Mächler Markus

Bachstrasse 12

8580 Amriswil

m.maechler91@hotmail.com

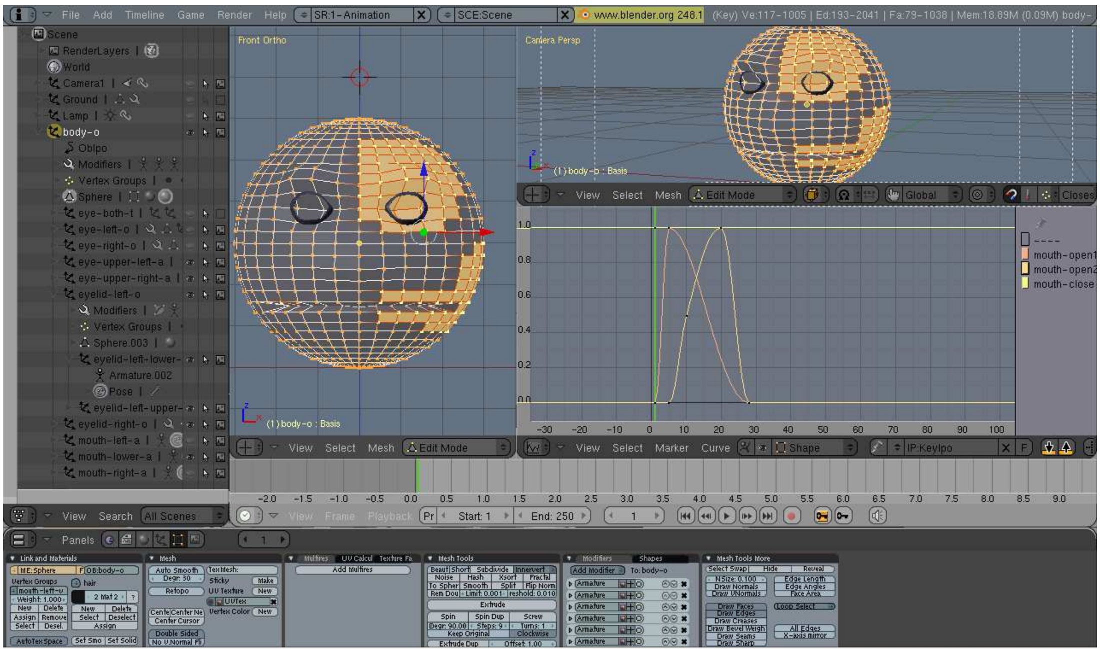
Romanshorn Grammar School

4Mz

Matura Paper

ANIMATING WITH BLENDER

HOW TO MAKE AN ANIMATION



Subject: Mathematics and Physics

Mentor: Dr. Clemens Wagner

Submitted: 10 August 2009

CONTENT

The front page picture is a screen print from my work

1 WHY AN ANIMATION?

Animations exist since the first cave-paintings thousands of years ago. Since then there was a huge development and change in the form of animations. The principle of how an animation works though stayed the same throughout all the changes. It is simply an array of images. Today's form is due to the usage of computers as advanced as never before. They apply in commercials, video games and nearly every movie. It is amazing how perfect reality can be displayed. I have always been fascinated by 3D animations. I had already some experience with the CAD software vectorworks. However there is no way to make an animation with vectorworks. I wanted to know what is needed for an animation and how they are produced. The best way to find that out would be to experience it myself. I had the idea of creating my own short animation movie. Big animation studios, such as Pixar, spend millions of dollars on their movies. I did not have the means to do that and neither would I have appreciated it; I wanted to make an animation fully free of costs.

My Matura work should consist of three parts. First I needed to find some suitable software and learn how to use it. Next I would have to find out how to organize an animation and create one myself. To end my project I would write down the whole process I went through, knowledge I gained and problems I faced.

2 ABOUT BLENDER

2.1 WHY BLENDER?

The question what software to use was as important as it was simple. There are powerful, professional programs like Maya or Cinema 4D, but they cost thousands of dollars. One alternative was the open source project Blender. Movies, like "Big Buck Bunny"[1] and "Elephant's dream"[2] made by professionals, convinced me I had found software as powerful as the commercials. The program runs on all current operating systems and the download data

[1] http://www.vimeo.com/1084537
[2] http://www.vimeo.com/1132937

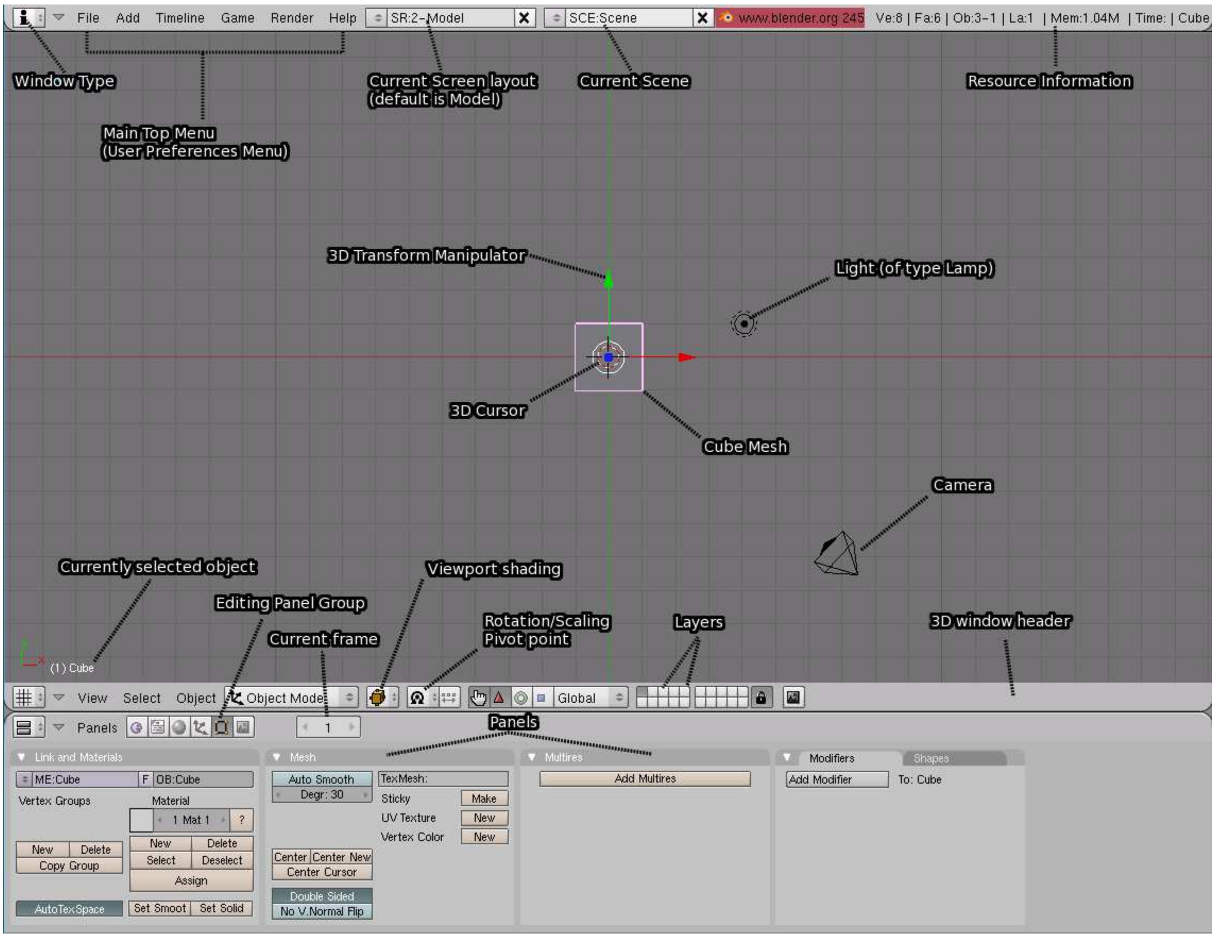
file has astonishingly 11 MB. In addition Blender is surrounded by a huge community, which developed during the eventful process of Blender becoming open source. There are two major parts; the developers and the users. Dozens of forums and tutorials exist, which make it a little bit easier to understand the complex structure of Blender and to deal with problems.

2.2 HISTORY OF BLENDER

Everything started with the foundation of the Dutch animation studio NeoGeo in 1988. It quickly became the largest studio in the Netherlands. After the creation of award winning productions and big projects for large corporate clients such as Philips the software needed to be rewritten in 1995. This was the kick-off for the future Blender. In 1998 a new company called Not a Number (NaN) was founded as a spin-off of NeoGeo. The aim of the company was to create 3D modeling software which was free of cost and provide commercial products and services around the software. NaN secured its financing from venture capitalists and employed 50 programmers. Unfortunately NaN was not able to survive in that difficult economic climate. The result was a restart of NaN with new investors as a smaller company in 2001. Due to disappointing sales and the difficult economic climate investors decided to shut NaN down. Many of the involved programmers though did not want to let Blender disappear into oblivion. In March 2002 the non-profit organization “Blender Foundation” was founded. Thanks to money contributed from enthused people all over the world Blender Foundation was able to buy the Blender source code. From there on Blender continued as open source software. Day after day a global community works on improving Blender and makes it the most powerful open source animation software. Thanks to which, non-affluent and private people can make their own images and animations.

