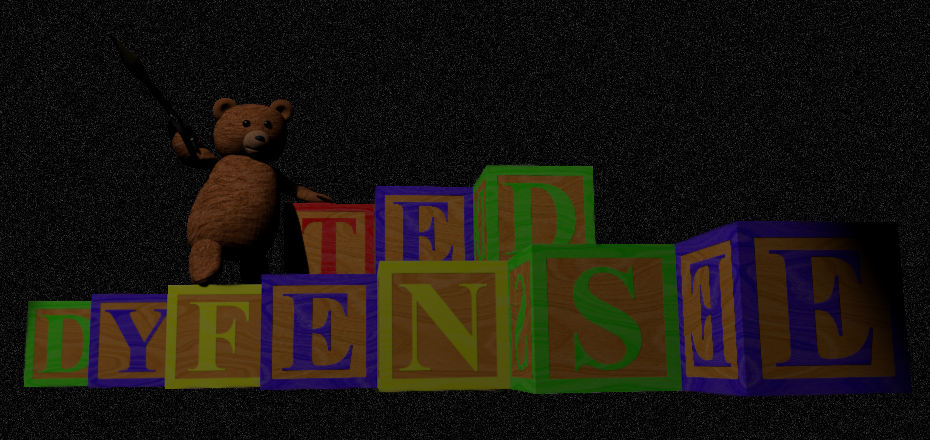
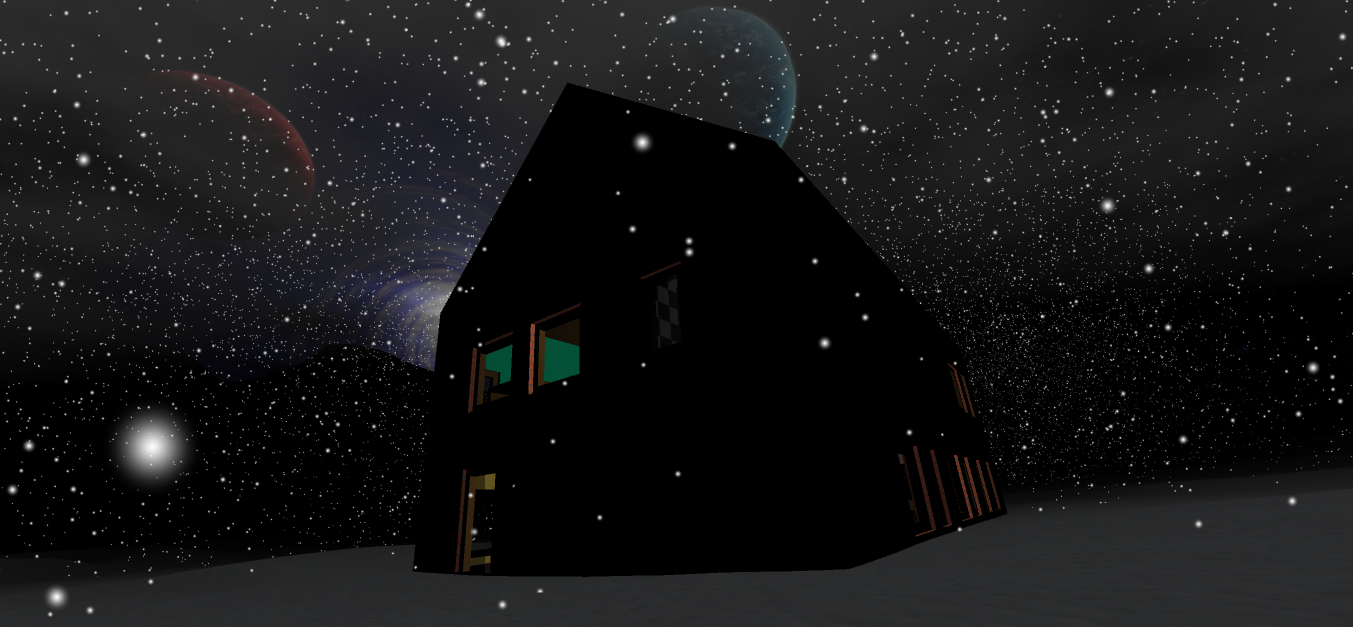
****

