VR SafeScaff

Solution & content description

Virtual Reality training platform for safety education of personnel in the construction industry 'SafeScaff'

NACIONĀLAIS ATTĪSTĪBAS PLĀNS 2020

EIROPAS SAVIENĪBA
Kohēzijas fonds

IEGULDĪJUMS TAVĀ NĀKOTNĒ

VIDZEME UNIVERSITY OF APPLIED SCIENCES

2022

Contents

1. Platform description

The Virtual Reality training platform for safety education of personnel in the construction industry (hereafter 'SafeScaff') is safety training platform used for reduction of training costs & limiting risk related unexpected costs. The VR SafeScaff ensures 3D immersive training experience, that enhances cognitive sense of surrounding abilities and provide individuals with the conditions for learning through physical reflex memory. In contrast with the existing training environments SafeScaff provides hands-on experience that provides the most real-life comparable settings, and event scenarios.

The training mode includes guide-through tutorial with pop-up training descriptions and construction element hologram blueprints for improved memorization with grading module. The multi-user experience provides conditions for team training opportunities and overall increased situational awareness and understanding of team co-work settings

2. Solution content description

The Virtual Reality training platform is created based on the know-how of the Vidzeme University of Applied Sciences VR Lab team and the tools provided by 3rd party vendors.

The third Party solutions include the following:

- VR Interaction framework –basic functionality (e.g., controllers, object selection)
- Rootmotion Final IK – Avatar movements
- EBS (Easy Build System – in-game blueprint creation
- Photon Unity Networking – network functionality
- ParrelSync – testing of multiplayer functionality
- Unity – virtual reality core game engine
- Blender – 3D model creation

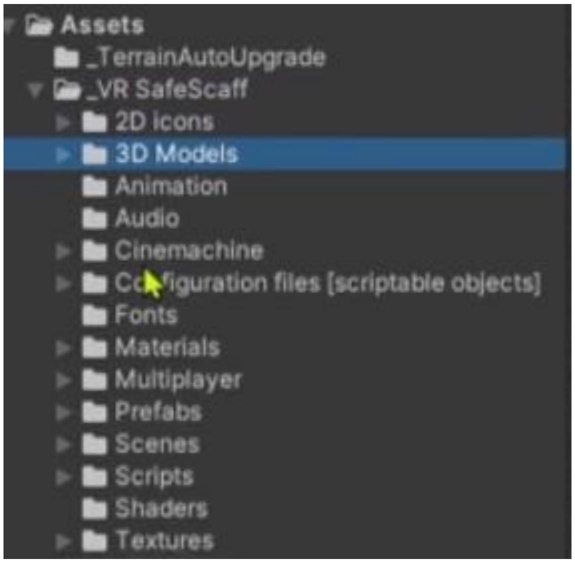
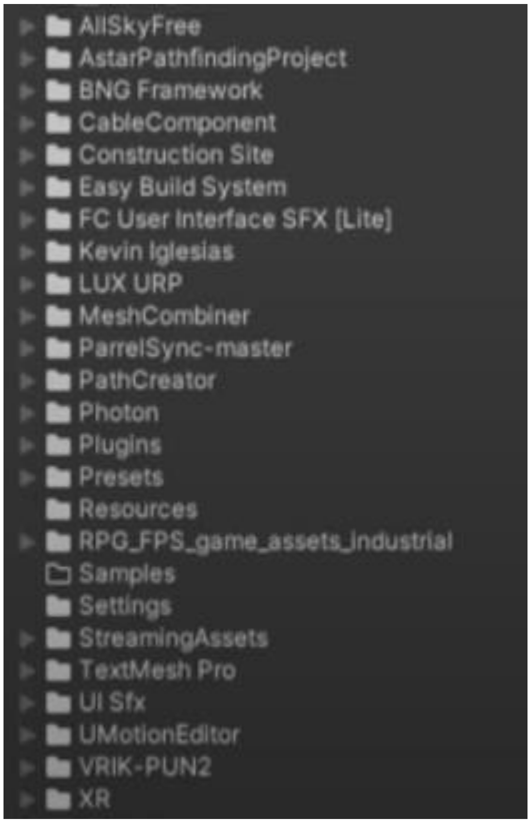


Image 1. **Location of developer files**



AllSkyFree
AstarPathfindingProject
BNG Framework
CableComponent
Construction Site
Easy Build System
FC User Interface SFX [Lite]
Kevin Iglesias
LUX URP
MeshCombiner
ParrelSync-master
PathCreator
Photon
Plugins
Presets
Resources
RPG_FPS_game_assets_industrial
Samples
Settings
StreamingAssets
TextMesh Pro
UI Sfx
UMotionEditor
VRIK-PUN2
XR

Image 2. **Location of Third-party files, plug-ins, asset packs**

The developer files are organized under the VRSafeScaff folder (image 1). Such files include scripts/3D models/ images/. The third-party files (image 2)., plug-ins, asset packs are located under the corresponding third party solution folders .