2.3 STRUCTURE OF BLENDER

Blender’s structure is quite different to common software standards, hence to everything I knew before. The user interface and navigation might seem peregrine and incomplete. This is often believed as immatureness due to the open source structure. But what might be seen as a disadvantage of Blender is in fact one of its biggest strengths. Once you understand the way Blender works, you will realize that this is the fastest and most efficient way of 3D modeling and animating. To get used to Blender I read a manual on wikibooks and did several tutorials.



INTERFACE

When you open up Blender for the first time you see a default cube, a camera, a lamp and many buttons, the screen is divided into three major parts (figure 1), each of which consists of a window and a header. At the top you find the user-preferences, if you drag it down you get a variety of options to edit, such as view and controls, edit methods and file paths. The window in the middle is the 3D-view, in the same way as before you see the header at the bottom of the window. The third part is the buttons window. The icon on the left of each header is to select what to use the current window for (figure 2). The most important of

them is the 3D View, where all the objects are displayed and altered. The Buttons Window shows all the settings for an object, such as material, modifiers, shapes, physics, vertex groups and many more. An important tool for big projects is the outliner, where every object is listed. It is the fastest way to select objects and change visibility settings. In addition it gives an overview of all the relations, such as parenting and constraints. The IPO Curve Editor is where all the animated actions are shown. Window areas can be split by right clicking (depends on the settings, → non-selecting mouse button) on the border to the next window, or the border of the screen. Exactly in the same way areas can be joined again. This feature allows you to create your own working space and to save it either as default setting or as a screen layout.

KEYBOARD AND MOUSE

To work in the most efficient way I had to find the equilibrium between using the keyboard and the mouse. It is the best to always have the left hand on the keyboard and the right hand moving the mouse.

Typical for Blender is that it is based on reams of keyboard shortcuts. In earlier times it was even the only way to work with Blender. It makes very fast working possible when you get used to it. At the beginning it was confusing and very cumbersome, because I could not remember all the short cuts. But the system is based on some simple rules. As an example there are the three most important hotkeys; s (size), r (rotation) and g (grab → translation). When you hit any of the hotkeys you can either control the altering with the mouse or by simply entering a number. “Alt + any hotkey” clears everything done so far through this key. For example it resizes the object to the initial size.

ORIENTATION & VIEW

The coordinate system in Blender is represented by the three axes;

x (red), y (green) and z (blue)

An object's position is clearly defined by these three numbers. Blender does not use official units, but "Blender-units". To translate or rotate an object there are four different orientation modes: global, local, normal and view

Depending on the selected mode the alignment of the 3D transform manipulator (figure 1) changes. Default views like front, top, side, camera, etc. are present. In addition it is possible to get every view wanted using the mouse to rotate and zoom.

OBJECTS

Everything seen in the 3D view window is an object. Blender provides a set of default object types for modeling, rigging, lighting and also filming (camera). Most of the objects may be altered in object or edit mode using control structures such as vertices. The editing means and use depend on the specific object. It will be accomplished in the individual parts of that paper. One thing every object has is its pivot point. It is the center of mass, rotation and location. That means if an object's location is zero its pivot point is at the position [0:0:0].

DRAW TYPES

There are five different ways to let an object be drawn in the 3D view window. Bounding box and wireframe are simple ones without much detail. It makes fast working possible, but limits the quality. Shaded is rather slowly, because it needs to compute the current shadows. The perfect intermediate is solid, it allows fast working and a good overview of the 3D properties.

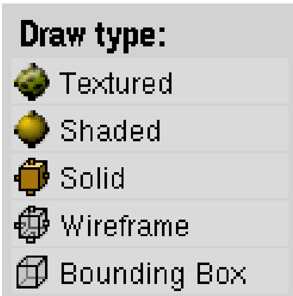


Figure 3 draw types

GROUPS AND PARENTING

Groups in Blender are not the same as in many other 3D graphics software, where grouped objects are translated together. In Blender it is simply an assemblage of objects which share attributes or may be selected all at once. It is creating an own object type. Force fields, particle systems or lightings may also restrict their effects to certain groups. To create a new group you need to select the objects and hit “Ctrl + G”. A group may be added to a different scene or copied and linked the same way as any other object.

In addition it is possible to create vertex groups. That way, effects can be restricted to certain parts of an object.

To translate objects together they need to be parented. To do so you have to select all the objects concerned and then hit “Ctrl + P”. The object last selected is the parent of all the others (children). Furthermore it is possible to parent objects to either one or three vertices. An object may be parent and child at the same time. As soon as a parent-child relationship is made the local coordinates of the parent get the global coordinates of the children. Thus children follow transformations (location, rotation, translation) of their parent, but can still be transformed independently of their parent.

MODES

Modes offer you the opportunity to change among altering options. The most important modes are object and edit mode. Changes in object mode concern the whole object (scaling, rotating, and translating), whereas changes in edit mode concern the geometry of the object with respect to the pivot point. These two modes work on every object except for lamps, cameras and empties, whose geometry cannot be edited. Further modes will be accomplished later on. Depending on the selected mode shortcuts are different.

LAYERS AND SCENES

One can imagine a layer as a transparency. It can be written on several transparencies and they can be placed on top of each other. But it is only possible to write on the topmost transparency. Therefore if several layers are activated the last selected is the active. That is the way it works in graphic programs such as Photoshop and also Blender. But in Blender layers affect much more than simply visibility and editing properties. They pilot lamp, particle-system and force-field effects. Objects can be added to a layer by hitting “M” in object mode. An object may belong to several layers.

A set of several layers is called a scene. It is possible to create new scenes, either containing objects from another scene or empty.

FRAMES

Not important for the creation of static images, but for an animation is the current frame. It can be changed in the header of the buttons window and is shown in each 3D window in the bottom left corner. A frame is one single image out of the whole animation. The frame rate is initially 25 frames per second, enough to make the human eye believe that there is a motion.

MODIFIERS

A modifier is a tool to alter an object; for example to mirror or to array. When having several modifiers exerted on one object the order may play a very important role. The topmost modifier is exerted first and every further modifier takes its input data from the output of the upper one.

3 HOW TO MAKE AN ANIMATION

3.1 INTRODUCTION

Creating an animation is a very complicated and time-consuming task. It is not only a question of modeling and animating skills around the 3D software Blender, it is much more storytelling, asset and time management, acting, editing and complex problem solving. To not lose track during that demanding process it is very important to have a clear structure and plan of what to do. I set up a timetable for myself. In the first couple of months I should get used to the software and look for good literature. I found a book called "Animating with Blender *How to Create Short Animations from Start to Finish".* It is not a beginner's book but it helped me a lot to get an overview of what is needed for an animation. The further process I went through and the knowledge I gained will be dealt with in the following passages.

3.2 ORGANIZATION

Before starting with an animation it is important to be organized properly. The final animation will not only consist of one single blend file. There may be dozens of production files, textures, sounds and at least thousands of rendered frames to keep track of. There are three different ways of how Blender deals with any of these assets:

Local assets: Local assets are contained directly within a particular blend file. That means that Blender does not have to look for them on the hard disk. Local assets are just used for small projects. In a typical short animation there are several sets and characters which would be very difficult to handle and lead to a huge file using only local assets.

Absolute assets: These assets are not contained within the particular blend file itself. Blender procures them by following an absolute disk path. Here is an example of an absolute disk path of a blend file called "Test":

C:\Document and Settings\My Documents\Blender\Object\Test.blend

Relative assets: Relative assets are in the end the same as absolute assets. However the disk reference is a little more complex. Instead of referring to an absolute path, relative paths describe the way how to get to the referenced file from the active blend file.

In figure 4 the relative path structure of a file called "face.blend" referring to a texture file called "blue.jpg" is shown. In standard code a move to the left is written as ".." and a move to the right by a forward slash and the directory name. For this example it reads:

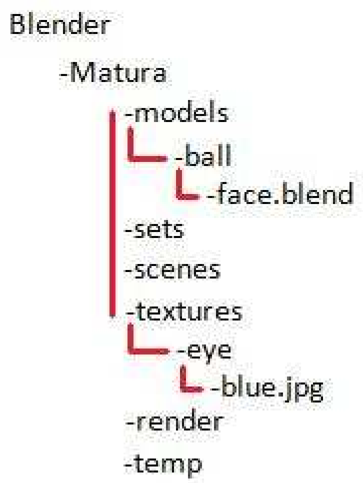


Figure 4 directory

../../textures/eye/blue.jpg

The advantage of relative assets is that they are not bond to one single computer as it is the case with the absolute ones. A project can easily be brought to another computer without having to re-link all the assets. For me this was of particular importance as I was working at three different locations.