TedDyfense





Development team:

Abdullah Tezcan 1353306 Bouwkunde

a.tezcan@student.tudelft.nl

Jeroen Methorst Nanobiologie

Arjan van Ramshorst Werktuigbouwkunde A.B.vanRamshorst@student.tudelft.nl

Esmeralda Tomasöa 4004329 Technische Natuurkunde

e.c.tomasoa@student.tudelft.nl

Damien Crielaard Technische Natuurkunde

Pieter Kools Werktuigbouwkunde

**Introduction**

This document will introduce the reader with TedDyfense, a game which we have been developing the past 10 to 12 weeks in order to complete our minor project. The document will focus on the technically challenging components, the code quality and its management, the art and design process, and finally the development process we used to achieve our final result.

**Target Audience**

**Platform and Controls**

The game has mainly been tested on windows platforms, and any debugging was focused on achieving a working Windows build, though we have also tried a few Mac builds which worked without problems.

The controls are simple and resemble the average shooter, with a few additions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Key** | **Action** | **Key** |
| Movement | W,A,S,D | Place block | F + Left Mouse Button |
| Jumping | Space | Remove block | F + Right Mouse Button |
| Running | Left Shift | Pause game | P |
| Fire weapon | Left Mouse Button | Reload weapon | R |
| Aim through sights/scope | Right Mouse Button | Switch between 1st/3rd person | F5 |
| Change weapon/blocks | Middle Mouse Button (Scroll) | Enable Torch | T |

**Story and setting**

The story is about a young child, "Little Johnny", having a nightmare and the only thing standing between him and his nightmares is the player, "Little Johhny's Teddybear".

The setting of the game is a nightmare version of Little Johnny's house, placed into a surreal environment. The house is filled with darkness and shadow and enemies like clowns and bats are coming from every corner of the house.

As Little Johnny's Teddybear, you are tasked with defending Little Johnny from his own nightmare. To accomplish this, various heavy weaponry (Colts, AKs, Rocket Launchers, Shotguns) can be found throughout the house for you to use, but don't stray too far from Little Johnny.

**Technical Components**

Toon Shader- Arjan

Because of the cartoon like backstory to our game it seemed logical to implement an unrealistic looking shader to emphasize the surrealism of the environment. This was done by adding a threshold value to the shader. If the intensity of the diffuse lighting exceeded this value the shader will interpret the surface as lit, otherwise it will be interpreted as shadow.

Log in Screen – Arjan, Damien

The first ingame screen is the log in screen. The player has the choice to submit a new username/password combination and/or log in. When the Server is off-line or the player wishes to play without registering the Skip button can be clicked to continue as an Anonymous player.

HighScore Logging – Arjan

A Highscore system was implemented into the game. Logging statistics such as accuracy, amount of shots fired, amount of headshots and so on provides the player with a simple overview on how his game session turned out.