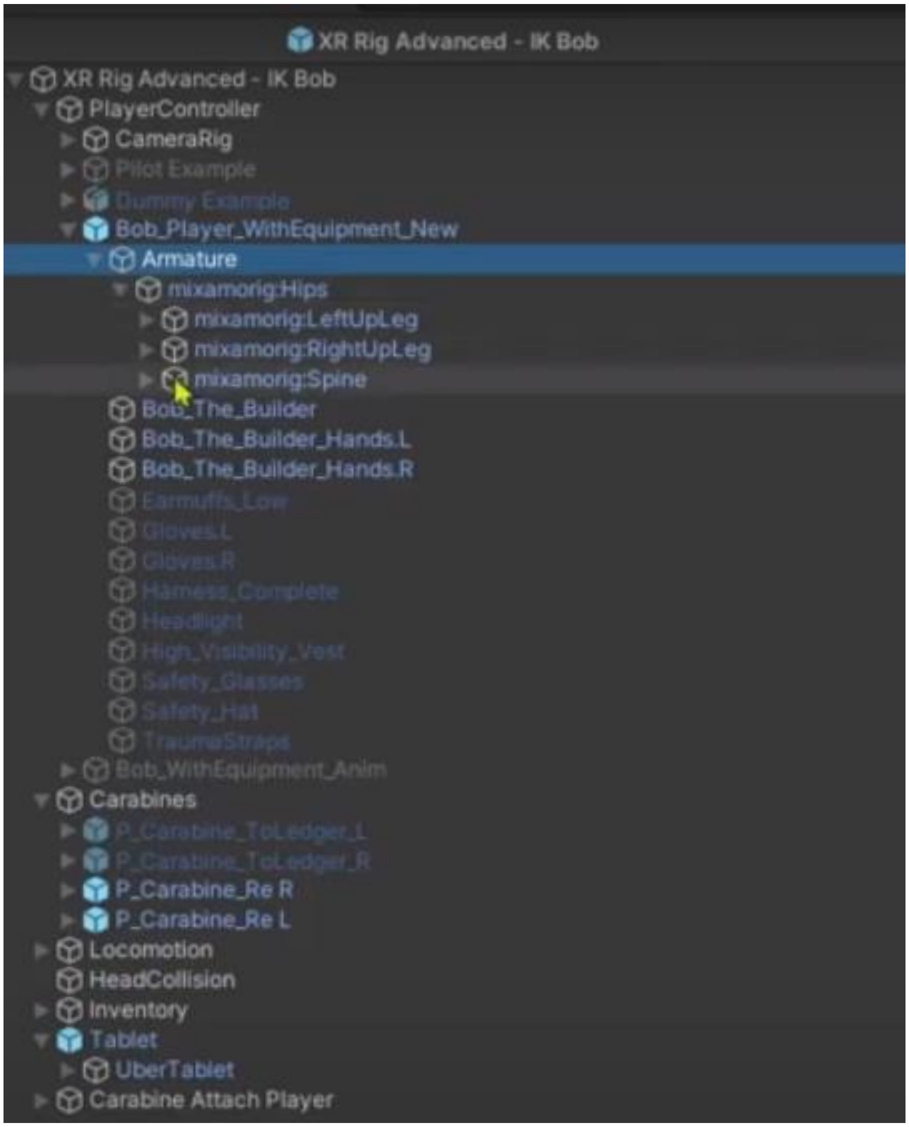
2.1. Character objects

Default VR IF Controller adjusted has been used Multiple objects with specific custom functionality have been added e.g., extra scripts (distance measurer – that calculates player's distance for grading needs; fall damage script – to calculate the height of the height related accident.

1) User avatar script – user avatar object – character equipment toggle script- that controls what equipment is to be activated on the user avatar, with included grading functionality – to check for missing or faulty equipment.

Equipment enabling is triggered by snapzone file objects under armature with scripts that call the necessary logic from the character equipment.

2) Walk animation controller script – used for user avatar limb movement predictions. There are two options – when using controller or other input devices - the built-in animation us used; whilst when using data from the actual user movement trajectories – the VR IK controls the leg placement.



```
XR Rig Advanced - IK Bob
XR Rig Advanced - IK Bob
  PlayerController
    CameraRig
    Pilot Example
    Dummy Example
    Bob_Player_WithEquipment_New
      Armature
        mixamorig:Hips
          mixamorig:LeftUpLeg
          mixamorig:RightUpLeg
          mixamorig:Spine
      Bob_The_Builder
      Bob_The_Builder_Hands.L
      Bob_The_Builder_Hands.R
      Earmuffs_Low
      Gloves.L
      Gloves.R
      Harness_Complete
      Headlight
      High_Visibility_Vest
      Safety_Glasses
      Safety_Hat
      TraumaStraps
    Bob_WithEquipment_Anim
  Carabines
    P_Carabine_ToLedger_L
    P_Carabine_ToLedger_R
    P_Carabine_Re R
    P_Carabine_Re L
  Locomotion
  HeadCollision
  Inventory
  Tablet
    UberTablet
  Carabine Attach Player
```

Image 3, **Character avatar object elements**

2.2. Carabines

Carabines in the in-game settings contain multiple scripts that defines the carabine movement. Scripts provide the logic In the case if carabine is attached to the scaffold, then the users character's movements are constrained by the user's distance script, that defines the set maximum distance from the carabine. Every valid (intended to be connected with carabine) scaffolding object has a child object with box collider that limits where the carabine can move while being attached in between objects. Tags and layer sets are utilized for proper functioning.