FILE MANAGEMENT

Figure 4 leads us to the next important step in a good organized work. It is essential to have a structure in managing all files, so as not to lose time looking for them or even lose whole files. A directory structure for an animation should contain the following:

models, renders, scenes, sound, textures

Each of which may contain several subfolders.

In the Blender user-Preferences one can change the default file paths, so they fit the new structure.

ORGANIZATION WITHIN THE DOCUMENT

In an animation there may be hundreds of objects, textures, parentings, vertex groups, constraints and modifiers. To not lose track of them and to work fast it is necessary to have a structure in naming these. In my work I decided to have the following structure:

Object concerned - more defining information - type

Example 1:

Eye-left-o → it is the left eye object

Example 2:

Eye-left-upper-v → it is the upper vertex group of the left eye

Having this system it is very easy to remember names and work fast with modifiers, constraints and the outliner.

3.3 STORY

A good animation story does not only have to be engaging but all the more producible! To get some ideas I watched other animations or read some stories. I asked myself who or what would drive my story. Would it be a unique character or rather a specific set? Is it all about the story or only the animating?

I wanted to find the equilibrium in having a unique character, a nice and quite realistic set and also an engaging and complete story. Of course I thought about creating animals like in the movies “Madagascar” or “Ice Age” and let them do the funniest things, or to have an epic mass battle like in “The Lord of the Rings”. But this would never have been possible within the given resources. It had to be much simpler. I thought about a ball, singing or dancing. There could even be many of them, all in a harmonic movement. This idea with the ball caught me and I started to think of where balls occur and what other story I could make. After several approaches and transformations I finally got to my story by adding parts of each idea.

STORYBOARD

To keep my thoughts and give them a bit more of a basis I had to make a storyboard (Figure 5). That means drawing my animation shot by shot. It may also contain instructions, camera details or arrows as indication of movements. It has to show the most important steps in the animation. The entire storyboard is appended to my Matura Paper.

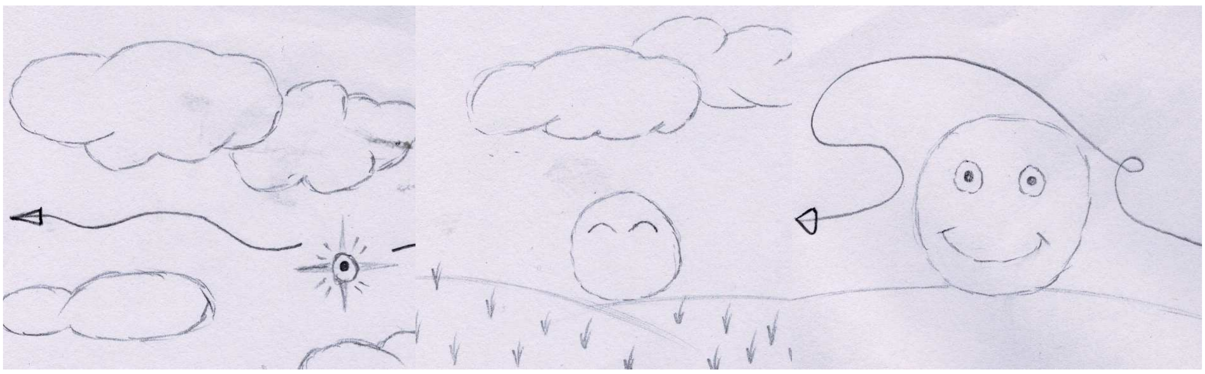


Figure 5 storyboard

3.4 LIBRARIES AND LINKING

It is not possible to create a whole animation in one single blend file, in fact it would be senseless as this file would become huge. But what is the best way to organize an animation? My first idea was to model all the characters and sets and then just copy them into different scene files to animate. This method has a huge disadvantage in that if I wanted to change something in my character or set, I would have to change it in every scene file I had copied so far. For this problem Blender provides libraries.

Libraries allow you to append and link parts of a blend file and even whole scenes to another file. Linked assets are always indicated by an "Li" icon. There are two types of linked assets described below.

Directly and indirectly

E.g. a linked scene is directly linked and all the assets contained within the scene are indirectly linked, because they are just brought along with the scene. Linked objects can not be manipulated except in the individual master file. Directly linked assets can be made local by right clicking on it, thus there are no more linked to the master file and may be edited. It is the same as if you copied or appended the object. I ended up with directory files where my characters where and linked them to the individual scene files.

ANIMATION PROXY

To animate linked objects there is another step to be taken. The easiest way to do so is to create a proxy of the linked object. "A proxy, by its definition is a stand-in for the real object. In Blender's case, creating an animation proxy makes a local duplicate of the object in question that retains its references to the original library asset."[3] To create a proxy one has to select the object and then press "Ctrl-Alt-P". Modifiers may be added, the object can be scaled, rotated and translated. It can even be changed in edit-mode, however these changes will get lost, even though they are saved, by reopening the file. Changes in edit-mode may

[3] Animating with Blender, p. 86

only be taken in the master file. Any animation done with the object is retained because it is saved in a separate asset local to the blend file.

DUPLIGROUP

Groups are usable as library assets like any other object. In the "Anim Settings" one can edit the dupligroup. The dupligroup function makes it possible to define an object as a stand-in for an entire group. Therefore an empty can be added to the scene and be animated. If the group is linked to the empty it will follow its transformations. An empty is an object whose geometry can not be edited, it is most often represented by an arrow or a cross.

3.5 MODELING

In 3D animations modeling is the process of building a model being used later on to generate the rendered image. In what follows I will describe knowledge I gained during the process of modeling and getting familiar with the software Blender.

Having finished the storyboard I had a rough idea of what my characters and sets would have to look like. To start with, Blender provides a series of default object types to choose from. Each type has its advantages in specific modeling and the decision of which object to take is quite important. Nevertheless, objects can be converted during the modeling process. This is especially important for the use of the powerful modifiers, which are the most developed for mesh objects.

MESH OBJECTS

The default mesh objects are a plane, cube (Figure 6), circle, UV-sphere[4], Ico-sphere[5], cylinder, cone, grid, monkey, torus and an empty mesh. The empty mesh is basically just a pivot point.

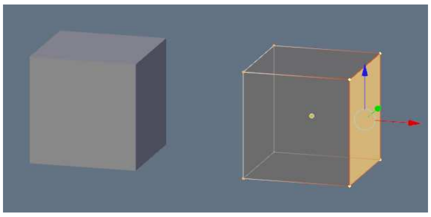


Figure 6 default cube: object- vs. edit-mode

[4] A sphere built up by several squares of different size
[5] A sphere built up by several triangles of equal size

These objects are defined by vertices, straight edges connecting them and spanned faces. Due to the freely movable vertices there are nearly unrestricted possibilities for modeling. However it is only possible to create faces with either three or four vertices. For instance a pentagon needed to be formed by a triangle and a rectangle.

An example of an object modeled with mesh is the eye (Figure 7) of my character. Its main body is a UV-sphere of which some top vertices were deleted. The pupil is simply represented by a circle. Further parts of a sphere make up the iris and the cornea. On one hand the parts are separated to give the eye a more 3D looking shape and on the other hand because of the different material settings on each of the parts.

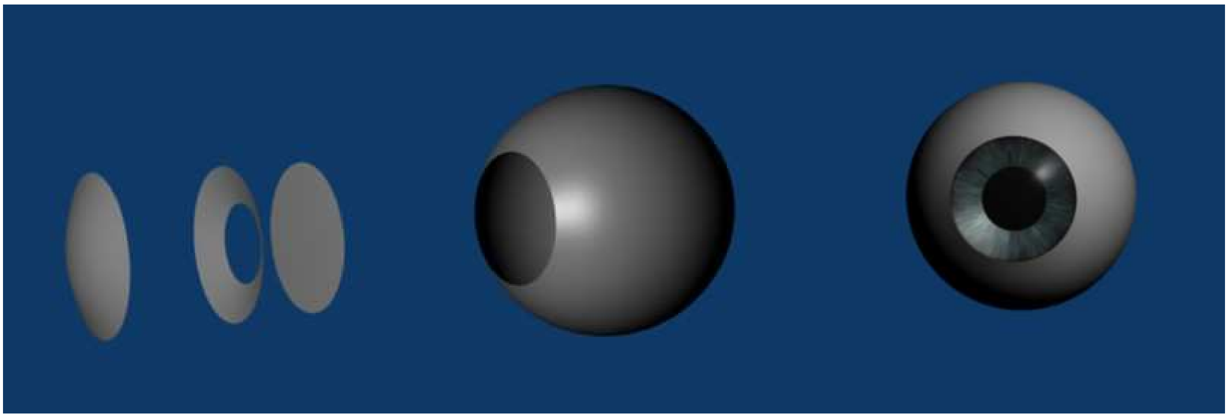


Figure 8 f.l.t.r. iris, cornea, pupil, eyeball, final rendered eye

To make the eye really look round you would have to increase the segments and rings up to at least one hundred vertices. This is not only cumbersome to work with but causes also a huge amount of calculations. In fact I only used a 12x12 UV-sphere. For that problem Blender provides a subsurf-modifier. It “subsurfes” faces and gives them a smooth look. The picture below shows how such a modifier works on different levels. It gives you the opportunity

