroup):
2  exportPath : bpy.props.StringProperty(name= "Export Path", subtype='
     FILE_PATH')
3
4  pathToTextures : bpy.props.StringProperty(name= "Image Path",
     subtype='FILE_PATH')
5
6  cubeProjectionSize : bpy.props.FloatProperty(name= "Projection Size
     ", default=0.08, soft_min= 0, soft_max= 1)
7
8  buildingSize : bpy.props.IntProperty(name= "Building Height",
     default=8, soft_min= 0)
9  partsToSeparate : bpy.props.IntProperty(name= "Segments", default=8,
       soft_min= 0)
10
11 operationChoice : bpy.props.EnumProperty(
12 name= "Texture type",
13 description= "sample Text",
14 items= [('OP1', "Use image", ""),
15 ('OP2', "Use color", "")
16 ]
17
18 )
```

3.2 Materiály

Tento kód slúži na vytváranie a priraďovanie materiálov k jednotlivým budovám podľa toho ktorá možnosť bola zvolená v užívateľskom rozhraní. Pri prvej možnosti, čiže používaní textúry, je najprv získaná získaná cesta, ktorá vedie k textúram ktoré chceme použiť. Následne je pre každú textúru v danom adresári vykonaný kód, ktorý skontroluje či už materiál s daním menom existuje. Ak tomu tak nieje, bude vytvorený materiál s rovnakým menom ako textúra. Bude mu povolené používať uzly a budú mu pridané uzly BSDF a ShaderNodeTexImage, ktoré sú potrebné na priradenie textúry k materiálu. Následne je načítaná textúra z adresára a je priradená materiálu pomocou uzlu BSDF ako farba. Po tom čo bol tento kód vykonaný pre každú textúru je vybraný ako aktívny materiál ten ktorý sa nachádza na indexe nula. Na tomto indexe by sa mali nachádzať steny budov. Následne je zýskaný odkaz na kolekciu ktorá obsahuje modely budov. Pre každú budovu v kolekcii je vykonaný kód ktorý vyberie aktívny materiál a ak existuje tak je mu náhodne priradený jeden z materiálov ktoré obsahujú textúry. V prípade že bola zvolená druhá možnosť je vytvorené počítadlo, ktoré bude počítať počet vytvorených materiálov obsahujúcich farbu. Je zvolený aktívny materiál na indexe nula a získaný odkaz na kolekciu s budovami. Následne je pre každú budovu vytvorený nový materiál s náhodnou farbou dokiaľ nieje dosiahnutá maximálna hodnota počtu materiálov, čo je v tomto prípade desať. Po dosiahnutí maxima sú budovám náhodne priradené materiály ktoré boli už predtým vytvorené.

```
1  class SinkOperator(bpy.types.Operator):
2  """Tooltip"""
3  bl_idname = "sink.1"
4  bl_label = "Simple Object Operator"
5  def execute(self, context):
6  scene = context.scene
7  myProps = scene.my_props
8
9  if myProps.operationChoice == 'OP1':
10 os.listdir(myProps.pathToTextures)
11 for name in os.listdir(myProps.pathToTextures):
12 if name in bpy.data.materials:
13 continue
14 new_mat = bpy.data.materials.new(""+name)
15 new_mat.use_nodes = True
16 bsdf = new_mat.node_tree.nodes["Principled BSDF"]
17 texImage = new_mat.node_tree.nodes.new('ShaderNodeTexImage')
```

```
18  texImage.image = bpy.data.images.load(myProps.pathToTextures+name)
19  new_mat.node_tree.links.new(bsdf.inputs['Base Color'], texImage.
     outputs['Color'])
20  x = int(0)
21  bpy.context.object.active_material_index = 0
22  sce = bpy.data.collections['buildings'].all_objects
23  for ob in sce:
24  x+=1
25  mat = ob.active_material
26  if mat:
27  ob.active_material = bpy.data.materials.get(""+str(random.randrange(
     len(os.listdir(myProps.pathToTextures)))+1)+".jpg")
28  if myProps.operationChoice == 'OP2':
29  x = int(0)
30  matsNum = int(0)
31  bpy.context.object.active_material_index = 0
32  sce = bpy.data.collections['buildings'].all_objects
33  for ob in sce:
34  x+=1
35  mat = ob.active_material
36  if matsNum < 10:
37  if mat:
38  ob.active_material = mat.copy()
39  ob.active_material.name = "col"+str(matsNum)
40  matsNum+=1
41  ob.active_material.diffuse_color = (random.uniform(0, 1), random.
     uniform(0, 1), random.uniform(0, 1), 1)
42  if matsNum >= 10:
43  ob.active_material = bpy.data.materials.get("col"+str(random.
     randrange(10)+1)+"")
44  return {'FINISHED'}
```

3.3 Predlženie budov

V tejto časti kódu je vykonávané predlžovanie budov tak aby sa nevznášali nad povrchom terénu v scéne. Najprv sú vybrané všetky budovy nachádzajúce sa v kolekcii. Následne je zvolený editačný režim, a ako aktívny materiál je vybraný ten na indexe dva. Potom je zvolená časť na ktorej sa materiál nachádza. Táto časť by mala byť strecha. Strechy budov sú následne posunuté o zvolenú výšku čo efektívne predĺži budovy. Po predĺžení je vykonané premietnutie kocky s hodnotou ktorá bola zadaná. Následne sú zvolené všetky budovy a sú posunuté smerom nadol tak aby bola strecha v pôvodnej výške. Po vykonaní týchto úkonov je edi-

tačný režim opustený, je zrušené vybratie všetkých budov a aktívny materiál je nastavený na index nula.

```
1  class BuildingOperator(bpy.types.Operator):
2  """Tooltip"""
3  bl_idname = "building.1"
4  bl_label = "Simple Object Operator"
5  def execute(self, context):
6  scene = context.scene
7  myProps = scene.my_props
8  for obj in bpy.data.collections['buildings'].all_objects:
9  obj.select_set(True)
10 bpy.ops.object.editmode_toggle()
11 bpy.context.object.active_material_index = 1
12 bpy.ops.object.material_slot_select()
13 bpy.ops.transform.translate(value=(0, 0, (myProps.buildingSize)),
    orient_type='GLOBAL', orient_matrix=((1, 0, 0), (0, 1, 0), (0, 0,
    1)), orient_matrix_type='GLOBAL', constraint_axis=(False, False,
    True), mirror=True, use_proportional_edit=False,
    proportional_edit_falloff='SMOOTH', proportional_size=1,
    use_proportional_connected=False, use_proportional_projected=False)
14 bpy.ops.mesh.select_all(action='SELECT')
15 bpy.ops.uv.cube_project(cube_size= myProps.cubeProjectionSize,
    correct_aspect=True, clip_to_bounds=False, scale_to_bounds=False)
16 bpy.ops.transform.translate(value=(0, 0, -(myProps.buildingSize)),
    orient_type='GLOBAL', orient_matrix=((1, 0, 0), (0, 1, 0), (0, 0,
    1)), orient_matrix_type='GLOBAL', constraint_axis=(False, False,
    True), mirror=True, use_proportional_edit=False,
    proportional_edit_falloff='SMOOTH', proportional_size=1,
    use_proportional_connected=False, use_proportional_projected=False)
17 bpy.ops.object.editmode_toggle()
18 bpy.ops.object.select_all(action='DESELECT')
19 bpy.context.object.active_material_index = 0
20 return {'FINISHED'}
```

3.4 Exportovanie

Pri exportovaní je najprv získaná cesta do adresára do ktorého chceme exportovať budovy. Následne je v tomto adresári otvorený alebo vytvorený súbor do ktorého bude uložená premenná obsahujúca počet modelov, ktoré budú exportované. Je taktiež vytvorené počítadlo ktoré zaznamenáva počet exportovaných modelov. Následne je získaný odkaz na kolekciu v ktorej sa nachádzajú modely budov. Pre každý model je vykonaný kód ktorý zvýši počítadlo o jeden, zvolí daný objekt,

vyexportuje ho s menom, ktoré je rovnaké ako hodnota počítadla a zvolenie modelu je zrušené. Po tom čo boli exportované všetky budovy je zapísaná premenná počítadla do otvoreného súboru a daný súbor je zatvorený.