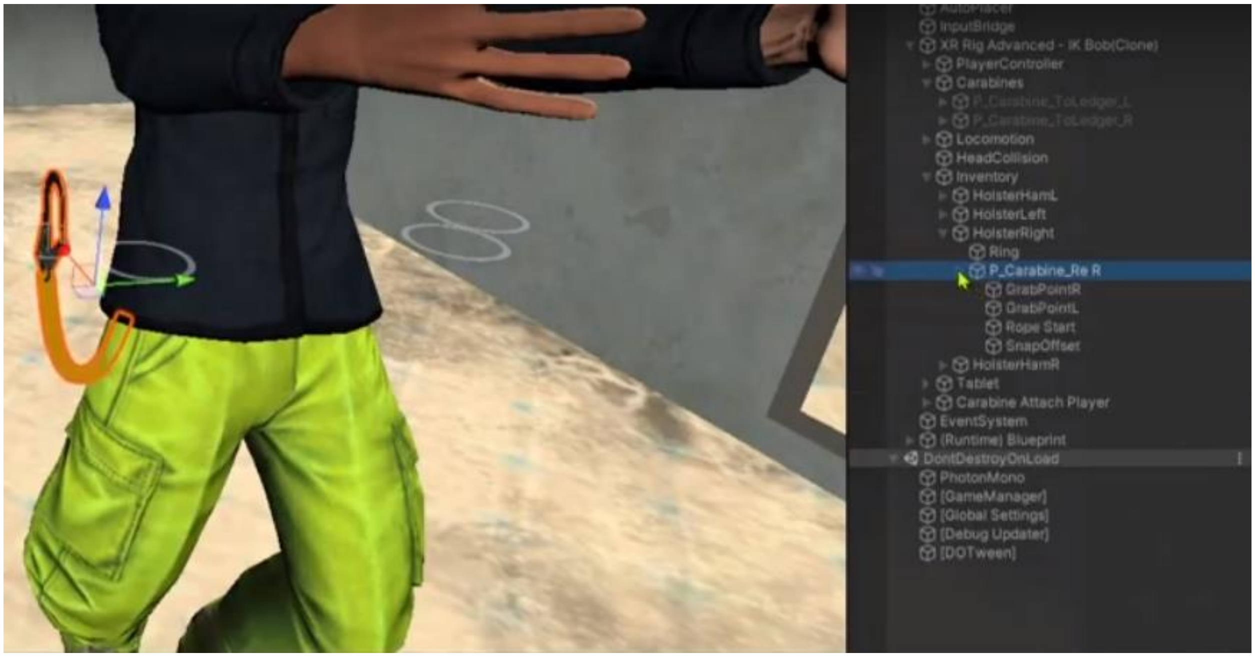


Image 4, **Carabine scripts for audio & visuals**

2.3. Scaffolding objects

Previously created scaffold blueprint prefabs are deployed to the construction site scene with the use of wireframe scaff part script which controls whether and incoming object can be placed in the place of a given blueprint object. It checks if all the necessary connections are placed, the rotation of objects and the damage status of the incoming object.

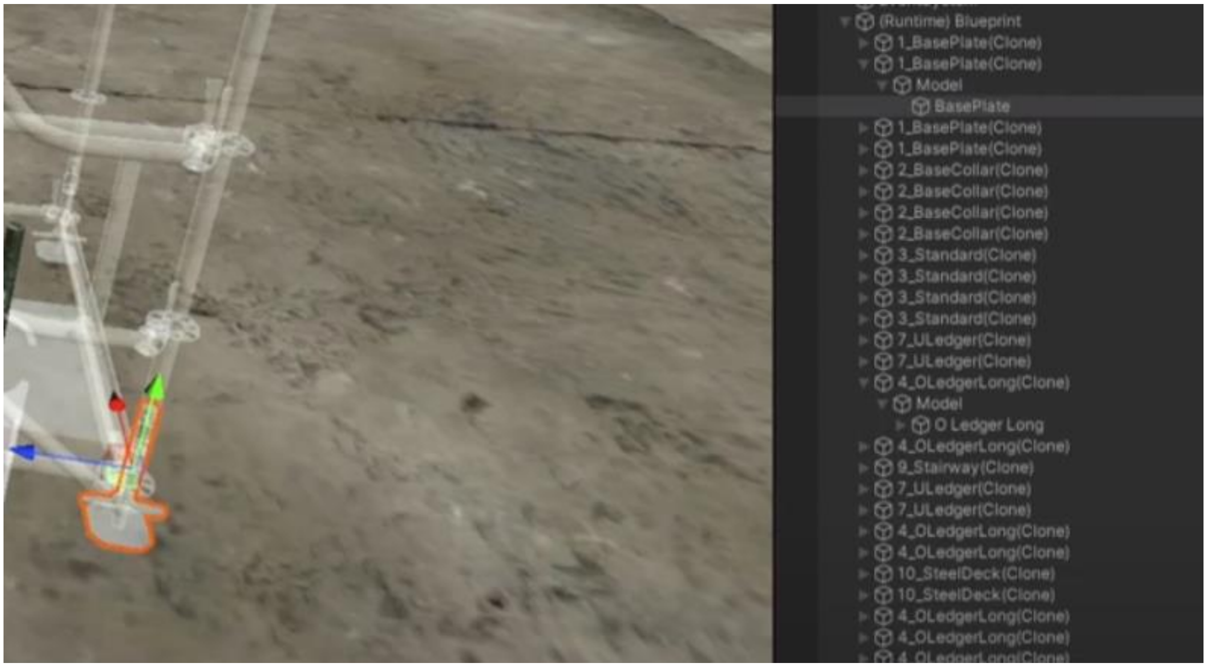


Image 5, **Scaffold blueprint objects**

The blueprint scheme allows for an easy deployment of scaffold elements. The blueprints define different types of elements and their dedicated point of location (image 5).