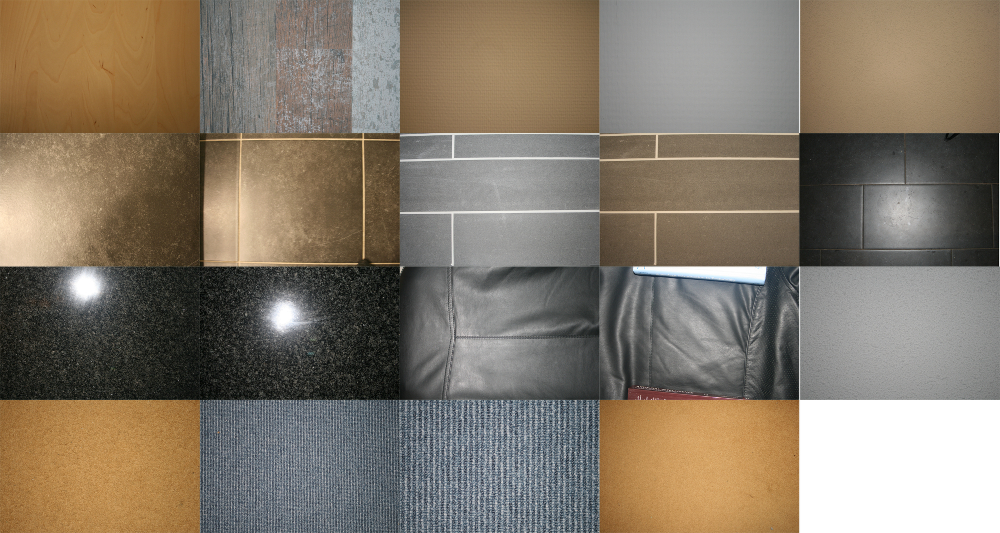
**Code Quality**

**Art, Design, and Style**

Custom Sounds – Arjan

Some in game sounds are custom made. Explosions, gun firing, blockplacement, menu transitions and menu clicks are actions that implement these custom sounds. By recording voice sounds and using common household objects the raw data was recorded. Afterwards the raw data was processed using Audacity. Using Pitch-Correction, Reverb and custom equalizers the final result was achieved.

Textures - Jeroen

Most of the textures in the game have been created from photographs made by myself. These photos were digitally edited to reduce differences in brightness and allowing them to be tiled seamlessly.  
   
*(Small example of the different photos used in texture creation)*

A small amount of base textures were taken from CGTextures. These textures were used in the creation of larger textures. (for example the planets in the skybox, these were made with small parts of 5 different textures). The sources of these 3rd party textures have all been documented and can be found in the GitHub directory (\Graphics\Textures\\_Base\_Textures\3rd Party\Sources.txt).

These textures have been used everywhere in the game. Walls, floors, furniture, weapons, the player character, and also the enemies.

Skybox - Jeroen

The game's skybox was created from a large texture built up from several smaller textures. The planets were created from base textures which have been projected on top of a sphere, while the galaxy was created using a rotation script in GIMP. The stars were generated from noise.

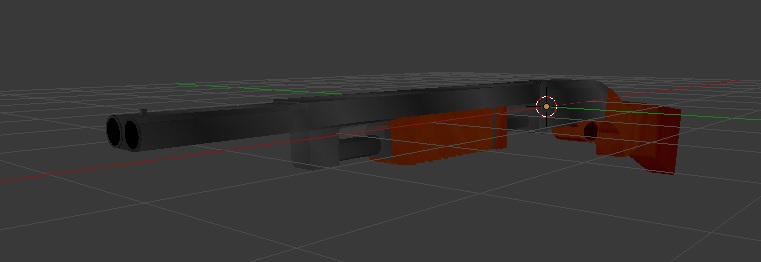
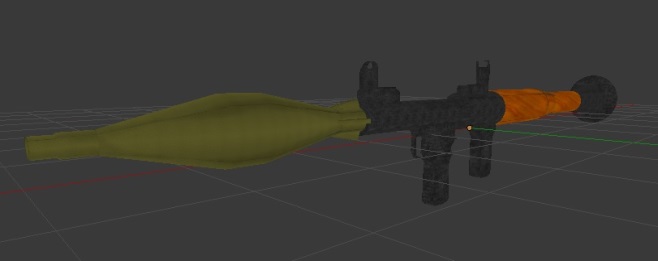


Then, to create the 6 skybox textures to be used in Unity, I imported the skybox texture into blender and projected in on top of a large dome covering most of a small cube. Then I used environment mapping to map the dome onto the cube, resulting in a file which could be used as a skybox in Unity.



Weapon Models and Textures - Jeroen

The weapon models were created using Blender by tracing over photographs of the actual weapons from different angles, this process took some time but resulted in pretty accurate models of the weapons, accurate enough for our game.



The textures of the weapon models were created by UV-mapping the individual parts of the weapons in Blender, exporting the UV-mapping to a .png file, and then adding textures to the exported UV-map.

*(Example of the Shotgun Texture)*

House – Abdullah

The house and the furniture are created with the program SketchUp. It is a standard gable design. The house is made in a grid of 1 by 1 units to work with the blockify function made by Pieter. The enemy’s can jump through the window and find their path to the flag (little johnny) in the house.