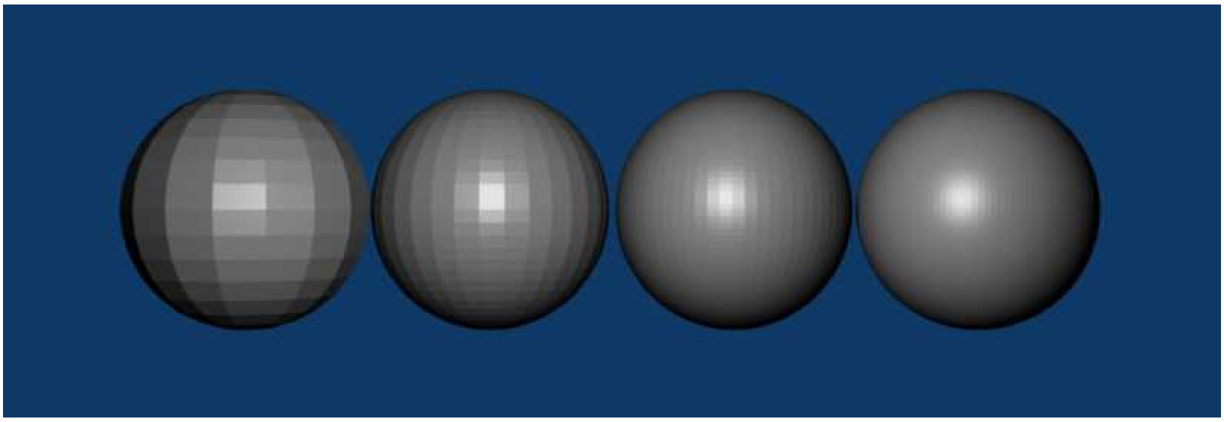


Figure 7 subsurf modifier f.l.t.r. no modifier, level 1, level 2, level 3

to work with low-vertex meshes in edit mode and have a high resolution after rendering. I applied on nearly every object at least a low level subsurf-modifier.

Because of the way a subsurf modifier works one may have to "cheat" to get a proper result. When I was modeling the body I was thinking of a mouth looking like in Figure 9 in the top left corner. In the end when I added the modifier it unfortunately led to the picture next to it. I realized that the very long edges between the lips and the inner mouth caused that problem. When the ball opened its mouth the vertices needed to be moved towards the center again. I always thought that modeling and animating would be all about perfectly coping reality. That was not the only part I had to see that I was wrong.

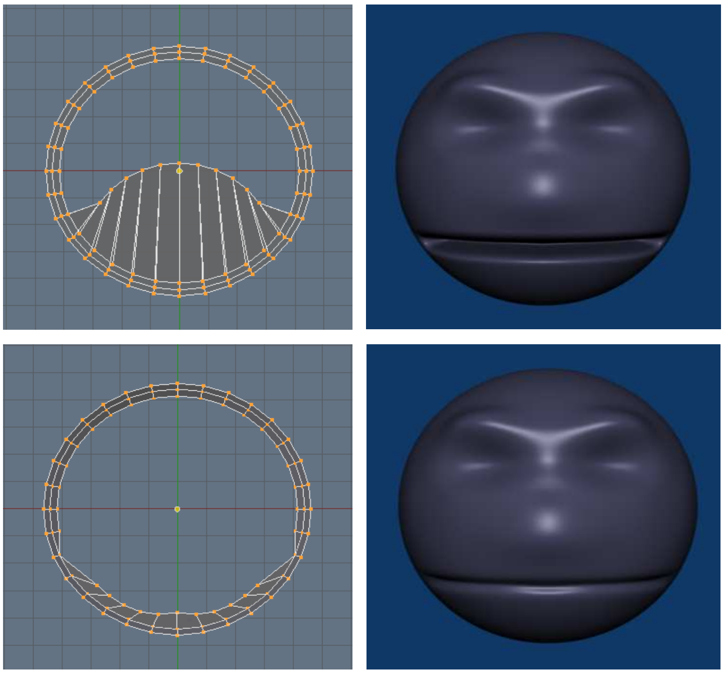


Figure 9 body with subsurf modifier

NURBS

As shown above mesh objects have the disadvantage of not being convenient to generate curved objects. It is possible to apply a subsurf modifier but it gets nearly impossible to work on a fully subsurfed object in real-time. Non-Uniform Rational B-Splines (NURBS) are the best way to model complex curved objects. In Figure 10 you see a NURBS donut with its control structure in edit mode on the right hand side. As single vertices may not be deleted or added, this makes it much more inflexible than mesh objects.

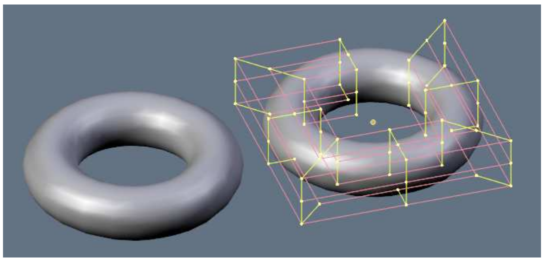
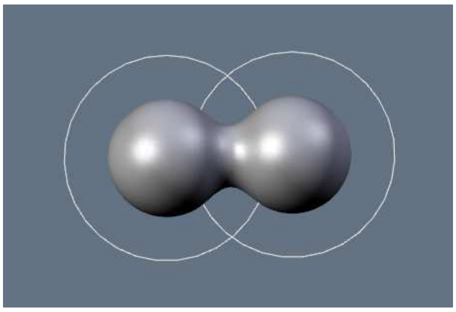


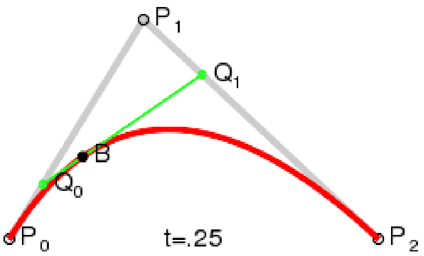
Figure 10 NURBS donut

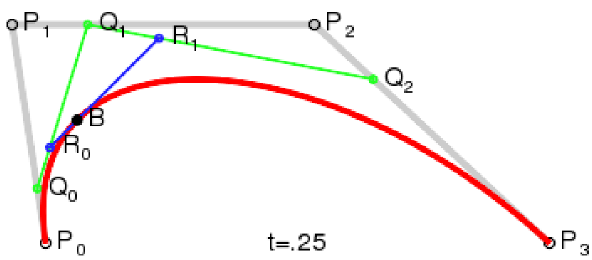
META OBJECTS



A meta object can be seen as an exerted field being visualized. Depending on the source the field changes. For example a point source causes a spherical object and a line causes a tube. An attribute of these objects is that they merge together when they are close to each other, similar to water drops. Thus a typical application for meta objects is to simulate water. The advantage of it is that it does not need so many calculations as real water simulation. It is also possible to set the field negative. When a negative object is brought close to a positive one the negative field is subtracted from the positive giving it a dent. Since negative objects are invisible no effect can be observed by two interacting negative objects.

BEZIER





Bezier curves were developed by Pierre Bézier, who used them to design automobile bodies. The way a Bezier curve is generated can easily be described by visualization. There are Bezier curves of different orders. A curve with order one is called linear, it is simply the straight connection between two points. A quadratic Bezier curve is described by three points. Figure 12 shows the way the curve is generated[6]. At the start Q_0 and B are on P_0 and Q_1 is on P_1. Then Q_0 moves to P_1 and Q_1 moves to P_2. B is following the

---

[6] Animated visualization: http://en.wikipedia.org/wiki/File:Bezier_2_big.gif

green connection line to Q_1 and they end up at P_2. All the points reach their target simultaneously. The line described by B is the Bezier curve. Figure 13 shows the way a cubic function is generated. It also works the same way for curves of higher orders.

Bezier curves in Blender:

At first sight it seems that Blender Bezier curves (Figure 14) would not have much in common with what was explained above. First of all the initial Bezier curve in Blender is cubic. Thus it must have four control points; these are labeled with a capital C and C'. The curve goes from C_1 to C_2 passing the points C_1' and C_2'. R_1 and R_2 do not have an effect on the shape of the curve. They are the prolongation of the connection between C and C' and are used to rotate C' around C without changing the distance CC'. Moving C causes also C' and R to move, whereas C' and R can be moved independently of C.