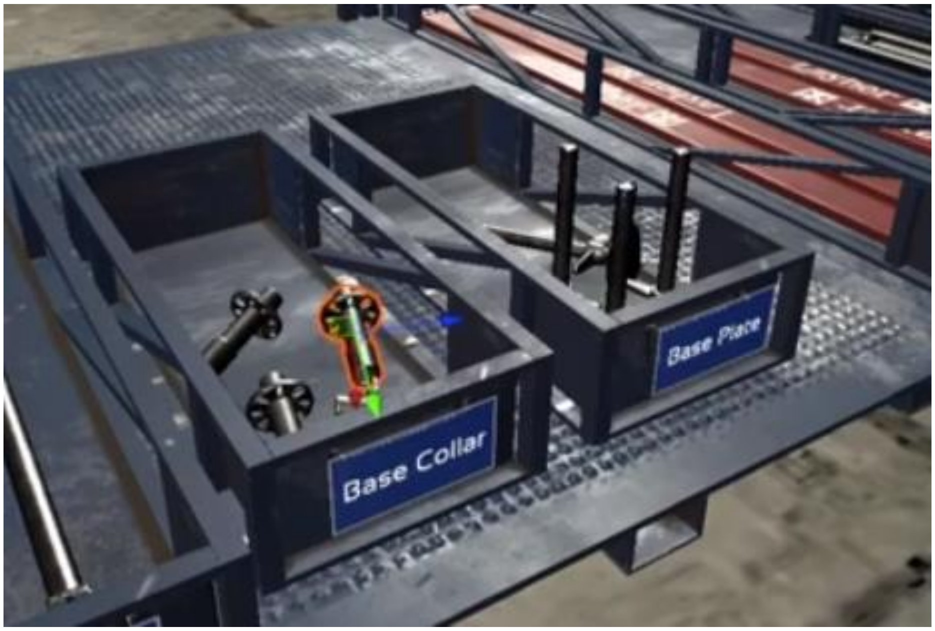
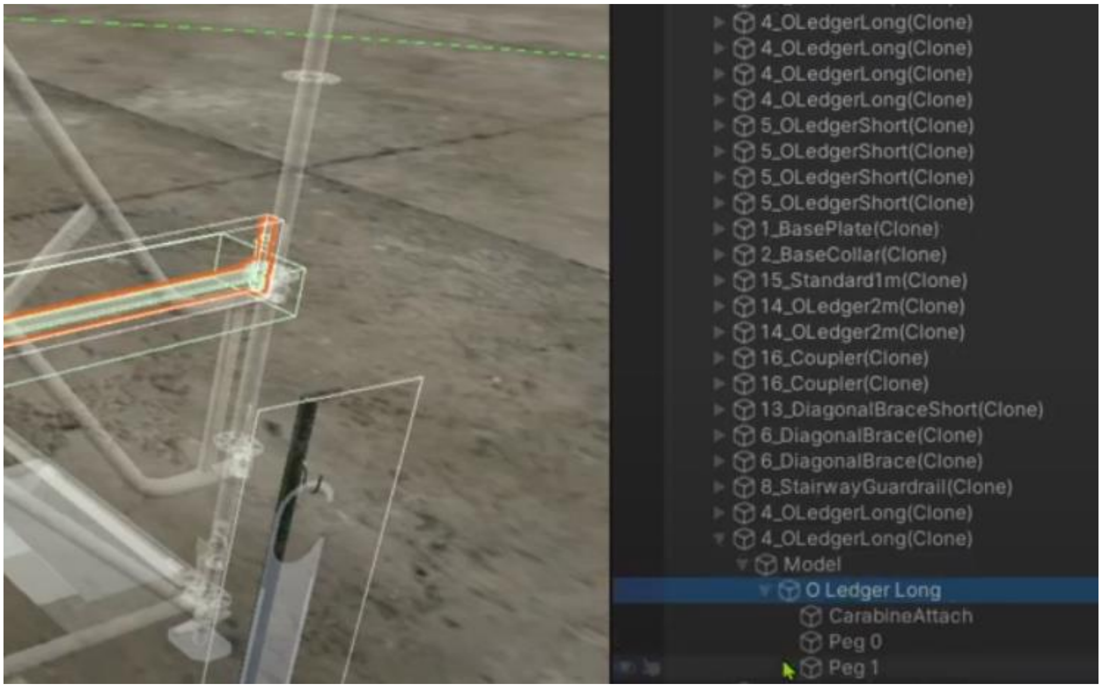


Image 6, **Scaffold part spawn area**

The physical objects that are to be placed on top of the blueprint objects are spawned at the spawn point. The objects are introduced to the scene when the user interacts with the spawn point. (image 6). The spawn point allows for easy collection of scaffold elements and their deployment.