```
class ExportOperator(bpy.types.Operator):
"""Tooltip"""
bl_idname = "export.1"
bl_label = "Simple Object Operator"
def execute(self, context):
scene = context.scene
myProps = scene.my_props

destfolder = myProps.exportPath
f1= open(destfolder+"code.js","w+")
x = int(0)
sce = bpy.data.collections['buildings'].all_objects
for ob in sce:
x+=1
bpy.ops.object.select_pattern(pattern = ob.name)
bpy.ops.export_scene.gltf(filepath = destfolder + str(x) + ".glb",
 use_selection = True)
bpy.ops.object.select_all(action='DESELECT')

f1.write("let amountOfBuildings = " + str(x))

f1.close()
return {'FINISHED'}
```

3.5 Segmentácia

Tento kód vykonáva zoskupenie budov do vybraného počtu segmentov. Tu je najprv použitá knižnica numpy, konkrétne jej funkcia $arraysplit$, ktorá rozdelí kolekciu obsahujúcu budovy do samostatných polí. V každom z týchto polí je vybraný objekt a tieto objekty sú následne spojené do jedného. Následne je vybranie tohoto spojeného objektu zrušené

```
class SeparationOperator(bpy.types.Operator):
"""Tooltip"""
bl_idname = "separation.1"
bl_label = "Simple Object Operator"
def execute(self, context):
scene = context.scene
myProps = scene.my_props
```

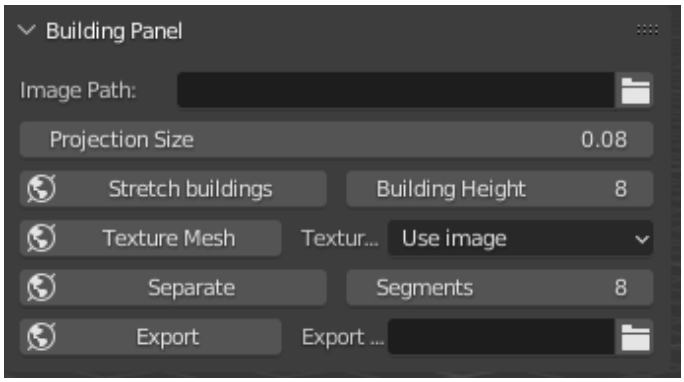
```
8   X = numpy.array_split(bpy.data.collections['buildings'].all_objects,
      myProps.partsToSeparate)
9   for part in X:
10  for obj in part:
11  obj.select_set(True)
12  bpy.context.view_layer.objects.active = obj
13  bpy.ops.object.join()
14  bpy.ops.object.select_all(action='DESELECT')
15
16  return {'FINISHED'}
```

3.6 Panel v užívateľskom rozhraní

Tento kód slúži na vytvorenie panela v užívateľskom rozhraní programu Blender.

```
1   class BuildingPanel(bpy.types.Panel):
2   """Creates a Panel in the Object properties window"""
3   bl_label = "Building Panel"
4   bl_idname = "OBJECT_PT_building"
5   bl_space_type = 'VIEW_3D'
6   bl_region_type = 'UI'
7   bl_category = "Buildings"
8   def draw(self, context):
9   layout = self.layout
10  scene = context.scene
11  myProps = scene.my_props
12  obj = context.object
13
14  layout.prop(myProps, "pathToTextures")
15  layout.prop(myProps, "cubeProjectionSize")
16
17  row1 = layout.row()
18  row1.operator(BuildingOperator.bl_idname, text="Stretch buildings",
      icon='WORLD_DATA')
19  row1.prop(myProps, "buildingSize")
20
21  row2 = layout.row()
22  row2.operator(SinkOperator.bl_idname, text="Texture Mesh", icon='
      WORLD_DATA')
23  row2.prop(myProps, "operationChoice")
24
25  row3 = layout.row()
26  row3.operator(SeparationOperator.bl_idname, text="Separate", icon='
      WORLD_DATA')
```

```
27  row3.prop(myProps, "partsToSeparate")
28
29  row4 = layout.row()
30  row4.operator(ExportOperator.bl_idname, text="Export", icon='
      WORLD_DATA')
31  row4.prop(myProps, "exportPath")
```



Obr. 3.1: Používateľské rozhranie doplnku

4 Popis funkcie kódu v aplikácii pre A-Frame

4.1 Komponent na vkladanie stromov do scény

Tento od najprv vytvorí neviditeľný objekt ktorý vysiela raycast priamo smerom nadol. Následne kód skontroluje či sa raycast dotýka objektu a ak nie je zmenená pozícia objektu. Následne je vykonaná kontrola či má raycast, ktorý pretína objekt, správny identifikátor. Ak áno je získané pretnutie raycastu a objektu a je do scény vložený strom. Existujú dve verzie tohoto komponentu. Jedna vkladá sprajty a druhá 3D inštancie stromov.

3D inštancie:

```
1  // This code spawns 3D trees into scene
2  let setTree = true;
3  let boxOrg;
4  let countDown = 0;
5  let treePos;
6  let posNew;
7  let intersection;
8  let tree;
9  let sceneEl;
10 let rayEl;
11 AFRAME.registerComponent('create-trees', {
12   init: function () {
13     this.el.addEventListener('raycaster-intersected', evt => {
14       this.raycaster = evt.detail.el;
15     });
16     this.el.addEventListener('raycaster-intersected-cleared', evt =>
   {
17       this.raycaster = null;
18     });
19     // Create invisible box with raycast shooting directly downwards
```

