

Vectorious- Documentation(01613480/01633063)

Brief description of implementation:

Most of our Models are created in Blender.

To use those models in Game we created a Data Structure called BasicMesh (in Mesh.cpp) which stores all Meshes(as Triangles) and potential Textures which are imported using the library assimp.

The Pipeline of Loading our Game is as follows:













- The Ini Files with Position and Information which obj. Files are supposed to be imported are called first.
- Then, using the assimp library those obj. Files are imported and the Information stored as BasicMesh() Entities which.
- The Textures are also extracted and stored as Material2 Entities within the BasicMesh Entity.
- All our Meshes are part of a GameEntitiy() which are part of GameEntities(). In GameEntities() we use the Information of the imported Meshes to create the corresponding Hitboxes via PhysX.
- Each BasicMesh() has stored a Pointer to our Shader. which is used in BasicMesh::Render() to set the necessary Uniforms.

Additional libraries:





- PhysX
- Assimp

Gameplay:

Mandatory:

- 3D Geometry: 
- Playable: 
- Advanced Gameplay: 
- Min 60 FPS and Framerate Independence: 60 FPS 
 - Did not manage to implement Framerate Independence
- Win/Lose Condition: 
- Intuitive controls: 
- Intuitive Camera: 
- Illumination model: 
- Textures: 
- Moving Objects: 
- Documentation: 
- Adjustable Parameters: 

Optional:

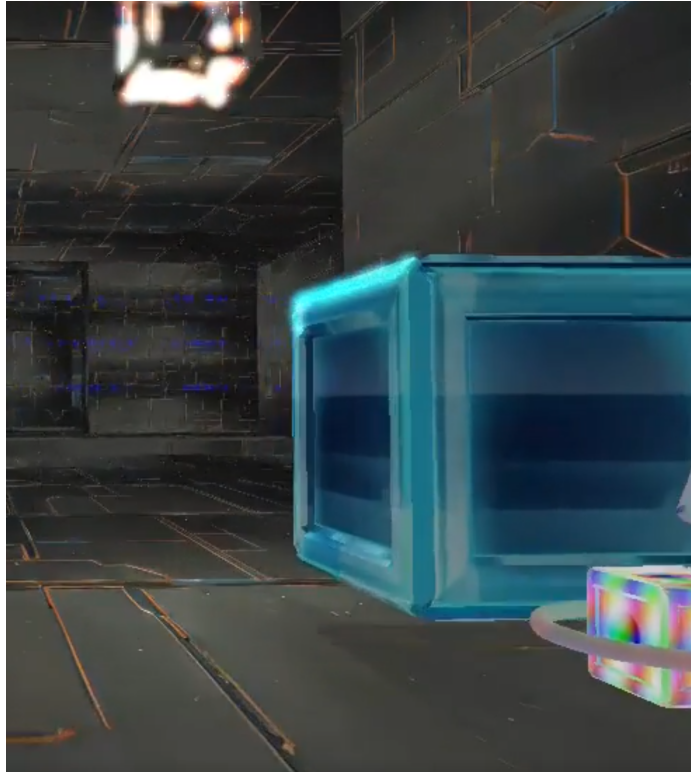
- Collision Detection (Basic Physics): 
- Advanced Physics - Sphere has to be moved via block 
- Scripting Language Integration: 
- View-Frustum Culling: 

- Heads-up Display: ✓

Effects:

Lighting: Shadow Map with PCF ✓

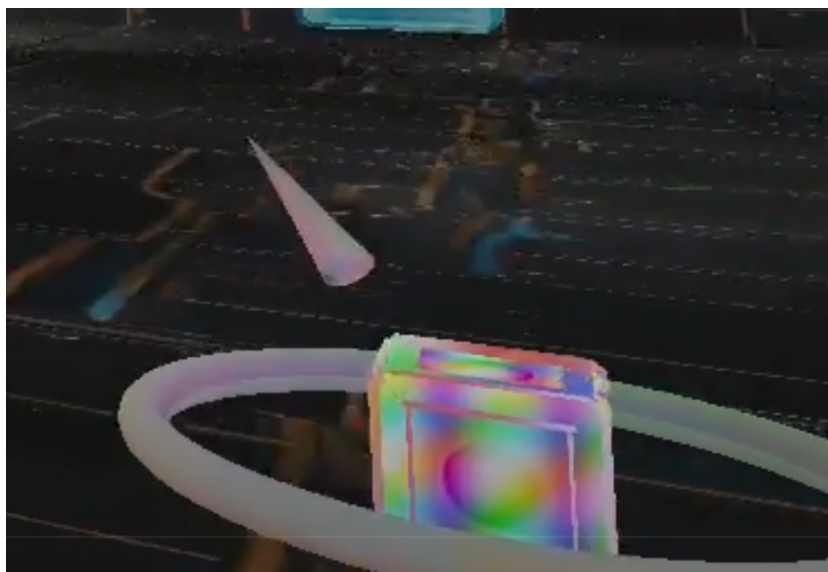
- The movable objects throw shadows.



Advanced Modelling: CPU Particle System (Lasers) ✓

Animation: Hierarchical Animation ✓

- Our Artifacts in the lower right corner move depending on the movement of the cube.