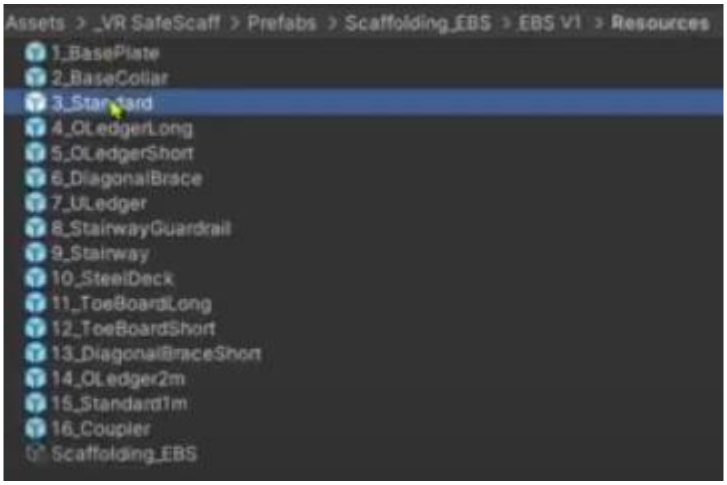
```
4_OLedgerLong(Clone)
4_OLedgerLong(Clone)
4_OLedgerLong(Clone)
4_OLedgerLong(Clone)
5_OLedgerShort(Clone)
5_OLedgerShort(Clone)
5_OLedgerShort(Clone)
5_OLedgerShort(Clone)
1_BasePlate(Clone)
2_BaseCollar(Clone)
15_Standard1m(Clone)
14_OLedger2m(Clone)
14_OLedger2m(Clone)
16_Coupler(Clone)
16_Coupler(Clone)
13_DiagonalBraceShort(Clone)
6_DiagonalBrace(Clone)
6_DiagonalBrace(Clone)
8_StairwayGuardrail(Clone)
4_OLedgerLong(Clone)
4_OLedgerLong(Clone)
  Model
    O Ledger Long
      CarabineAttach
      Peg 0
      Peg 1
```

Image 7, **Scaffold blueprint objects with pin-functionality**

Objects that require pins, have additional child objects attached for proper functioning of each pin. Child objects have scripts that provide object securing logic, visuals and sending state information to the wireframe scaff part script thus ensuring actual securing and further logics. (image 7)

2.4. Scaffolding part prefabs

There are different scaffolding part prefabs whilst using identical models with different functional logics. Each type of prefab models have different functionality and use-case. The available prefabs are used to create blueprint layout and to execute the allocation of real objects over the blueprint objects. The blueprint layout is done via set-up of EBS specific scripts and settings that allow for the objects (with individual object - piece behavioural scripts for overall settings,) to be placed together correctly (socket objects attached defining where the factual scaffold part object can be placed).

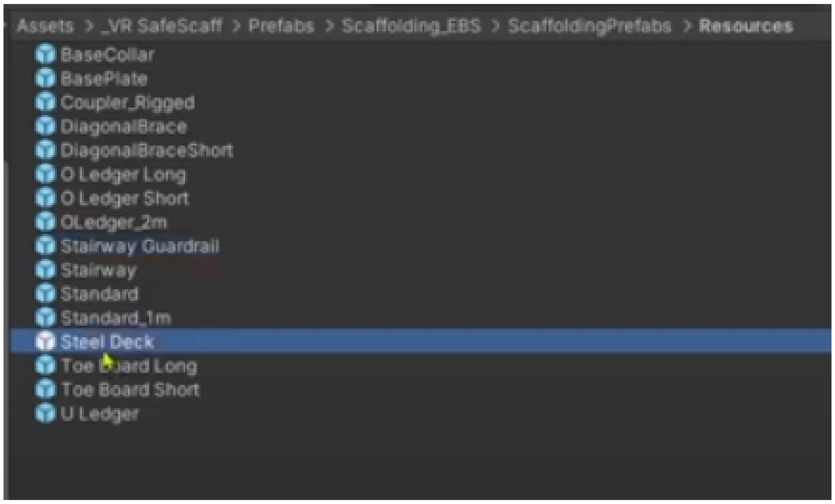


Assets > _VR SafeScaff > Prefabs > Scaffolding_EBS > EBS V1 > Resources

1_BasePlate
2_BaseCollar
3_Standard
4_OLedgerLong
5_OLedgerShort
6_DiagonalBrace
7_ULedger
8_StairwayGuardrail
9_Stairway
10_SteelDeck
11_ToeBoardLong
12_ToeBoardShort
13_DiagonalBraceShort
14_OLedger2m
15_Standard1m
16_Coupler
Scaffolding_EBS