```
20      boxOrg = document.createElement("a-box");
21      sceneEl = document.querySelector('a-scene');
22      rayEl = document.createElement("a-entity");
23      document.querySelector('a-scene').appendChild(boxOrg);
24      rayEl.setAttribute("raycaster", "showLine: false; far: 150");
25      rayEl.setAttribute('rotation', '-90 0 0');
26      rayEl.setAttribute('id', 'ray2');
27      boxOrg.setAttribute('visible', 'false');
28      boxOrg.appendChild(rayEl);
29      sceneEl.appendChild(boxOrg);
30      boxOrg.setAttribute('position', "0 120 0");
31    },
32    tick: function () {
33      // Chech if element is reycasted to, if not change position and
    return
34      if (!this.raycaster) {
35        posNew = "" + Math.floor((Math.random() * 936) - 468) + " " +
    120 + " " + Math.floor((Math.random() * 709) - 358) + "";
36        boxOrg.setAttribute('position', posNew)
37        return;
38      }
39      // Check if raycaster has correct id and is suposed to plant
    tree
40      if(setTree === true && this.raycaster.getAttribute("id") === "
    ray2"){
41        // Get intersection and plant tree at its position
42        intersection = this.raycaster.components.raycaster.
    getIntersection(this.el);
43        // Return if there is none
44        if (!intersection) { return; }
45        treePos = intersection.point;
46        posNew = "" + Math.floor((Math.random() * 936) - 468) + " " +
    120 + " " + Math.floor((Math.random() * 709) - 358) + "";
47        boxOrg.setAttribute('position', posNew)
48        makeBox(treePos,"tree"+ countDown)
49        countDown++;
50        if(countDown >= 300){
51          countDown = 0;
52          setTree = false;
53          boxOrg.parentNode.removeChild(boxOrg);
54        }
55      }
56
57    }
58  });
```

```
59  // Create element that is instance of 3D tree mesh
60  function makeBox(treePos, id){
61    tree = document.createElement("a-entity");
62    // Set it as an instance
63    tree.setAttribute('instanced-mesh-member', "mesh:#treeInst");
64    // Randomize Scale and Rotation a little bit to create ilusion of
      diversity od trees
65    tree.setAttribute('scale', ""+ (Math.random() * 3 + 4) + " " + (
      Math.random() * 5 + 3) + " " + (Math.random() * 3 + 4) + "");
66    tree.setAttribute('rotation', "" + (0 + Math.random() * 10) + " "
      + Math.random() * 360 + " "+ (0 + Math.random() * 10) + "");
67    // Give tree unique ID
68    tree.setAttribute('id', id);
69
70    // Give tree position
71    tree.setAttribute('position', treePos);
72    document.querySelector('a-scene').appendChild(tree);
73  }
```

2D Sprajty:

```
1   // This code spawns sprite trees into scene
2   let setTree = true;
3   let boxOrg;
4   let countDown = 0;
5   let treePos;
6   let posNew;
7   let intersection;
8   let tree;
9   let sceneEl;
10  let rayEl;
11  AFRAME.registerComponent('create-trees', {
12    init: function () {
13      this.el.addEventListener('raycaster-intersected', evt => {
14        this.raycaster = evt.detail.el;
15      });
16      this.el.addEventListener('raycaster-intersected-cleared', evt =>
      {
17        this.raycaster = null;
18      });
19      // Create invisible box with raycast shooting directly downwards
20      boxOrg = document.createElement("a-box");
21      sceneEl = document.querySelector('a-scene');
22      rayEl = document.createElement("a-entity");
23      document.querySelector('a-scene').appendChild(boxOrg);
24      rayEl.setAttribute("raycaster", "showLine: false; far: 150");
```

```
25      rayEl.setAttribute('rotation', '-90 0 0');
26      rayEl.setAttribute('id', 'ray2');
27      boxOrg.setAttribute('visible', 'false');
28      boxOrg.appendChild(rayEl);
29      sceneEl.appendChild(boxOrg);
30      boxOrg.setAttribute('position', "0 120 0");
31    },
32    tick: function () {
33      // Chech if element is raycasted to, if not change position and
    return
34      if (!this.raycaster) {
35        posNew = "" + Math.floor((Math.random() * 936) - 468) + " " +
    120 + " " + Math.floor((Math.random() * 709) - 358) + "";
36        boxOrg.setAttribute('position', posNew)
37        return;
38      }
39      // Check if raycaster has correct id and is suposed to plant
    tree
40      if(setTree === true && this.raycaster.getAttribute("id") === "
    ray2"){
41        // Get intersection and plant tree at its position
42        intersection = this.raycaster.components.raycaster.
    getIntersection(this.el);
43        // Return if there is none
44        if (!intersection) { return; }
45        treePos = intersection.point;
46        treePos.y = (intersection.point.y + 2);
47        posNew = "" + Math.floor((Math.random() * 936) - 468) + " " +
    120 + " " + Math.floor((Math.random() * 709) - 358) + "";
48        boxOrg.setAttribute('position', posNew)
49        makeBox(treePos,"tree"+ countDown)
50        countDown++;
51        if(countDown >= 300){
52          countDown = 0;
53          setTree = false;
54          boxOrg.parentNode.removeChild(boxOrg);
55        }
56      }
57
58    }
59  });
60  // Create plane with image of tree and set transparency
61  function makeBox(treePos){
62    tree= document.createElement("a-plane");
63    document.querySelector('a-scene').appendChild(tree);
```

```
64     tree.setAttribute('src', "img/tree"+ Math.floor((Math.random() *
       3) + 1) +".png");
65     tree.setAttribute('transparent', "true");
66     tree.setAttribute('scale', "5 5 5");
67     tree.setAttribute('rotate-sprite', '');
68     tree.setAttribute('position', treePos);
69   }
```

4.2 Komponent na rotáciu sprajtov

Tento komponent získava referenciu na kameru v scéne a otáča sprajt na ktorom sa nachádza na základe osy Y danej kamery.

```
1  // This code serves purpose of rotating tree sprites
2  let cameraRot;
3  let el;
4  // Component for rotation of sprite in users direction
5  AFRAME.registerComponent('rotate-sprite', {
6    init: function () {
7      // Get reference to users camera element
8      cameraRot = this.el.sceneEl.camera.el;
9    },
10   tick: function () {
11     // Rotate this sprite on Y axis of users camera
12     this.el.setAttribute('rotation', "0 "+ cameraRot.getAttribute('
     rotation').y +" 0");
13   }
14 });
```

4.3 Komponent na vkladanie budov do scény

Tento komponent slúži na vkladanie budov do scény.

```
1  // This code serves purpose of spawning buildings
2  let sceneElm;
3  let buildingAsIt;
4  let buildingEl;
5  AFRAME.registerComponent('spawn-buildings', {
6    init: function () {
7      console.log(amountOfBuildings)
8      for (let i = 1; i <= amountOfBuildings; i++) {
9        // Get reference to scene
10       sceneElm = document.querySelector('a-scene');
```

```
11        // Create elements
12        buildingAsIt = document.createElement("a-asset-item");
13        buildingEl = document.createElement("a-entity");
14        // Set attributes of asset element
15        buildingAsIt.setAttribute('id', ''+ i +'');
16        buildingAsIt.setAttribute('src', './models/'+ i +'.glb');
17        // Append asset to scene
18        sceneElm.appendChild(buildingAsIt);
19        // Set attributes of building element
20        buildingEl.setAttribute('id', '#'+ i +'');
21        buildingEl.setAttribute('gltf-model', '#'+ i +'');
22        buildingEl.setAttribute('scale', '1 1 1');
23        buildingEl.setAttribute('position', '0 0 0');
24        // Check if collision on buildings is allowed
25        if(localStorage.getItem('allowCollision') === 'true'){
26          buildingEl.setAttribute('static-body', 'shape: mesh');
27        }
28        // Append building to scene
29        sceneElm.appendChild(buildingEl);
30      }
31    },
32  });
```

4.4 Komponent na pohyb po teréne

Tento komponent zabezpečuje pohyb po teréne tak že vysiela raycast lúč od používateľa smerom nadol a ak lúč pretne terén je používateľ premiestnený nad daný terén. Taktiež je možné prepínať medzi pohybom po teréne a lietaním nad terénom.