The house has two levels. On the entry level you can find the kitchen, living room and the toilet. The stairs connects the entry level with the first floor where you can find two bedrooms and the bathroom. The master bedroom has stairs connected to the entry level.

Jeroen placed the textures of the furniture, walls and the floors. Damien placed the model in to the Unity file and added the Mesh Colliders.

User Interface – Abdullah

The UI elements are created with Adobe Illustrator and exported as \*.png file. In Unity they are converted to a Sprite, which they can be added in to the scene.

 Button

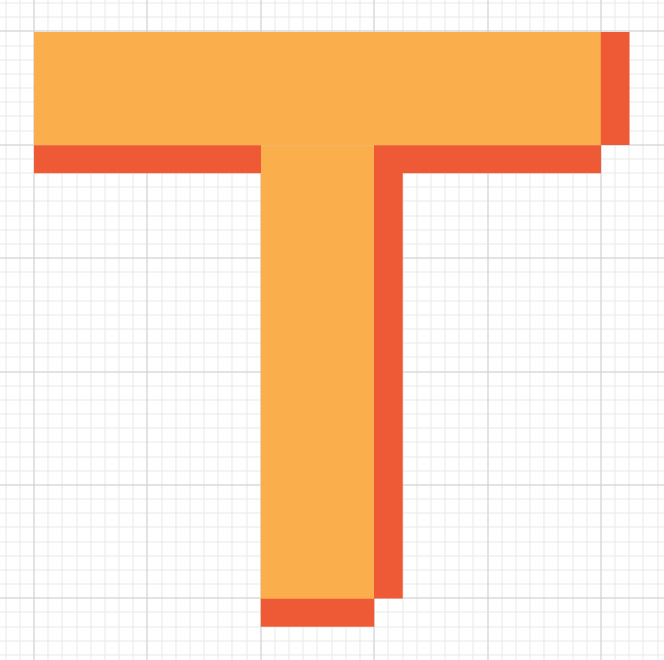
 Hover

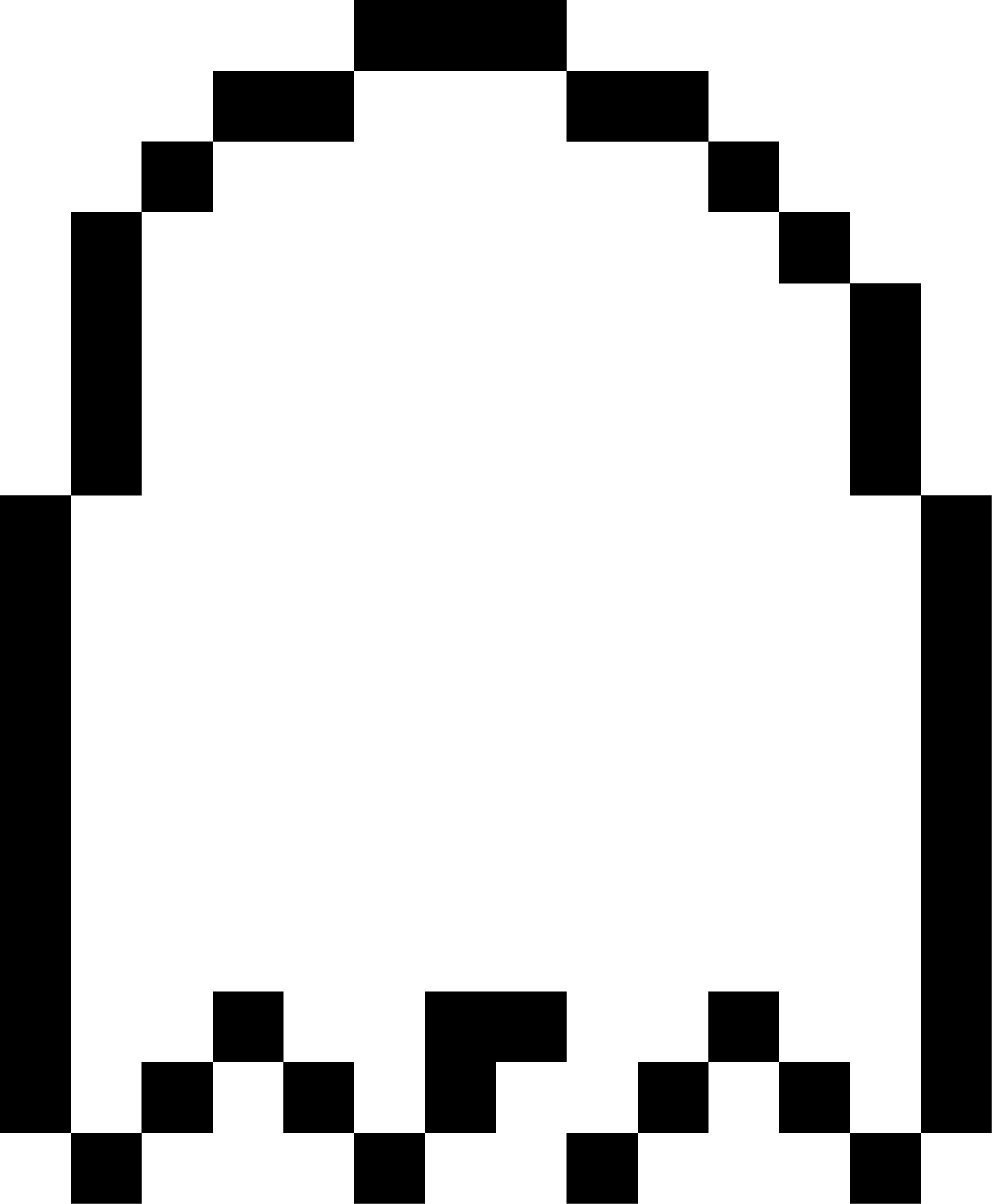
 Click

 Disabled

The buttons change when they are in interaction with the mouse. They have 4 stages as shown above.





The characters are created in a 5 by 5 grid as shown left. With the shadows the characters gets an 8-bit look.

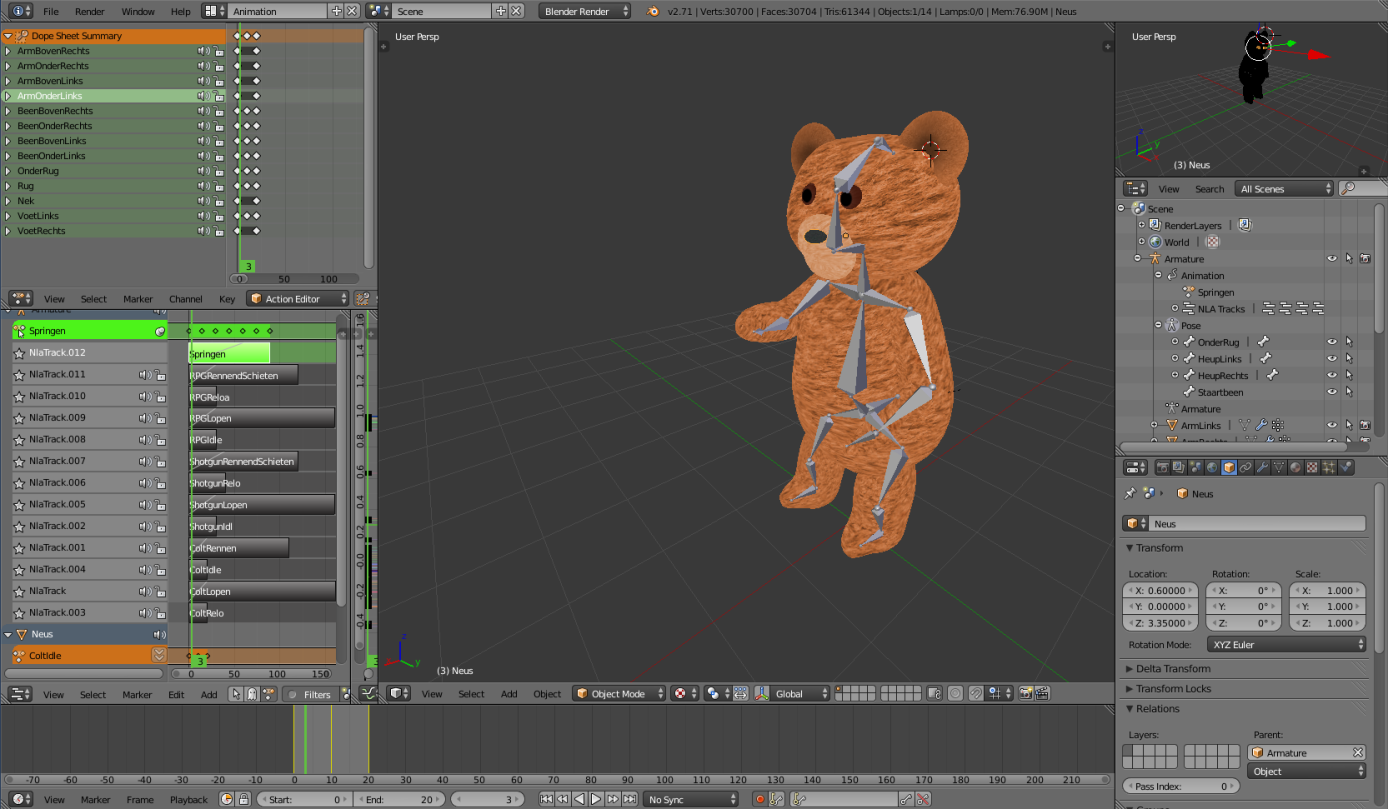
The enemy counter at top right of the screen is also made in 8-bit style.