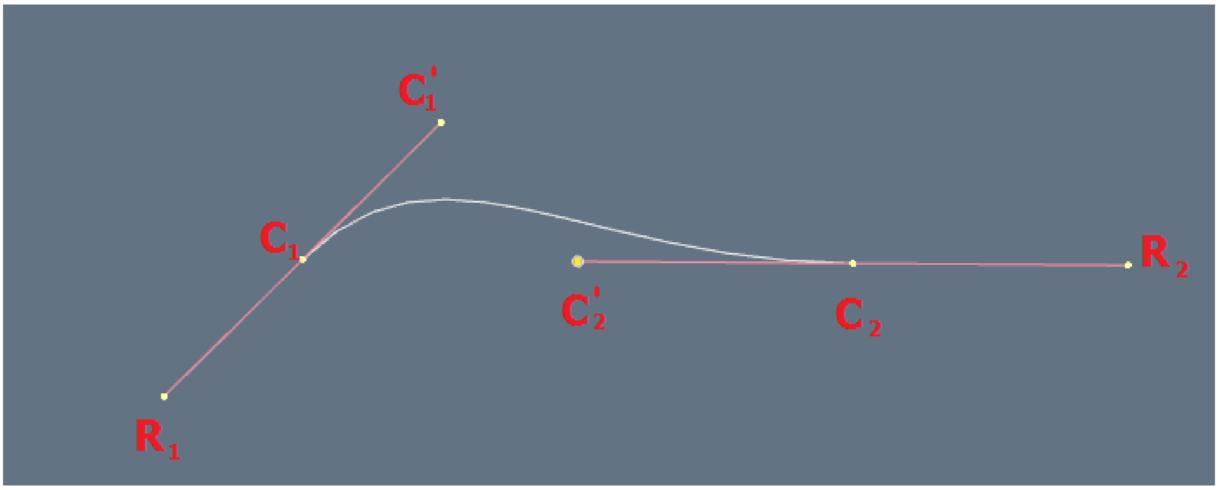


Figure 14 default Bezier curve in edit mode

A special attribute of Bezier curves is the possibility to let the bevel shape be defined by another Bezier curve. This tool makes it easy to model wires or complex curved objects like glasses. To create a wire one has to define the bevel object of any curve in the edit buttons menu as a Bezier circle. The other way round it works for creating for example glasses. The Figure below shows a cup object and its defining bevel curve.

Figure 16 bevel Bezier curve

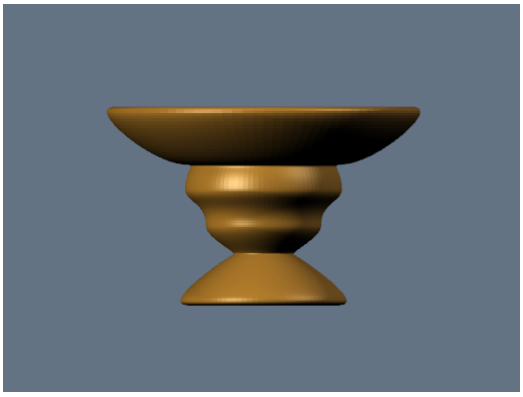


Figure 15 cup

FURTHER OBJECTS

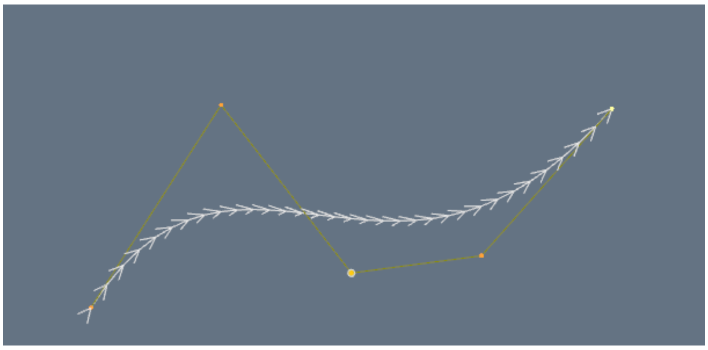


Figure 17 path in edit mode

Important tools for animations along curves are paths (Figure 17). They work as Bezier curves except for the different looking control structures and purposes. Not really important for modeling but more for animating are empties. An empty in contrast to a single pivot point is an object itself. However it is not editable or visible after rendering. I used empties to create wind and spherical fields, but also as tracking target for the eyes and stand-in for dupligroups. In addition to all these objects Blender also provides a text tool.

EDITING MEANS

Most of the editing tools only work on mesh objects, thus nearly every object gets converted to mesh for the major modeling. An object can partly be edited in object mode. That is the case for changes made to the whole body. For example scaling or rotating an object. These can also be edited in the transform properties panel (Figure 18). There certain properties can also be locked to not be editable anymore. The biggest part of the modeling is directly done with the mouse and the keyboard. There are some important hot keys giving all the means needed to model.

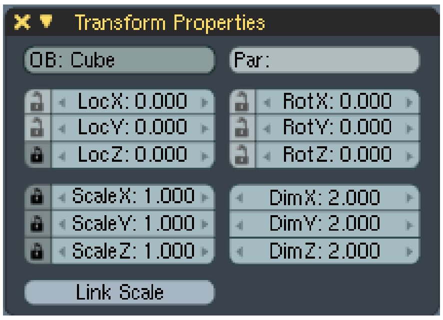


Figure 18 transform properties

S: scale the object or selected vertices (scales the edges connecting the vertices)

R: rotate the object or selected vertices

E: extrude, it is possible to only extrude vertices or edges, but also regions (faces)

G: grab the selected objects or vertices and translate them

F: face the area among the selected vertices

H: hide objects or vertices to focus better on certain objects or parts of them

To get exact numbers for translating, rotating, etc. first select the operation you would like to apply and then simply enter the number.

Sculpt mode:

Sculpting was first released in Blender 2.43. Since then it became a very popular edit method because of its power and simplicity. In contrast to edit mode, vertices, edges or faces are not altered, but the form of the object by using a brush. It is comparable to sculpture or working with argil. There are tools like draw, smooth, pinch, inflate, grab, layer and flatten. You can vary the size and strength of each of the tools. In addition it is feasible to use symmetry on the axes or to lock axes. However sculpting only works on existing vertices, they are never added or deleted. It is also possible to restrict the sculpting to certain vertex groups.

I used sculpting in particular for the creation of my main body.

Since my character is a ball I chose to have a sphere with 32 rings and 32 segments as the initial object. That was not too much to handle, but still enough to be able to work properly with the sculpt tools. I used an x-axis mirror for obvious reason.

The modeling was not finished as soon as I started to animate. I still had to change and adjust a lot of things. Because I linked the files properly this was no problem.

3.6 MATERIAL AND LIGHTING

So far objects just appear in a light grey color. They have to be given a material and be properly lighted. The material settings panel is found in the context button “shading”. There a new material can be added. An object can also have several materials, which can be linked for example to a particle system of the object. Materials may also be shared with other objects.

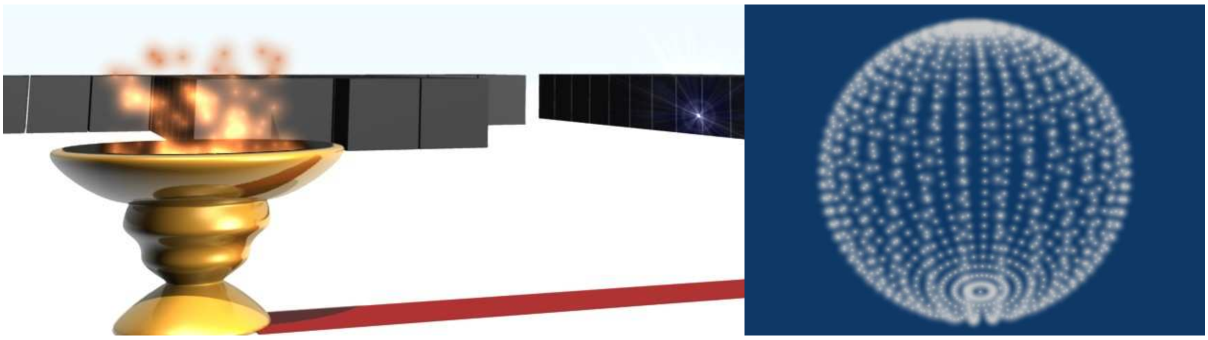


Figure 19 extract of the final animation and a UV-sphere using halo settings

The most basic change is a color change with the RGB spectrum. There also the color of the specular (sheen, normally white) and the mirror is editable. The most important setting beside the color is the alpha value, the opacity of a material. Special pre-settings are for example the possibility of letting the wireframe be rendered or giving the object a halo effect (Figure 19). It lets the vertices of an object shine. I used such settings to create fire and the shiny blue object in the Figure below. A particle emitter sends out particles using halo material with an orange color and additive behavior. The additive behavior leads to a color change due to the density of the particles. A special challenge was to let the halo effect be visible. It is nearly impossible to see it on a light background. Thus I often had to adjust camera angles and positions of the halos to have a dark background.

A more specific way to color an object is colorband. It enables color gradients. Figure 20 shows my colorband settings for realistic looking grass. Surprisingly for people new to computer graphics may be that it goes from black to yellow and that only about half of it is green. This is due to the lighting causing it to look different.

Figure 20 colorband for grass

To have even more control of the coloring, vertices and their surrounding area may be colored using vertex paint. It is like painting the object with a brush, but using the mouse.

TEXTURES

Textures are pictures put on objects to give them a high amount of detail without having a huge amount of modeling and calculations. For example instead of modeling each brick of a wall, one simply puts a texture on a flat plane.