```
1   // This code lets player switch between flying mode and grounded
      mode
2   let fly = true;
3   AFRAME.registerComponent('collider-check', {
4     init: function () {
5       this.el.addEventListener('raycaster-intersected', evt => {
6         this.raycaster = evt.detail.el;
7       });
8       this.el.addEventListener('raycaster-intersected-cleared', evt =>
      {
9         this.raycaster = null;
10      });
11    },
12    tick: function () {
```

```
13      // Chech if element is reycasted to, if not return
14      if (!this.raycaster) { return; }
15      // Check if raycaster has correct id
16      if(this.raycaster.getAttribute('id') === 'ray1'){
17        // Get intersection and set user position at its Y position
18        let intersection = this.raycaster.components.raycaster.
    getIntersection(this.el);
19          // Return if there is nointersection
20          if (!intersection) { return; }
21          let playerPos = document.getElementById('player').getAttribute
    ('position');
22          playerPos.y = (intersection.point.y + 0.1);
23          document.getElementById('player').setAttribute('position',
    playerPos);
24      }
25    }
26  });
27  // Add event listener on space key
28  document.addEventListener('keydown', function(event) {
29    if(event.key == ' ') {
30      // If flying is allowed disallow it and create raycast from user
     aimed straight down to check for ground
31      // If it is not allowed, set it to allowed
32      if(fly === true){
33        var rayElPlay = document.createElement("a-entity");
34
35          rayElPlay.setAttribute("raycaster", "showLine: false; far:
    150");
36          rayElPlay.setAttribute('rotation', '-90 0 0');
37          rayElPlay.setAttribute('id', 'ray1');
38
39          document.getElementById('player').appendChild(rayElPlay);
40          fly = false;
41      }else{
42          document.getElementById('ray1').parentNode.removeChild(
    document.getElementById('ray1'));
43          fly = true;
44      }
45    }
46  });
```

4.5 Kód na meranie výkonu

Tento kód slúži na zaznamenávanie snímkov za sekundu v scéne a taktiež na vytváranie a sťahovanie csv súboru ktorý tieto hodnoty obsahuje. Je zaznamenávaných 6000 záznamov každých sto milisekúnd čo by malo zabrať desať minút. Potom čo je počet záznamov 6000 je vytvorený csv súbor, ktorý je následne stiahnutý.

```
1  // This code serves purpose of creating csv file with fps values
2  const fpsArray = new Array();
3  setInterval(function () {
4    if (typeof document !== 'undefined') {
5      // Push FPS to Array
6      fpsArray.push(document.querySelector('.rs-counter-base:nth-child
     (2) .rs-counter-value').innerHTML)
7    }
8    // Stop after array has 3600 values which should take 30 minutes
9    if(fpsArray.length === 3600){
10     createdBlob()
11   }
12 }, 500);
13 // This function creates and downloads blob which can be downloaded
    as csv file
14 function createdBlob() {
15   let csvContent = arrayToCsv(fpsArray)
16   downloadBlob(csvContent, ''+ localStorage.getItem('nameOfDevice')
    +' '+ document.title +' fps values.csv', 'text/csv;charset=utf-8;')
17 }
18 // Separate data with '\n', so they are in separate rows
19 function arrayToCsv(data){
20   return data.join('\n')
21 }
22 // This function creates and downloads blob which can be downloaded
    as csv file
23 function downloadBlob(content, filename, contentType) {
24   // Create a blob
25   var blob = new Blob([content], { type: contentType });
26   var url = URL.createObjectURL(blob);
27   // Create a link to download it
28   var pom = document.createElement('a');
29   pom.href = url;
30   pom.setAttribute('download', filename);
31   pom.click();
32 }
```

4.6 Kód statickej stránky v Js

Tento kód slúži na zaznamenávanie mena zariadenia pri testovaní a taktiež na zapamätanie si či je povolená kolízia na budovách alebo nie.

```
1  // This is code that stores, gets and sets values of device name and
     checkbox of whether building collision is allowed
2  window.addEventListener("load", isCheckedStart);
3  let allowCollision = false;
4  let checkBox;
5  // Check whether collision localstorage is null or not and if is set
     it to allowed
6  if(localStorage.getItem('allowCollision') === null){
7    localStorage.setItem('allowCollision', true);
8  }
9  // Load value form collision local storage
10 collision = localStorage.getItem('allowCollision');
11 // This function saves name of device that is read from form
12 function saveName()
13 {
14   var input = document.getElementById("userInput");
15   localStorage.setItem('nameOfDevice', input.value);
16 }
17 // Sets value of form
18 function setValueOfForm() {
19   document.getElementById("userInput").value = localStorage.getItem
    ('nameOfDevice');
20 }
21 // Onload function of setting collision checkbox to value from local
     storage
22 function isCheckedStart() {
23   checkBox = document.getElementById("myCheck");
24   var text = document.getElementById("text");
25   if (collision === 'true'){
26     text.style.display = "block";
27     checkBox.checked = true
28   } else {
29     text.style.display = "none";
30     checkBox.checked = false
31   }
32 }
33 // Set local storage value to value of checkbox
34 function checkCheckBox() {
35   var text = document.getElementById("text");
36   if (checkBox.checked == true){
```

```
37        text.style.display = "block";
38        localStorage.setItem('allowCollision', true);
39      } else {
40        text.style.display = "none";
41        localStorage.setItem('allowCollision', false);
42      }
43    }
```

4.7 Kód statickej stránky v HTML

Tento kód slúži na vytváranie grafického rozhrania pre navigáciu medzi rôznymi výškami v testovacích scénach a taktiež pre zapamätanie si mena zariadenia a toho či je povolená kolízia.

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4   <meta charset="utf-8">
5   <title>A-frame</title>
6   <meta name="description" content="A-frame">
7   <script src="js/index.js"></script>
8   </head>
9   <body onload="setValueOfForm()">
10  <label for="fname">Meno zariadenia</label><br>
11  <form id="form" onsubmit="return false;">
12  <input type="text" id="userInput">
13  <input type="submit" onclick="saveName()">
14  <label for="myCheck">Collision allowed:</label>
15  <input type="checkbox" id="myCheck" onclick="checkCheckBox()">
16  <p id="text" style="display:none">Building collision is ACTIVE!</p>
17  </form>
18  <a href="level1.html">Free Roam</a>
19  <br>
20  <a href="test-ground.html">test ground</a>
21  <a href="test-air.html">test air</a>
22  <a href="test-air-high.html">test air high</a>
23  </body>
24  </html>
```