Teddy, main character - Esmeralda

Made in Blender by sculpting multiple spheres. The model is animated, using bones in blender.

When using this feature of blender it was found out that it is very important to think about the distance of the different objects and the bones, when this distance is to big the objects won’t participate in the animation (the eyes and nose in this case).

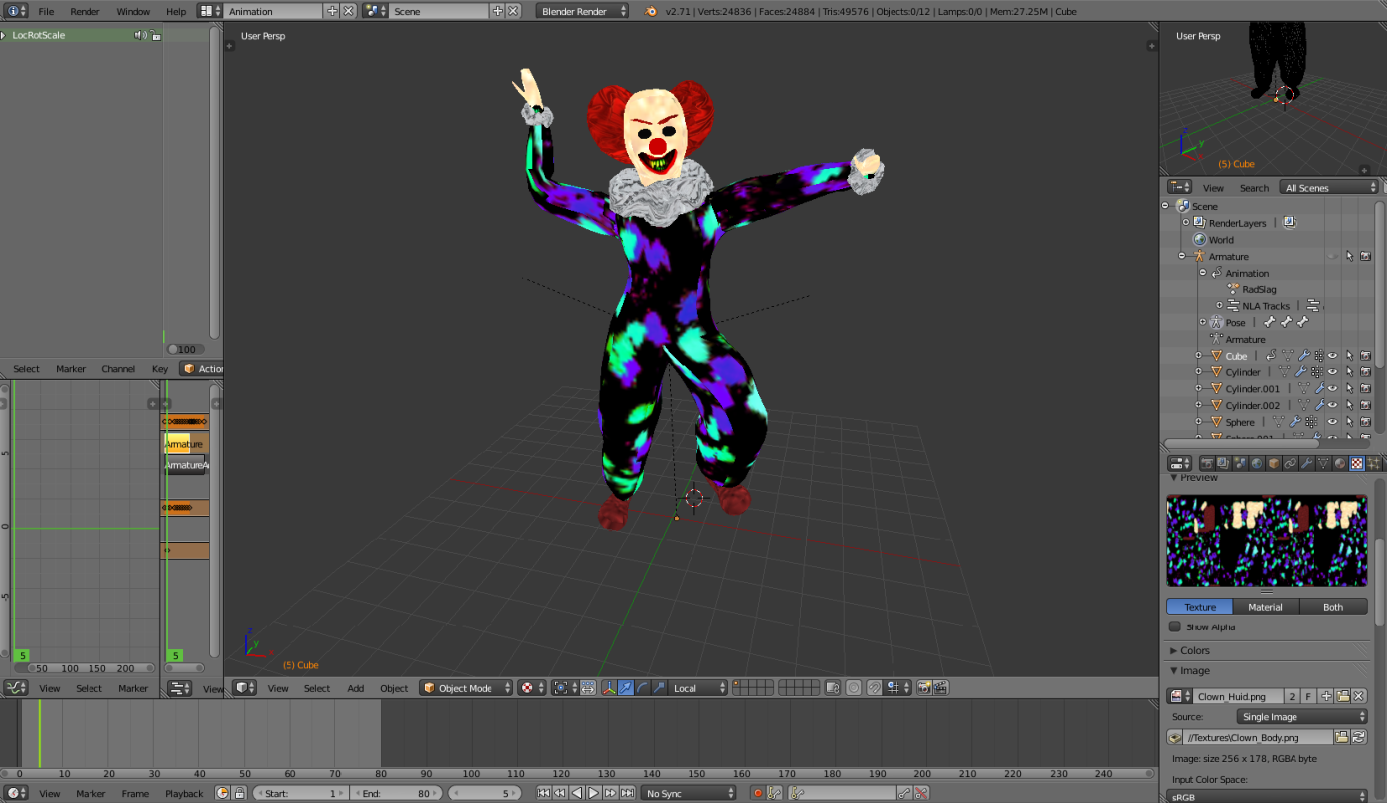
The produced animations are: Walking, Running, Dying, Jumping, Running/Shooting while holding RPG, Reload RPG, Walking while holding RPG, Running/Shooting while holding Shotgun, Reload Shotgun, Walking while holding Shotgun, Idle holding Shotgun, Running while holding Colt weapon, Idle holding Colt, Walking while holding Colt, Colt reload.