Image 8**, Available prefabs**

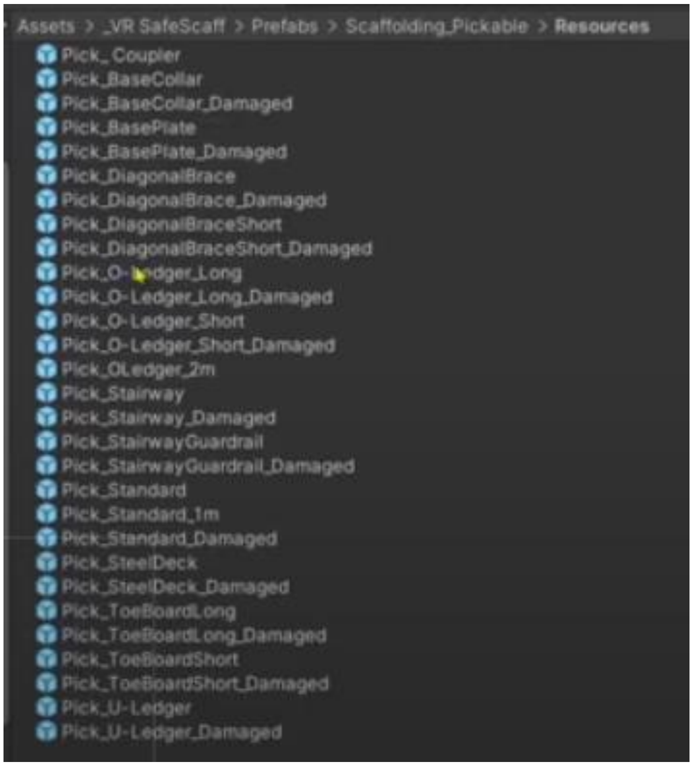
Scaffolding part prefab objects are spawned into the blueprints when starting a scene. The prefab objects are run by *Wireframe Scaff Part* script (image 8).



Assets > _VR SafeScaff > Prefabs > Scaffolding_EBS > ScaffoldingPrefabs > Resources

BaseCollar
BasePlate
Coupler_Rigged
DiagonalBrace
DiagonalBraceShort
O Ledger Long
O Ledger Short
OLedger_2m
Stairway Guardrail
Stairway
Standard
Standard_1m
Steel Deck
Toe Board Long
Toe Board Short
U Ledger

Image 9**, Prefab object prefabs for blueprints**



Assets > _VR SafeScaff > Prefabs > Scaffolding_Pickable > **Resources**

Pick_ Coupler
Pick_BaseCollar
Pick_BaseCollar_Damaged
Pick_BasePlate
Pick_BasePlate_Damaged
Pick_DiagonalBrace
Pick_DiagonalBrace_Damaged
Pick_DiagonalBraceShort
Pick_DiagonalBraceShort_Damaged
Pick_O-Ledger_Long
Pick_O-Ledger_Long_Damaged
Pick_O-Ledger_Short
Pick_O-Ledger_Short_Damaged
Pick_OLedger_2m
Pick_Stairway
Pick_Stairway_Damaged
Pick_StairwayGuardrail
Pick_StairwayGuardrail_Damaged
Pick_Standard
Pick_Standard_1m
Pick_Standard_Damaged
Pick_SteelDeck
Pick_SteelDeck_Damaged
Pick_ToeBoardLong
Pick_ToeBoardLong_Damaged
Pick_ToeBoardShort
Pick_ToeBoardShort_Damaged
Pick_U-Ledger
Pick_U-Ledger_Damaged

Image 10**, Pickable Prefab object for blueprints**

Movable scaffold part objects are spawned into the scene when the user interacts with the part spawner. The objects contain scripts for object pick-up while synchronising their position in network scenarios. There is a custom *Pickable Scaff Part Script.* In order for the objects to function properly they need to have the correct tag and layer (image 10,).

2.5. Forklift object

In single-user mode there is an option to deploy forklift functionality for easing access to the scaffold parts. When building a large multi-story scaffold model the forklift can be called up.