4.8 Kód scény v HTML

Vzhľadom na to že kódy všetkých testovacích scén sú rovnaké alebo podobné, bude uvedený len kód jednej scény. Rozdieli medzi testovacími scénami sú hlavne

v importovaných skriptoch, budovách a vo výškach trate nad terénom po ktorej sa kamera pohybuje. V tomto kóde sú priamo importované modely ciest, terénu a mapy stromového porastu vzhľadom na to že sú vo všetkých scénach rovnaké.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Level 1 Buildings Split in Two Air</title>
6  <meta name="description" content="Level 1 Buildings Split in Two Air
    ">
7  <script src="https://aframe.io/releases/1.0.4/aframe.min.js"></
    script>
8  <script src="https://cdn.jsdelivr.net/gh/n5ro/aframe-physics-
    system@v4.0.1/dist/aframe-physics-system.min.js"></script>
9  <script src="https://unpkg.com/aframe-physics-extras/dist/aframe-
    physics-extras.min.js"></script>
10 <script src="https://cdn.jsdelivr.net/gh/donmccurdy/aframe-extras@v6
    .1.1/dist/aframe-extras.min.js"></script>
11 <script src="https://rawgit.com/protyze/aframe-curve-component/
    master/dist/aframe-curve-component.min.js"></script>
12 <script src="https://rawgit.com/protyze/aframe-alongpath-component/
    master/dist/aframe-alongpath-component.min.js"></script>
13 <script src="https://unpkg.com/aframe-orbit-controls@1.2.0/dist/
    aframe-orbit-controls.min.js"></script>
14 <script src="js/playerGravity.js"></script>
15 <script src="js/plantTreeSprites.js"></script>
16 <script src="js/spriteRotate.js"></script>
17 <script src="models/code.js"></script>
18 <script src="js/spawnBuildings.js"></script>
19 <script src="js/worker.js"></script>
20 </head>
21 <body>
22 <a-scene renderer="sortObjects: true;" stats background="color:
    skyblue" physics="debug: false; gravity: 1" >
23 <template id="avatar-template">
24 <a-entity class="avatar">
25 <a-obj-model src="#hmd-obj" mtl="#hmd-mtl" scale="0.5 0.5 0.5"></a-
    obj-model>
26 <a-sphere class="head" color="#A0A0B2" scale="0.2 0.2 0.2" position
    ="0 -0.08 0" random-color opacity="0.8">
27 </a-sphere>
28 <a-text class="idtext" rotation="0 180 0" position="0 0.3 0" color="
    cyan" align="center" scale="0.8 0.8 0.8" value="LIRKIS USER"></a-
    text>
```

```
29  </a-entity>
30  </template>
31  <template id="avatar-cBottemplate">
32  <a-entity class="cBotavatar">
33  <a-obj-model src="#hmd-obj" mtl="#hmd-mtl" scale="0.5 0.5 0.5"></a-
     obj-model>
34  <a-cylinder class="head" color="#9d5300" scale="0.2 0.4 0.2"
     position="0 -0.08 0" opacity="0.8"></a-cylinder>
35  <a-text class="idtext" rotation="0 180 0" position="0 0.3 0" color="
     cyan" align="center" scale="0.8 0.8 0.8" value="LIRKIS USER"></a-
     text>
36  </a-entity>
37  </template>
38  <template id="avatar-body">
39  <a-entity class="bodyavatar">
40  <a-obj-model src="#body-obj" mtl="#body-mtl" scale="0.05 0.04 0.05 "
      opacity="0.7">
41  </a-obj-model>
42  </a-entity>
43  </template>
44  <template id="avatar-cBotbody">
45  <a-entity class="cBotbodyavatar">
46  <a-obj-model src="#cBotbody-obj" mtl="#cBotbody-mtl" scale="0.5 0.5
     0.5" opacity="0.7"></a-obj-model>
47  </a-entity>
48  </template>
49  <template id="avatar-bodyhands">
50  <a-entity class="bavatar">
51  <a-obj-model src="#body-obj" mtl="#body-mtl" position="-0.3 0.7 0"
     scale="0.015 0.015 0.015"
52  opacity="0.7">
53  </a-obj-model>
54  <a-obj-model src="#body-obj" mtl="#body-mtl" position="0.3 0.7 0"
     scale="0.015 0.015 0.015"
55  opacity="0.7">
56  </a-obj-model>
57  </a-entity>
58  </template>
59  <template id="avatar-cBotbodyhands">
60  <a-entity class="cBotbavatar">
61  <a-obj-model src="#body-obj" mtl="#body-mtl" color="#9d5300"
     position="-0.2 0.7 0"
62  scale="0.01 0.01 0.01"
63  opacity="0.7">
64  </a-obj-model>
```

```
65  <a-obj-model src="#body-obj" mtl="#body-mtl" color="#9d5300"
     position="0.2 0.7 0" scale="0.01 0.01 0.01"
66  opacity="0.7">
67  </a-obj-model>
68  </a-entity>
69  </template>
70  <!-- Cbot dirty location -->
71  <template id="dirty-location">
72  <a-entity scale="1 0.2 1" geometry="primitive: box">
73  </a-entity>
74  </template>
75  <!-- END OF TEMPLATES
     ---------------------------------------------------------------------------

76  <!-- SHARED ENTITIES
     ---------------------------------------------------------------------------

77  <!-- User -->
78  <a-curve id="track1">
79  <a-curve-point position="-35.089484602097 20 79.96670888664927"></a-
     curve-point>
80  <a-curve-point position="-26.075668836153 20 75.2091565702513"></a-
     curve-point>
81  <a-curve-point position="-17.009545872420 20 69.68833974239551"></a-
     curve-point>
82  <a-curve-point position="-7.7876194032172 20 64.21345807909861"></a-
     curve-point>
83  <a-curve-point position="1.37043480985688 20 59.04124002388851"></a-
     curve-point>
84  <a-curve-point position="10.2229658287991 20 54.126667313234016"></a
     -curve-point>
85  <a-curve-point position="19.0671773103207 20 49.22561282697069"></a-
     curve-point>
86  <a-curve-point position="27.8540935676583 20 44.393345458551906"></a
     -curve-point>
87  <a-curve-point position="36.6683868069174 20 39.546022421852044"></a
     -curve-point>
88  <a-curve-point position="45.5790470244197 20 34.64570343176871"></a-
     curve-point>
89  <a-curve-point position="54.7893939417705 20 29.580575057139544"></a
     -curve-point>
90  <a-curve-point position="63.8229414750903 20 24.606589698684232"></a
     -curve-point>
91  <a-curve-point position="72.9637499467477 20 19.718609394522783"></a
     -curve-point>
```