Very powerful tools are displacement textures (Figure 21). They change an objects surface according to brightness changes on the texture. I used it to make the hair of my character look more like dreadlocks.

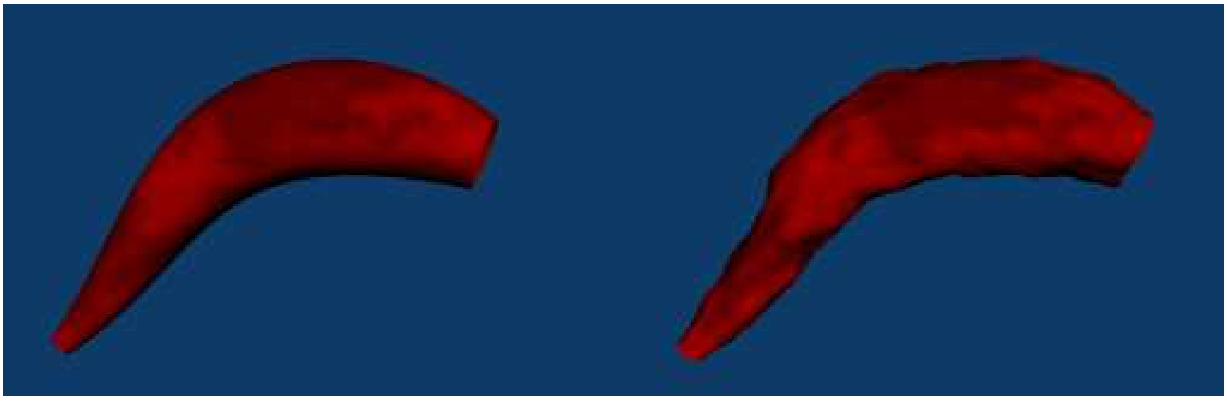


Figure 21 displacement texture

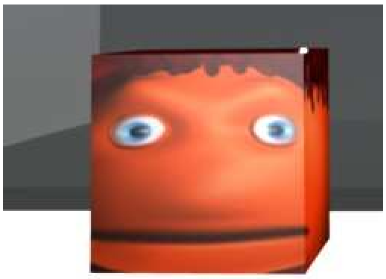


Figure 22 UV-mapping on a cube

Textures on 3D objects are kind of wrapped around it, sometimes leading to unsatisfying results. UV-mapping gives you the opportunity to control the exact placement. You select the individual faces and then load a picture on it. In the UV-window it is also possible to stretch, compress or clip the images. The picture on the right shows a version of my character in the two final scenes. The UV-mapping has later been replaced by a surrounding box with glass looking settings.

Similar to vertex paint it also possible to paint different textures on an object using the texture paint mode.

For the sky settings I used a cloud texture. The clouds could be animated by linking them to a moving empty. The problem was that the sky setting is always with respect to the camera that caused two problems. On one hand the color gradient I added to get a horizon would have needed to be changed every time the horizon moved up or down on the screen. On the other hand the clouds got messed up whenever the camera was moving. To solve that problem I needed to find the equilibrium between slow movements of the camera and parenting constraints between it and the empty.

LIGHTING

The lighting is one of the most important and difficult parts in an animation. Blender provides five different lamp objects; Lamp (all around shiny, point light source), sun (parallel light beams of equal strength), spot (cone of light), Hemi (light emission of a hemisphere) and area (plane expansion, luminous row)

The most important settings are the color, energy (brightness) and the falloff (decrease of energy with increasing distance).

I mostly used lamps, because it was the easiest to handle and led to good results. Often it is not enough to light an object with one single lamp. To keep lamps from lighting the wrong objects their effect needs to be restricted to their layer.

3.7 SIMULATION

Blender provides a set of physical simulation tools. There are different kind of fields, cloth-, water- and softbody-simulation and particle systems. Using an extract of my final animation (Figure 23) I will explain how some of them work. The flag in the picture is made using cloth settings and an animated wind field. For cloth animation it is important to have a high amount of vertices to get proper results. Wind fields have also been added to the grass to make it move natural. The grass is made with a hair particle emitter. And finally the balls are deflecting the grass with a spherical field around them. The grass uses weight paint for difference in density and length.

Figure 23 extract of the final animation

It is also possible to simulate gravity and physical body interaction in the so-called game mode. These movements may be saved to an IPO-curve and used in the animation.

Examples of physics animations are included on the CD in the folder “Sample Files”. To start game mode animations hit “P” and to start the others hit “ALT + A”.

3.8 RIGGING

Rigging is the process of giving an object control structures to animate it. There are several ways of doing that and depending on the needs and the object the most suitable has to be chosen.

ARMATURES

Armatures are similar to bones in the human body. An armature can be added like an object. Then it needs to be linked to the object it should deform through an armature modifier. The deformation is based on moving the vertices, thus it is important to have vertices at the key positions where the object needs to be bent. It is also possible to restrict the effect of a bone to a vertex group. For an armature three different modes exist. The first one is the object mode, where the armature is placed in the right position and scaled to make it the easiest to animate. Edit mode is where an armature can be extruded, which makes it possible to construct an entire skeleton. Thus armatures are always used to animate body motions. The last mode is the pose mode, this is the mode where changes on the armature (rotating, scaling, etc.) cause the object to deform (Figure 25) For easy visibility changes it is very useful to place the object and its control structure on different layers.

To make smooth motions possible one has to weight paint the deformed object (Figure 24). A weight paint value of one means that this part of the object completely follows the bone. On the other hand a value of zero lets the bone not at all have an effect on the object. The Figure below shows a very simple weight paint and deformation of a box.

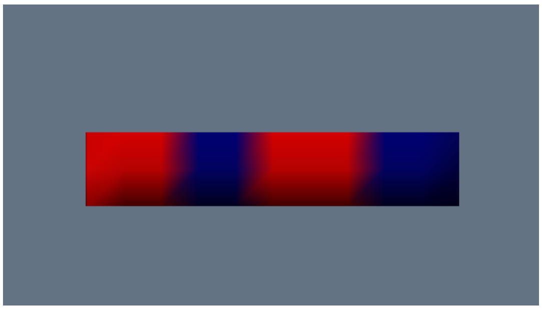


Figure 25 weight painted box

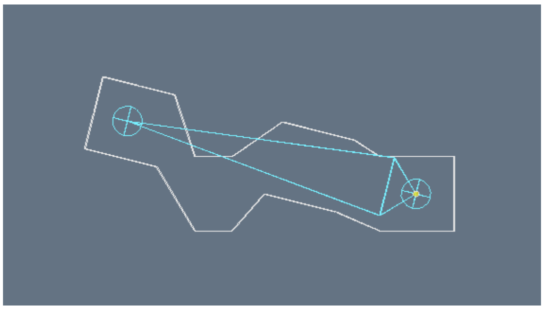


Figure 24 deformed box

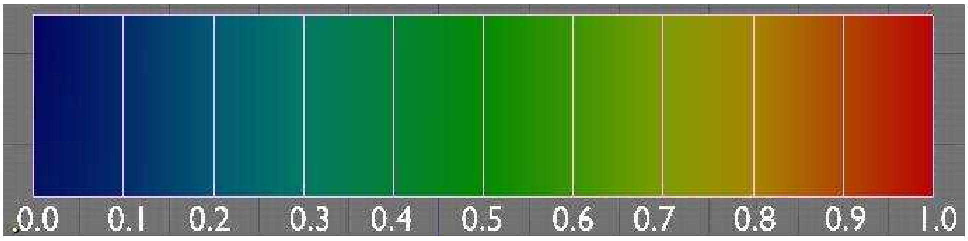


Figure 26 weight paint spectrum

Weight paint is not only used for armatures, but also for driving the influence and strength of particle systems.

Figure 27 armature deformation

My first attempt to animate was using armatures. I wanted to make the corners of the mouth move up and down related to armature movements. I was working very carefully with vertex groups and weight painting. Despite all the effort it was just infeasible to get a smooth and nice result. There must have been another way.

Because of the complexity of the body and the high number of vertices I decided to make the eyelids as an additional object parented to the body. An eyelid would look like half of a sphere, which would perfectly fit to my roundish character. I used half a sphere, deleted some vertices, copied and scaled it down and connected the edges (Figures 28-30). In the same way as with the body the results of the eyelid were not satisfying. So far it looked far too thin and did not close properly.