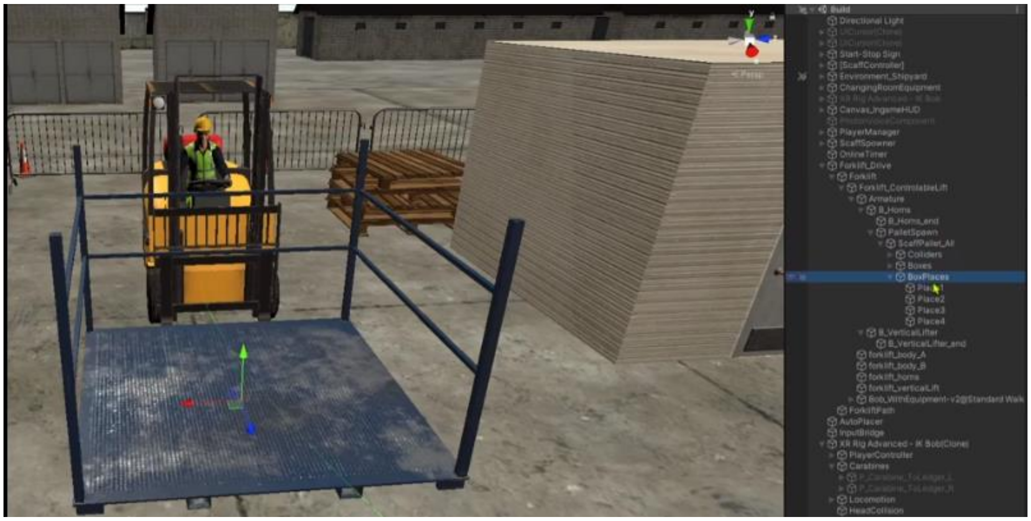
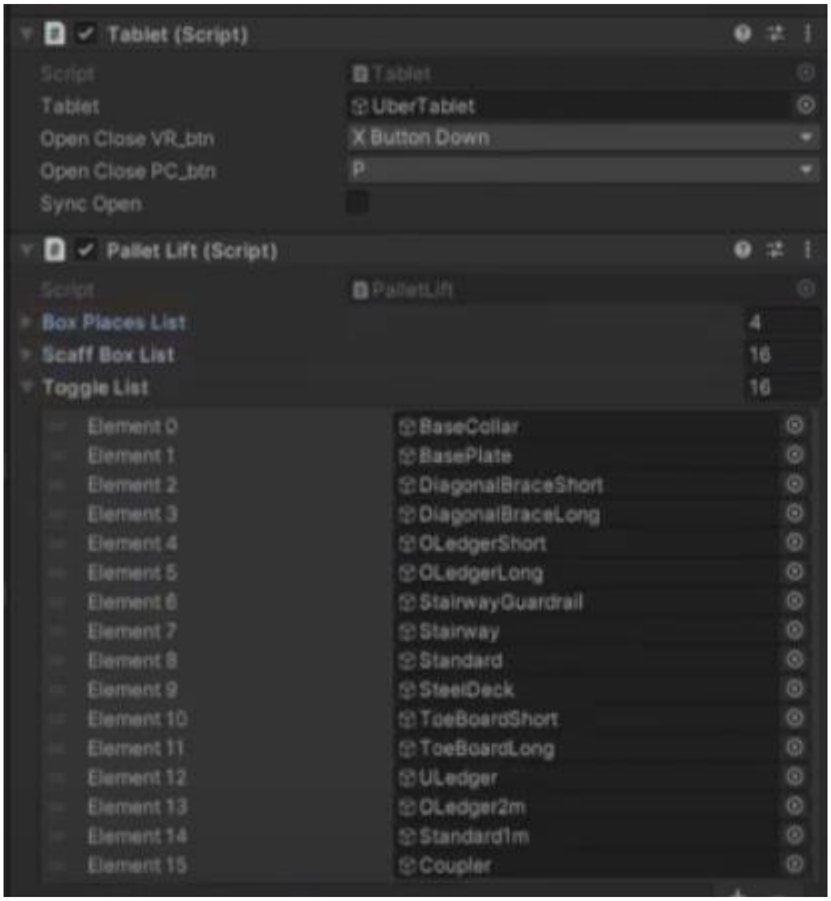


Image 11**, Forklift object**

The forklift (image 11) contains four empty objects where the scaffold part spawners can be placed and the lift can be programmed for a specific height. The forklift is party controlled with the help of *forklift lift controller script* which is connected to the UI and defines the lift position, animation and functioning height limits.



Tablet (Script)

Script	Tablet
Tablet	UberTablet
Open Close VR_btn	X Button Down
Open Close PC_btn	P
Sync Open	

Pallet Lift (Script)

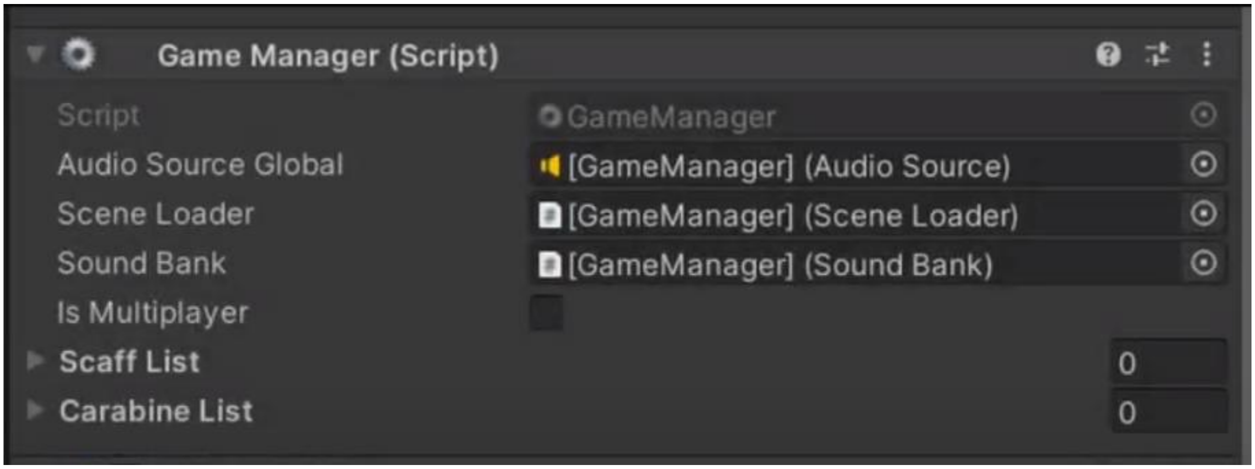
Script	PalletLift
Box Places List	4
Scaff Box List	16
Toggle List	16
Element 0	BaseCollar
Element 1	BasePlate
Element 2	DiagonalBraceShort
Element 3	DiagonalBraceLong
Element 4	OLedgerShort
Element 5	OLedgerLong
Element 6	StairwayGuardrail
Element 7	Stairway
Element 8	Standard
Element 9	SteelDeck
Element 10	ToeBoardShort
Element 11	ToeBoardLong
Element 12	ULedger
Element 13	OLedger2m
Element 14	Standard1m
Element 15	Coupler

Image 12**, Forklift object**

The script controls the lift position animation and the height to which the lift can operate. The tablet object is used in-game to control the (image 12). The *pallet lift script* is connected to the UI and controls which spawner will be displayed.