```
92  <a-curve-point position="81.9850704630897 20 14.943804646319425"></a
      -curve-point>
93  <a-curve-point position="90.9160038679128 20 10.222122164171687"></a
      -curve-point>
94  <a-curve-point position="99.8981774844259 20 5.480350883878529"></a-
      curve-point>
95  <a-curve-point position="108.874359742878 20 0.7294853189440116"></a
      -curve-point>
96  <a-curve-point position="118.174164471707 20 -4.235528839802395"></a
      -curve-point>
97  <a-curve-point position="127.126414059453 20 -9.019713662396322"></a
      -curve-point>
98  <a-curve-point position="136.124027138010 20 -13.81984284985705"></a
      -curve-point>
99  <a-curve-point position="145.188670550494 20 -18.617577541612228"></
      a-curve-point>
100 <a-curve-point position="154.112448486292 20 -23.328706402784768"></
      a-curve-point>
101 <a-curve-point position="162.997586689019 20 -28.00870820299237"></a
      -curve-point>
102 <a-curve-point position="172.031393180492 20 -32.76701692911295"></a
      -curve-point>
103 <a-curve-point position="180.965842151761 20 -37.47299181834667"></a
      -curve-point>
104 <a-curve-point position="190.133965290324 20 -42.302048138465096"></
      a-curve-point>
105 <a-curve-point position="198.511903024053 20 -47.788596872900754"></
      a-curve-point>
106 <a-curve-point position="207.560477220939 20 -41.3642830910304"></a-
      curve-point>
107 <a-curve-point position="214.644362710864 20 -34.10797486442935"></a
      -curve-point>
108 <a-curve-point position="221.032496066313 20 -26.36889026107841"></a
      -curve-point>
109 <a-curve-point position="230.730222856254 20 -23.889156616328293"></
      a-curve-point>
110 <a-curve-point position="239.473823887955 20 -28.924780998250725"></
      a-curve-point>
111 <a-curve-point position="248.469537722488 20 -33.6885018772475"></a-
      curve-point>
112 <a-curve-point position="257.327621499520 20 -38.33164600971315"></a
      -curve-point>
113 <a-curve-point position="266.421858091533 20 -43.11751117866605"></a
      -curve-point>
```

```
114   <a-curve-point position="275.411286329787 20 -47.8754359811269"></a-
        curve-point>
115   <a-curve-point position="284.345998094711 20 -52.60440044721619"></a
        -curve-point>
116   <a-curve-point position="293.283233615071 20 -57.334700686647096"></
        a-curve-point>
117   <a-curve-point position="302.186364164086 20 -62.12362032854561"></a
        -curve-point>
118   <a-curve-point position="311.035294795759 20 -66.89819262685826"></a
        -curve-point>
119   <a-curve-point position="319.879661028428 20 -71.59843248439"></a-
        curve-point>
120   <a-curve-point position="328.770530486449 20 -76.40063987642169"></a
        -curve-point>
121   <a-curve-point position="337.59328181538  20 -81.33855228692552"></a
        -curve-point>
122   <a-curve-point position="346.446027769380 20 -86.26926066065626"></a
        -curve-point>
123   <a-curve-point position="355.520485394801 20 -91.29594957746104"></a
        -curve-point>
124   <a-curve-point position="364.339194138573 20 -96.0452215609452"></a-
        curve-point>
125   <a-curve-point position="373.349760355424 20 -100.88376416741683"></
        a-curve-point>
126   <a-curve-point position="382.969694285858 20 -106.04952834623177"></
        a-curve-point>
127   <a-curve-point position="389.971088702029 20 -113.45531881810189"></
        a-curve-point>
128   <a-curve-point position="384.911971669832 20 -122.34174762246549"></
        a-curve-point>
129   <a-curve-point position="380.017909333593 20 -131.5105795245166"></a
        -curve-point>
130   <a-curve-point position="375.382419266257 20 -140.52366695405007"></
        a-curve-point>
131   <a-curve-point position="370.753740442424 20 -149.5908936005562"></a
        -curve-point>
132   <a-curve-point position="366.028066425203 20 -158.84812638824127"></
        a-curve-point>
133   <a-curve-point position="361.451987122729 20 -168.1077689770048"></a
        -curve-point>
134   <a-curve-point position="356.673404316007 20 -177.8014154590075"></a
        -curve-point>
135   <a-curve-point position="352.172011623116 20 -186.7769351995755"></a
        -curve-point>
```

```
136   <a-curve-point position="347.477125969976 20 -197.81702241408973"></
       a-curve-point>
137   <a-curve-point position="343.646154061395 20 -207.11144661925192"></
       a-curve-point>
138   <a-curve-point position="339.750968439193 20 -217.04414493563328"></
       a-curve-point>
139   <a-curve-point position="335.985110725549 20 -226.92889616178317"></
       a-curve-point>
140   <a-curve-point position="332.246249470733 20 -236.74278616592477"></
       a-curve-point>
141   <a-curve-point position="328.526342245953 20 -246.6249428217284"></a
       -curve-point>
142   <a-curve-point position="324.914805672414 20 -256.66402049139015"></
       a-curve-point>
143   <a-curve-point position="323.154940503125 20 -266.6621330201052"></a
       -curve-point>
144   <a-curve-point position="319.616162035289 20 -276.09876338290195"></
       a-curve-point>
145   <a-curve-point position="316.100969106144 20 -285.4761254976833"></a
       -curve-point>
146   <a-curve-point position="312.780837284685 20 -294.9374832150825"></a
       -curve-point>
147   <a-curve-point position="309.301197128898 20 -304.6505338605183"></a
       -curve-point>
148   <a-curve-point position="305.862197152300 20 -314.25014170697614"></
       a-curve-point>
149   <a-curve-point position="299.495705557334 20 -322.3070669326302"></a
       -curve-point>
150   <a-curve-point position="290.593378237932 20 -327.3289873712514"></a
       -curve-point>
151   <a-curve-point position="280.927766301514 20 -331.82132934315325"></
       a-curve-point>
152   <a-curve-point position="271.418638920856 20 -335.8632838673011"></a
       -curve-point>
153   <a-curve-point position="261.417698648587 20 -338.20749164609595"></
       a-curve-point>
154   <a-curve-point position="251.163309861721 20 -336.6924136461186"></a
       -curve-point>
155   <a-curve-point position="242.520671610039 20 -331.21602059479613"></
       a-curve-point>
156   <a-curve-point position="233.870806476094 20 -326.09743481331515"></
       a-curve-point>
157   <a-curve-point position="225.107647415016 20 -320.9492351954983"></a
       -curve-point>
```