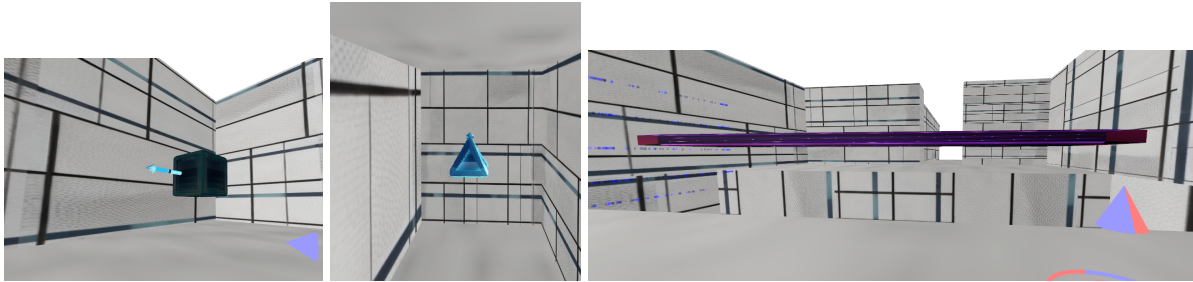


Texturing: Procedural Texture - Again, our artifacts ✓

Post Processing: Bloom (Light Sources) ✓

Other special features:

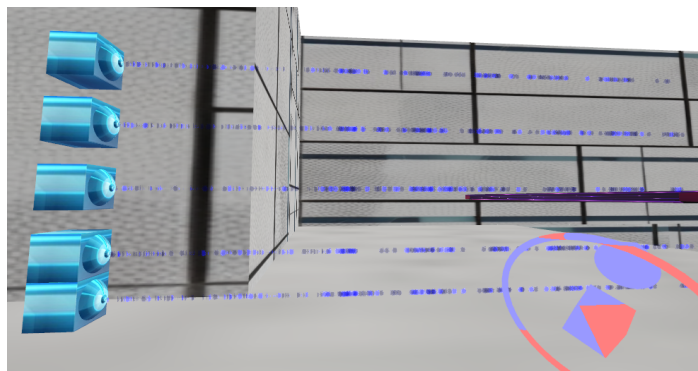
- Level Design generated from mesh_0 file in assets/level_setup
 - Can be manipulated for easier object movement and later loading of multiple levels.
 - Light Sources (lights_0.ini) and Particle Sources (particles_0.ini) can also be adapted
- Gameplay:
 - Movement via ASDW and Jumping with the Space Bar
 - The player can die by falling from heights.
 - F10 can be used to display the FPS. Fonts can be displayed. (Freetype)
 - Certain Objects (see Below) can be focused and:
 - Translated (Hotkey 1)
 - Rotated (Hotkey 2)
 - These movements can be confirmed via left click and stopped via right click.
 - Positive and Negative Movement can be confirmed by:
 - Focus Mouse on Object
 - The object (cube, pyramid or board) gets a darker shade
 - Press and hold left click - Drag into a direction and release the mouse click
 - The object should now be moving
 - The movement can be stopped with a right click.
 - You can not stand while it is moving or make it move
 - (The board does move, which will be disabled later on)
 - By pressing the respective Hotkeys the Translation and Rotation direction can be chosen (e.g. Tap '1' twice for movement along the Y coordinate)
 - The current mode can be seen in the upper left corner.



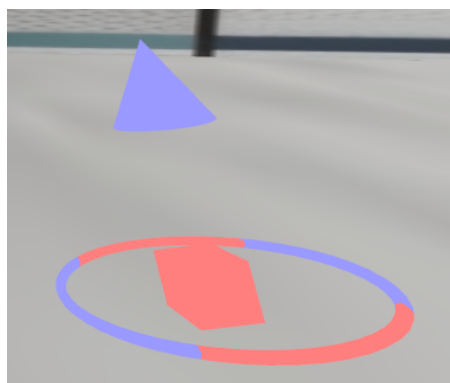
- The current objects that can be found in our game.

- For easier movement around the map a basic fly cheat has been inserted - Useable via 'F3'
 - Press Space Bar + Look into the direction that you want to 'fly' (it's pretty poorly implemented, but does it's job)

- Projectiles are generated (-> Lasers) that will 'kill' the player
 - Due to time constraints - Wrong orientation but they do their job.
 - <https://learnopengl.com/In-Practice/2D-Game/Particles>



- The artifact is currently having a procedural texture generation
 - http://learnwebgl.brown37.net/10_surface_properties/texture_mapping_procedural.html



Walkthrough:

- 1) Move the blue cube (via left clicks and axis setting (By pressing 1 - Visual cues in which the cube is going to move are given via arrows that appear once the cube is selected)) towards the lasers and block the lowest one with the cube.
- 2) Rotate the board on the left side of the big room and move it over the cliff
- 3) Enter the next room with both, the cube and the board.
- 4) Direct the cube towards the large sphere and let it fall over.
- 5) Use the board (or the cube) to shorten the distance between yourself and the pyramid.
- 6) Go with both the pyramid and the cube towards the end-goal.
 - a) Use the cube to jump over the edge.
- 7) Place the pyramid on top of the blue rectangle goal.
 - a) It might take some fine tuning to see the 'You won' appear.

References:

<https://assimp-docs.readthedocs.io/en/v5.1.0/>

<https://gameworksdocs.nvidia.com/PhysX/4.1/documentation/physxguide/Index.html>

<https://freetype.org/>