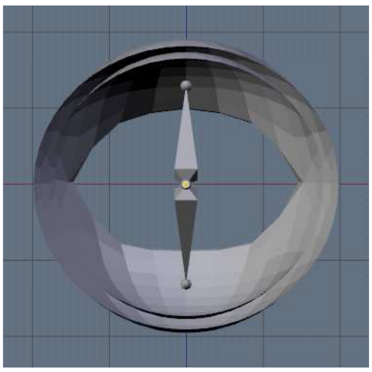


Figure 30 eyelid rear view

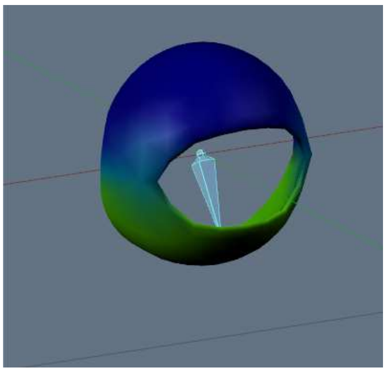


Figure 29 eyelid weight paint

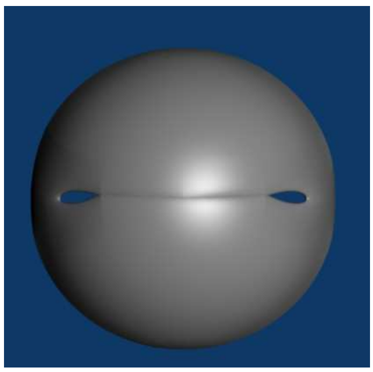


Figure 28 eyelid closed

Next I tried to make the eyelids part of the body. I deleted some vertices around the eye and then started to form an eyelid by extruding vertices and faces. The result was still not satisfying. Things changed when I started to work with another powerful tool of Blender.

DRIVEN SHAPE KEYS

As its name indicates shape keys are savings of shapes. That means that the position of vertices can be stored. By adding a shape key for the first time the current shape is saved as the base-shape of the object. When further shape keys are added, the object at first does not change, because every added shape key is initially a copy of the base. Then the object can be manipulated. Before starting to make dozens of shape keys I had to check which shapes I really needed. Thus I had a look at my storyboard and thought the whole story through. I often ended up in front of the mirror watching my face change when I smiled or looked sad. A good smile needs more than simply moving the corner of the mouth, nearly the whole face is concerned; in particular parts around the eye. In the end I had a whole set of shapes from different happy expressions up to fear and eyebrow motions. It is possible to apply several shape keys simultaneously, the deformation is then added up.

To animate shape keys it is necessary to add a driver. That is an object which changes in scale, rotation or position, affecting the value of the shape key. A value of zero means there is no effect and a value of one means the current key is fully applied. All the intermediate steps are interpolated by Blender. In addition it is possible to have shape key values higher than one or lower than zero. In this case Blender extrapolates the deformation which might be very helpful, but also lead to bad results.

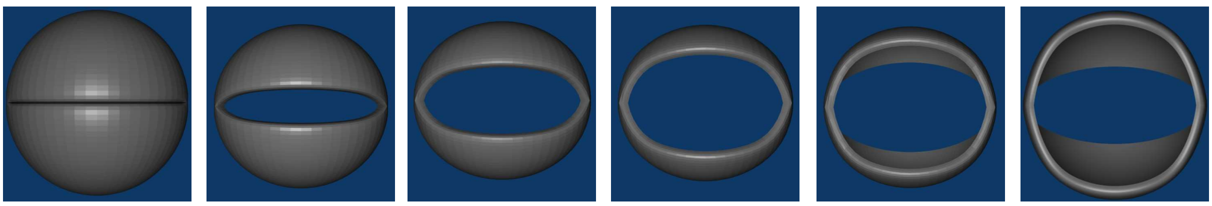


Figure 31 shape key values f.l.t.r. 0, 0.2, 0.4, 0.6, 0.8, 1

I decided to have armatures as the driver of my eyelid and also the shapes of my body. The Figure below shows the settings for a driver of the lower part of an eyelid. On the horizontal axis (x-axis) the rotation of the armature is displayed. The vertical axis (y-axis) shows the shape key value. The initial position of the armature is horizontal, pointing in the same direction as the eye. I set the extend mode to constant, thus I restricted the shape key value to values between zero and one. For the armature of the upper part of the eyelid it looks perfectly the same, except that it is mirrored at the y-axis. There I want positive rotation values to affect the opening of the eyelid.

Figure 32 driven shape key setting

For all the shape keys of my body it worked the same way. Except that I did not relate the motion to rotation, but scaling.

LATTICE

Another method to deform an object is to use a lattice (Figure 33). In principle it is a surrounding low-vertex box, which deformation causes the linked object to deform. So it gets possible to easily modify an object with even more than one thousand vertices only by changing an object with eight vertices. The deformation may also be restricted to certain vertex groups. That is the way I animated the compressing and stretching of my ball.

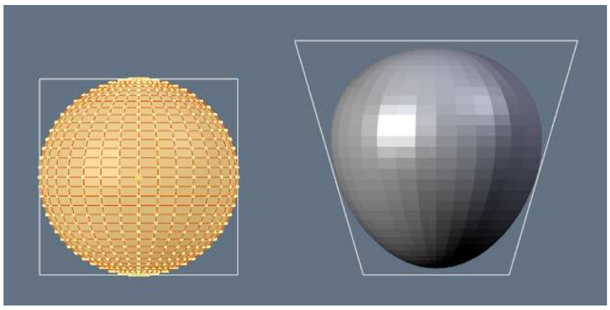


Figure 33 lattice deformation

For some reason lattice shape keys cannot be driven. As an alternative I discovered hooks. Vertices can be linked to an empty in a way that they follow the motion of it. Thus I linked the top four vertices of the lattice to a hook and then animated an up and down motion of the empty. Since the empty was a child of the lattice the global coordinate system of it changed to the local one of the lattice making it easy to have such a motion.

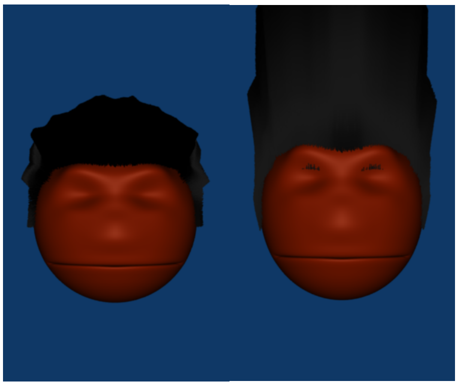


Figure 34 body normal vs. body stretched

The lattice modifier caused even more problems to solve. When I stretched or compressed the ball it had an effect on the particle system making the hair (Figure 34). Thus I could not use it anymore and I found a substitute in the Bezier curves. As seen in the material section I ended up with dreadlocks.

3.9 ANIMATING

When I had rigged everything the drivers needed to be animated. Animating is all about keyframes and IPO-curves. A keyframe has a saved value of any item. Such a value may be the location of an object on the x-axis or also the alpha value of materials. The IPO-curve describes the developing of these values with respect to time.

The figure below shows a simple movement of an object. The vertical axis describes the development of the selected attribute. In this case it is the z-location of the object. The horizontal axis is the timeline. One has to keep in mind that time is not measured in seconds, but frames. To remind you the initial setting is 25 frames per second. If it is necessary one can also let seconds be displayed. During the animating part I started to think in frames rather than seconds. The green line indicates the current frame. The following motion is built up by three key frames. When having the object selected just hit “I” to add a key. In the buttons window the current frame can be changed.

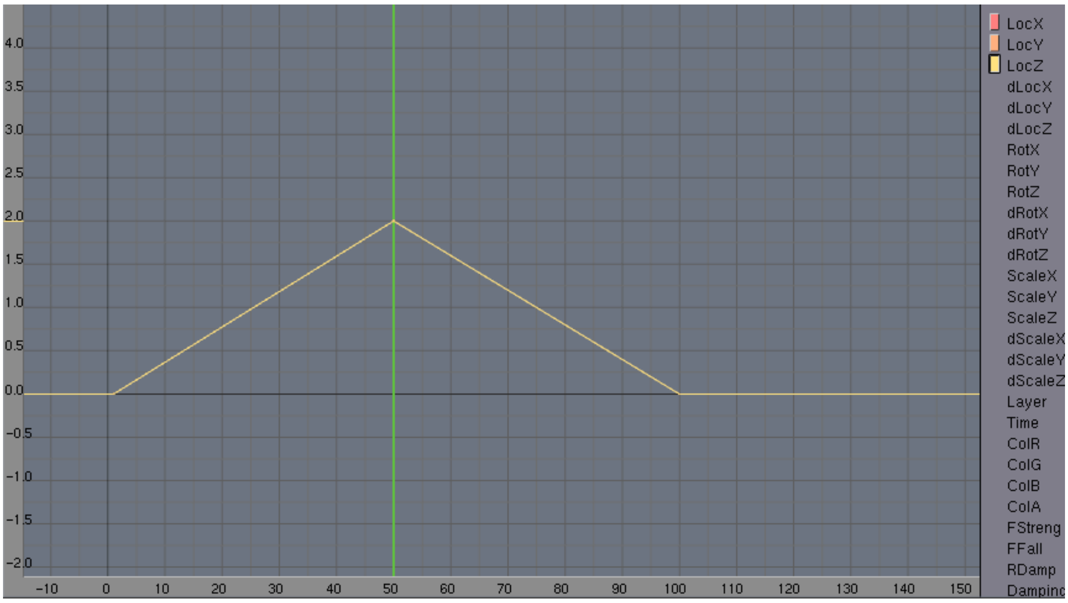


Figure 35 Ipo curve

The object starts at position LocZ = 0, t = 0. After 50 frames the object is at the position two on the z-axis. Again 50 frames later the object is at its initial position.