The textures are by Jeroen

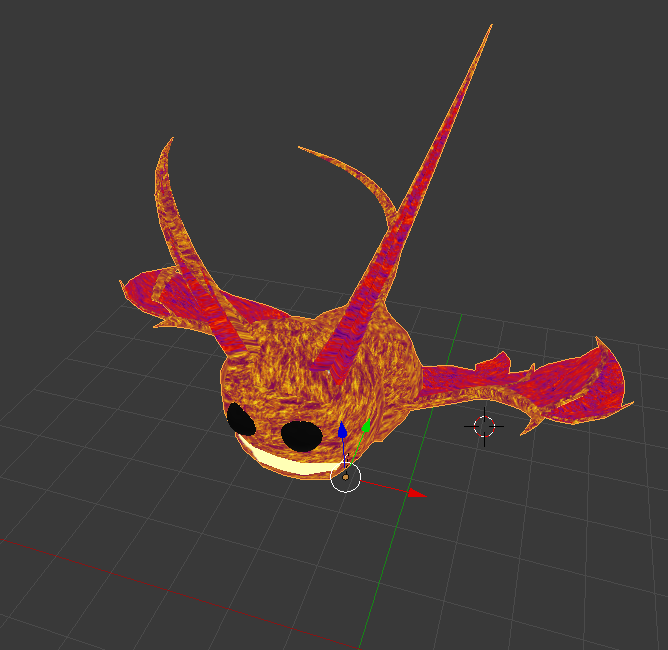
Clown Enemy – Esmeralda

Made in Blender by editing the vertices of a start object (cube). Using this method a simple character was created. In Sculpt mode the function inflate was used to give the illusion of baggy clothes. Starting from cylinders the collar of the Clown was made. The clown is texturized using base textures from cgtextures.com and google.com which were edited using GIMP. A part of the Clown his attire was given color by UV-mapping the texture on the model. This model is also animated, using bones in blender. The produced animations are: Radslag and Karatetrap.



Bat Enemy – Esmeralda

Made in Blender by editing the vertices of a start object (cube). This model is textured by editing base textures from cgtextures.com and using UV-mapping to put these edited textures on the model. The model is animated using the bones feature in blender. A single animation is made for this model: Fly.



**Development Process**

**Conclusion**