```
158   <a-curve-point position="216.141396610231 20 -315.9981540015132"></a
        -curve-point>
159   <a-curve-point position="207.119641412426 20 -311.0884020075921"></a
        -curve-point>
160   <a-curve-point position="198.098251808821 20 -306.17884897460794"></
        a-curve-point>
161   <a-curve-point position="189.005279179404 20 -301.5187383395102"></a
        -curve-point>
162   <a-curve-point position="179.707114963592 20 -296.81441730135725"></
        a-curve-point>
163   <a-curve-point position="170.839325066653 20 -292.168471224969"></a-
        curve-point>
164   <a-curve-point position="161.898292439362 20 -287.44082263302994"></
        a-curve-point>
165   <a-curve-point position="152.627597448986 20 -282.5388622058016"></a
        -curve-point>
166   <a-curve-point position="143.681730638400 20 -277.8177134158091"></a
        -curve-point>
167   <a-curve-point position="134.457207103796 20 -273.02157402850713"></
        a-curve-point>
168   <a-curve-point position="125.409218490154 20 -268.375434261711"></a-
        curve-point>
169   <a-curve-point position="116.436934608024 20 -263.8287587343157"></a
        -curve-point>
170   <a-curve-point position="107.354582164497 20 -259.25641709012285"></
        a-curve-point>
171   <a-curve-point position="98.2201349429443 20 -254.65784929752934"></
        a-curve-point>
172   <a-curve-point position="89.1813188498021 20 -250.10742522030574"></
        a-curve-point>
173   <a-curve-point position="79.8595934034050 20 -245.39977620949526"></
        a-curve-point>
174   <a-curve-point position="70.8384429087800 20 -240.8356077268566"></a
        -curve-point>
175   <a-curve-point position="61.8165491677990 20 -236.2710632055302"></a
        -curve-point>
176   <a-curve-point position="52.6467494369730 20 -231.63168700307352"></
        a-curve-point>
177   <a-curve-point position="43.6248556959920 20 -227.06714248174717"></
        a-curve-point>
178   <a-curve-point position="34.2569808586955 20 -222.32755199302096"></
        a-curve-point>
179   <a-curve-point position="25.1860328697372 20 -217.73818892413905"></
        a-curve-point>
```

```
180   <a-curve-point position="16.0667738796005 20 -213.12438334661317"></
       a-curve-point>
181   <a-curve-point position="7.06308966946823 20 -208.5690517706683"></a
       -curve-point>
182   <a-curve-point position="-1.9903920345543 20 -203.98852560870412"></
       a-curve-point>
183   <a-curve-point position="-10.963603151178 20 -199.44861161536556"></
       a-curve-point>
184   <a-curve-point position="-19.985496892602 20 -194.8840670938151"></a
       -curve-point>
185   <a-curve-point position="-29.060049759094 20 -190.3007225269697"></a
       -curve-point>
186   <a-curve-point position="-38.071685632372 20 -185.62760451636356"></
       a-curve-point>
187   <a-curve-point position="-47.006669677916 20 -181.02451923682446"></
       a-curve-point>
188   <a-curve-point position="-56.009974599550 20 -176.40132453408955"></
       a-curve-point>
189   <a-curve-point position="-64.984285803707 20 -171.75707383716548"></
       a-curve-point>
190   <a-curve-point position="-73.903029084190 20 -167.13212666418988"></
       a-curve-point>
191   <a-curve-point position="-82.626392336258 20 -162.6084967151385"></a
       -curve-point>
192   <a-curve-point position="-91.657375319970 20 -157.9253459722414"></a
       -curve-point>
193   <a-curve-point position="-100.73678964363 20 -153.23322273156128"></
       a-curve-point>
194   <a-curve-point position="-109.72629958802 20 -148.59975876699383"></
       a-curve-point>
195   <a-curve-point position="-118.72530810168 20 -144.0014907347846"></a
       -curve-point>
196   <a-curve-point position="-127.72913843986 20 -139.40075887097768"></
       a-curve-point>
197   <a-curve-point position="-136.73333968771 20 -134.7998374816108"></a
       -curve-point>
198   <a-curve-point position="-145.74902064852 20 -130.33685798542416"></
       a-curve-point>
199   <a-curve-point position="-154.87805974278 20 -125.81673020537943"></
       a-curve-point>
200   <a-curve-point position="-163.85310677934 20 -121.06173595398276"></
       a-curve-point>
201   <a-curve-point position="-172.95348718767 20 -116.18872065319664"></
       a-curve-point>
```

```
202   <a-curve-point position="-181.35378218327 20 -110.4830877727518"></a
        -curve-point>
203   <a-curve-point position="-189.61948901092 20 -104.46664772382873"></
        a-curve-point>
204   <a-curve-point position="-197.99000347834 20 -97.85020746763982"></a
        -curve-point>
205   <a-curve-point position="-205.69309871600 20 -91.30541185247218"></a
        -curve-point>
206   <a-curve-point position="-212.92789703359 20 -84.17264474293934"></a
        -curve-point>
207   <a-curve-point position="-219.62690980886 20 -76.59960874336814"></a
        -curve-point>
208   <a-curve-point position="-225.98345432757 20 -68.8074793477999"></a-
        curve-point>
209   <a-curve-point position="-232.02581137080 20 -60.838795360878144"></
        a-curve-point>
210   <a-curve-point position="-238.20852028511 20 -52.42272979218389"></a
        -curve-point>
211   <a-curve-point position="-244.13123030100 20 -44.15948490579507"></a
        -curve-point>
212   <a-curve-point position="-249.72295256151 20 -35.53439845470522"></a
        -curve-point>
213   <a-curve-point position="-255.01496260337 20 -26.97000949534125"></a
        -curve-point>
214   <a-curve-point position="-259.86842498871 20 -17.81903003435703"></a
        -curve-point>
215   <a-curve-point position="-264.02016638589 20 -8.412566173365661"></a
        -curve-point>
216   <a-curve-point position="-267.77246528425 20 1.158043495480049"></a-
        curve-point>
217   <a-curve-point position="-270.83466182166 20 10.773552900204704"></a
        -curve-point>
218   <a-curve-point position="-273.71498249485 20 20.61802093751486"></a-
        curve-point>
219   <a-curve-point position="-276.41234346628 20 30.39473662146148"></a-
        curve-point>
220   <a-curve-point position="-278.33762802756 20 40.41546447137214"></a-
        curve-point>
221   <a-curve-point position="-279.81839713998 20 50.519493488918265"></a
        -curve-point>
222   <a-curve-point position="-280.89204064552 20 60.50780364719231"></a-
        curve-point>
223   <a-curve-point position="-281.59780486833 20 70.66697637945244"></a-
        curve-point>
```

```
224   <a-curve-point position="-281.94184493116 20 80.76395823612121"></a-
        curve-point>
225   <a-curve-point position="-282.18286440708 20 90.97793247737164"></a-
        curve-point>
226   <a-curve-point position="-280.81696317472 20 101.132747236694"></a-
        curve-point>
227   <a-curve-point position="-279.24293909351 20 111.32832288851567"></a
        -curve-point>
228   <a-curve-point position="-277.67378688023 20 121.2468472372409"></a-
        curve-point>
229   <a-curve-point position="-275.96173176162 20 131.10180921695775"></a
        -curve-point>
230   <a-curve-point position="-273.63354978154 20 140.9225664702504"></a-
        curve-point>
231   <a-curve-point position="-271.23340730060 20 150.79850411054935"></a
        -curve-point>
232   <a-curve-point position="-268.29465331981 20 160.49715512804062"></a
        -curve-point>
233   <a-curve-point position="-264.55353379634 20 169.92241053430496"></a
        -curve-point>
234   <a-curve-point position="-260.88443649094 20 179.38972260212137"></a
        -curve-point>
235   <a-curve-point position="-257.19763612685 20 188.91338559648963"></a
        -curve-point>
236   <a-curve-point position="-253.45150442434 20 198.5368017142146"></a-
        curve-point>
237   <a-curve-point position="-249.63206499500 20 207.88992854340054"></a
        -curve-point>
238   <a-curve-point position="-245.68903550517 20 217.36343818077336"></a
        -curve-point>
239   <a-curve-point position="-241.75593637615 20 226.65652764216702"></a
        -curve-point>
240   <a-curve-point position="-237.53399597186 20 236.42913425463232"></a
        -curve-point>
241   <a-curve-point position="-233.34122227793 20 245.79533946957164"></a
        -curve-point>
242   <a-curve-point position="-223.86178791761 20 249.0153653126755"></a-
        curve-point>
243   <a-curve-point position="-214.39649704641 20 245.61997523440235"></a
        -curve-point>
244   <a-curve-point position="-204.48763696885 20 241.97261165921736"></a
        -curve-point>
245   <a-curve-point position="-195.00416122203 20 238.457120242983"></a-
        curve-point>
```