2.6. Game manager script

The game manager script contains the list of all scaffolding objects. It is used to check whether a scaffold part (object) can be placed, and whether all scaffolding parts are secured.



Game Manager (Script)

Script	GameManager
Audio Source Global	[GameManager] (Audio Source)
Scene Loader	[GameManager] (Scene Loader)
Sound Bank	[GameManager] (Sound Bank)
Is Multiplayer	□
▸ Scaff List	0
▸ Carabine List	0

Image 13**, Game manager script, in-game.**

The manager script (image 13) also is used for playing global audio one-shot effects (e.g., for UI effects, scaffolding placement, securing scaffolding pins, carabine interactions).

3. Scaffold Easy Build System

In order to build a new scaffold the *Easy Build System builder* scene is to be utilized. The scene allows for the necessary scaffold configuration to be created, which is saved to the scaffold blueprint state. When the necessary configuration has been created, the scene blueprint template has to be saved through *Group Behaviour* script *general settings*. The file is saved as JSON file and can be uploaded in the building scene.

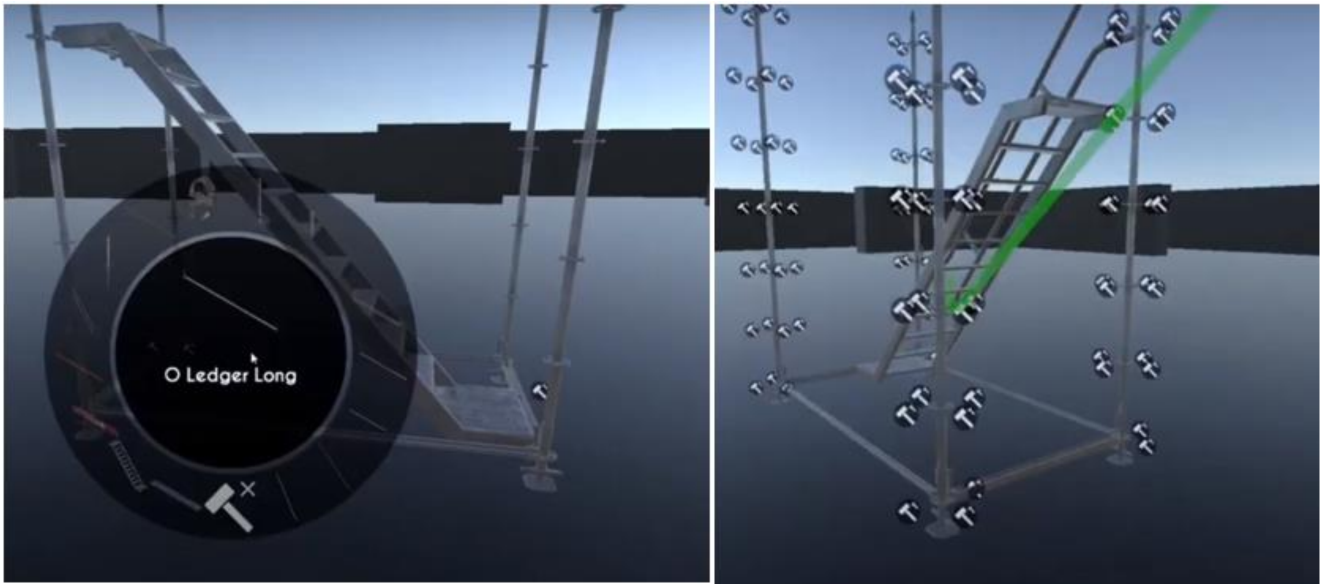


Image 14**, Easy Build System user interface and scaff object deployment**

The Easy Build System user interface allows the user to deploy the necessary size and scale of scaffold blueprints (image 14). The easy-to-use interface ensures that the training environment can be run and set up by industry experts with beginners level of knowledge of the virtual reality environment.

4. SafeScaff grading module

The grading is controlled via safescaff grading script that provides functionality to add grading log entries and to configure them to be showed in the grading user interface (UI) on the tablet object. New grading log can be added with the use of entry code has to be added in the appropriate script. With an appropriately configured line of script, when a condition is met, when the line is executed a log file is created thereafter. The grading controller script can be accessed from anywhere. In order for the grading functionality to work the game manager prefab has to be included into the scene. By default it is added to the subscene and moved along to the other scenes.

5. Multiplayer functionality

The multiplayer functionality allows for simultaneous multiple user interaction in the VR platform environment. The functionality has been implemented using Photon guidelines and examples (for instance, lobby functionality and object positioning synchronisation). There is custom logic made for scaffolding object part transfer between the users. This event is synchronised in the PlayerNetworkManager.cs script. The rest of functionality has been implemented using regular scripts and by using PunRPC callbacks to execute methods or remote users.

VR SafeScaff

More information: info@va.lv

NACIONĀLAIS
ATTĪSTĪBAS
PLĀNS 2020

EIROPAS SAVIENĪBA
Kohēzijas fonds

IEGULDĪJUMS TAVĀ NĀKOTNĒ

VIDZEME UNIVERSITY
OF APPLIED SCIENCES

2022