```
246   <a-curve-point position="-184.67979140436 20 239.1616224123033"></a-
        curve-point>
247   <a-curve-point position="-177.75623296822 20 246.57583772258195"></a
        -curve-point>
248   <a-curve-point position="-171.50376243367 20 254.86581527104192"></a
        -curve-point>
249   <a-curve-point position="-165.19753403099 20 262.75227225812745"></a
        -curve-point>
250   <a-curve-point position="-158.21138417725 20 270.1340790991023"></a-
        curve-point>
251   <a-curve-point position="-150.60920392483 20 277.21421053307745"></a
        -curve-point>
252   <a-curve-point position="-143.82865487449 20 285.3217567632623"></a-
        curve-point>
253   <a-curve-point position="-135.24435512548 20 292.1138122541844"></a-
        curve-point>
254   <a-curve-point position="-124.90206851834 20 293.0622803829734"></a-
        curve-point>
255   <a-curve-point position="-116.03198972213 20 288.10612567470343"></a
        -curve-point>
256   <a-curve-point position="-107.31109969264 20 282.71622672445636"></a
        -curve-point>
257   <a-curve-point position="-98.625321304686 20 277.43378328848775"></a
        -curve-point>
258   <a-curve-point position="-89.782952036017 20 272.2984560108491"></a-
        curve-point>
259   <a-curve-point position="-80.989796920213 20 267.389223372639"></a-
        curve-point>
260   <a-curve-point position="-71.981590024242 20 262.2973325116498"></a-
        curve-point>
261   <a-curve-point position="-63.030410788768 20 257.18056646167355"></a
        -curve-point>
262   <a-curve-point position="-54.250758521751 20 252.07413132061765"></a
        -curve-point>
263   <a-curve-point position="-44.678582247189 20 247.66974807032412"></a
        -curve-point>
264   <a-curve-point position="-36.406663718931 20 241.7532799243738"></a-
        curve-point>
265   <a-curve-point position="-27.393379069645 20 235.88598072843138"></a
        -curve-point>
266   <a-curve-point position="-18.571526076658 20 230.2661833233182"></a-
        curve-point>
267   <a-curve-point position="-9.4818408792503 20 225.3954185017602"></a-
        curve-point>
```

```
268   <a-curve-point position="-0.6581835196463 20 220.66599322545179"></a
        -curve-point>
269   <a-curve-point position="8.49170609491225 20 215.93958778170722"></a
        -curve-point>
270   <a-curve-point position="17.5977144712955 20 211.2201056721875"></a-
        curve-point>
271   <a-curve-point position="21.6739990104647 20 201.80174504786018"></a
        -curve-point>
272   <a-curve-point position="18.4909256778163 20 191.85360889004122"></a
        -curve-point>
273   <a-curve-point position="14.9946313752023 20 182.28087286221248"></a
        -curve-point>
274   <a-curve-point position="11.3286000724401 20 172.81945865480304"></a
        -curve-point>
275   <a-curve-point position="7.33225901661582 20 163.47207333416915"></a
        -curve-point>
276   <a-curve-point position="2.73030283428988 20 154.52921155804378"></a
        -curve-point>
277   <a-curve-point position="-1.8969120598740 20 145.5094451871002"></a-
        curve-point>
278   <a-curve-point position="-6.5644462945333 20 136.41108488010278"></a
        -curve-point>
279   <a-curve-point position="-11.257258464333 20 127.26345064169591"></a
        -curve-point>
280   <a-curve-point position="-16.045662342923 20 117.85878963354963"></a
        -curve-point>
281   <a-curve-point position="-20.746197580683 20 108.4659954569309"></a-
        curve-point>
282   <a-curve-point position="-25.394148284199 20 99.17827776246907"></a-
        curve-point>
283   <a-curve-point position="-29.925220954741 20 90.14183123487044"></a-
        curve-point>
284   <a-curve-point position="-35.089484602097 20 79.96670888664927"></a-
        curve-point>
285   </a-curve>
286   <a-entity alongpath="curve: #track1; dur: 120000; loop: true" id="
        player" camera look-controls orbit-controls="rotateTo: 0 10 0;
        minDistance: 0.5; maxDistance: 10000"></a-entity>
287   <!-- Light -->
288   <a-entity light="type: ambient; color: #CCC; intensity: 1.7"></a-
        entity>
289   <a-entity light="type: directional; color: #FFF; intensity: 0.2"
        position="500 100 500"></a-entity>
290   <!-- Loading assets -->
```

```
291   <a-asset-item id='treemap' src='./models/treemap.glb'></a-asset-item
       >
292   <a-asset-item id='ground' src='./models/ground.glb'></a-asset-item>
293   <a-asset-item id='roads' src='./models/roads.glb'></a-asset-item>
294   <!-- Map for spawning trees -->
295   <a-entity id='#treemap' gltf-model='#treemap' visible="false" scale
       ='1 1 1' position='0 0 0'></a-entity>
296   <a-entity id='#ground' collider-check static-body='shape: mesh' gltf
       -model='#ground' scale='1 1 1' position='0 0 0'></a-entity>
297   <a-entity id='#roads' gltf-model='#roads' scale='1 1 1' position='0
       0 0'></a-entity>
298   <!-- Component for spawning buildings -->
299   <a-entity spawn-buildings id='buildingHolder'></a-entity>
300   </a-scene>
301   </body>
302   </html>
```

5 Požiadavky

5.1 Technické požiadavky

Doplnok napísaný pre program Blender je kompatibilný so všetkým hárdverom s ktorým je kompatibilný program Blender. Pri tejto práci bol používaný na osobnom počítači Acer Nitro 5 an515-52 s operačným systémom Windows 10 Home.

Aplikácia v rámci A-Frame by mala fungovať na všetkých osobných počítačoch a smart-fónoch ktoré majú webový prehliadač.

5.2 Programové požiadavky

Doplnok vyžaduje mať nainštalovaný program Blender s minimálnou verziou 2.93.

Aplikácia vo webovom rámci A-Frame vyžaduje webový prehliadač. V prípade že je spúšťaná priamo z počítača vyžaduje aj vytvorenie lokálneho servera.

6 Zhodnotenie

Popísaný kód je implementáciou navrhnutého postupu na transformáciu objektov reálneho sveta do prostredia virtuálnej reality. Kód pre scény bol vytvorený za účelom testovania tohto riešenia.