In the example above I used linear interpolation and constant extrapolation. That way a singular motion with constant velocity occurs.

There are two more ways of interpolation:

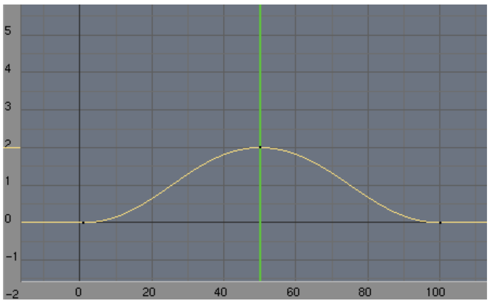


Figure 36 shows the same motion of the cube as above, but with Bezier interpolation. This gives you more control over the speed between the single keyframes. The curve is editable like a standard bezier curve I have mentioned earlier.

Figure 36 bezier interpolation

The last way is to use constant interpolation. This causes the object or item to jump within one frame to another state. This is especially of use when constraints such as parenting or tracking change within the animation. I used a tracking constraint to make the eye follow an empty. In the first scene the empty was first parented to the body, so the eyes would not look in another direction when the body rotated. But then, at the point of time where the halo object entered and the body opened its eyelids for the second time, the empty was suddenly child of the halo object.

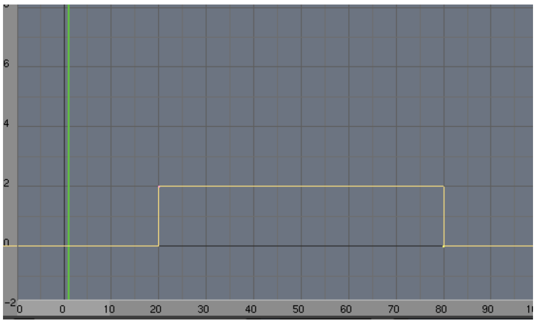


Figure 37 constant interpolation

All of these examples are using constant extend mode, but there are three more possibilities to extend:

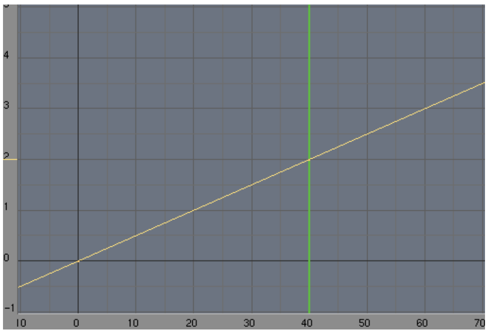


Figure 38 extrapolation

Extrapolation:

The animated part starts at the origin and ends at frame 40 (green line). The rest is done by extrapolation. I used extrapolation for the movement of the empty driving the clouds. Otherwise I would have had to adjust it in every scene, because the length of the scenes is different. Its constant slope leads to a constant velocity.

Cyclic:

Here again the animated part starts at the origin and ends at the green line. This extend mode is mostly applied for up and down motions of the chest to simulate breathing.

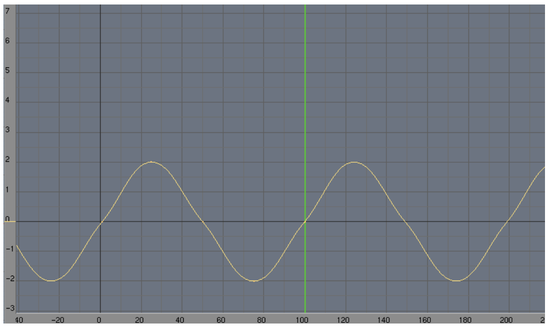


Figure 39 cyclic extend

Cyclic extrapolation:

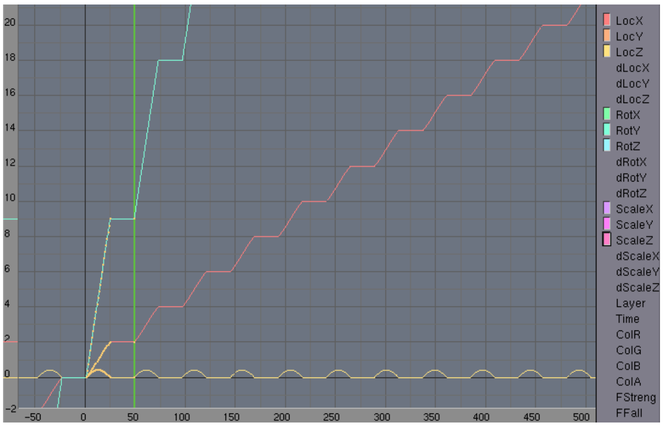


Figure 40 cyclic extrapolation

After 50 frames the part I animated is finished. The rest is done by the cyclic extrapolation. Having some experience in reading IPO-curves one can see what the movement is. Since location changes are always indicated by the motion of the pivot point, one sees that the center of the object must go up (LocZ →yellow) for about 12 frames and is at its initial height again after 12 frames. A forward movement is shown by the constant increase of LocX (red). And there is a continuous rotation around the y-axis (blue). We end up with the motion of a cube rotating around its edges.

Path animation

Objects may be animated along a path (Figure 41, upper part). As soon as the “follow path”-constraint is activated the origin of the objects location changes to the beginning of the path. The speed IPO (Figure 41, lower part) of the path defines the movement of the object. At the origin the object is at the beginning of the path. At a value of one on the y-axis the object is at the end of the path. The even parts of the speed IPO show that the object is at rest there.

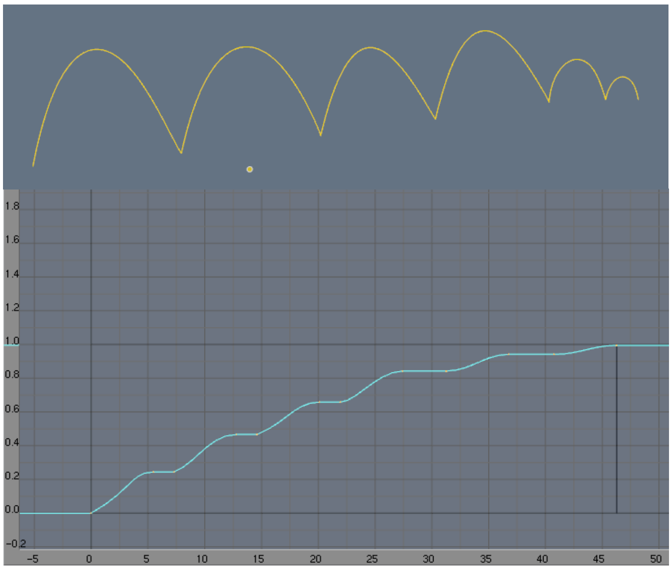


Figure 41 path animation

When it came to my first renders I soon realized that many things did not work as I wanted or expected them to. Motions were far too fast and facial expressions were not recognizable. I had to rethink and animate a lot of scenes. For example I had to keep in mind that when a body turns not every part starts at the same time. Or that one often stops to breathe when something new occurs or astonishing happens. The eyelids were parented to the body, because of the stretching and compressing they jumped out of the body. I had to parent them to three vertices around the eye so they stayed in the right place. My grass sort of jumped at some points in my animation. To find the frame where that happened I calculated the time the video was already running times the frame rate. I found out that after the wind fields "flew" across the screen I let them jump back to their initial position. I needed to smooth this motion. These are some examples of the problems I faced during the animating part.

3.10 RENDERING

Rendering images and even whole animations can be a very time-consuming task. It is the process of generating an image out of the data from 3D-objects. In addition to the geometry the data includes viewpoint, material, and lighting information. The rendering of an animation can run on a single personal computer for days or even longer. As I was working on my animation I to realized that it takes my laptop (2 GHz, 1 GB RAM) even some minutes to just render one complex image. Not much changed when I started working on a more powerful computer (2 x 2GHz, 2 GB RAM). I had to look for more power.

RENDERFARM

Big animation studios hire, or even have their own, huge renderfarm, where thousands of processors work together to generate an animation. There is also the possibility for private people to hire a renderfarm, but I was looking for a free possibility to render my animation.

Since my school had many computers not being used at night or during holidays I needed to investigate the possibility of setting up my own renderfarm there. After some discussions with the head of the IT section I got the permission for my renderfarm.

Setting up a renderfarm:

There are dozens of renderfarm or task-distributing software. After having checked and tested some of them I concluded “farmerjoe” to be the best solution. It is easy to set up and handle. All you need is a network and a share. I got my own share called R drive. First the software had to be installed on the share. Unlike many other software it was sufficient to adjust the configuration file on the master computer. This means entering the IP of the master and the paths of the output folder and the Blender software. Of course that had to be on the share to be accessible for all the computers. Because room P 1.02 is the only one with fixed IPs Mr. Brunnschweiler (head of IT RGS) offered to set up the farm there. Since the firewall was blocking the transactions between the master and the slaves (computers receiving a task) a port had to be opened. Another port was opened to make it possible to follow the progress in a browser window. Further preparation had to be done, namely to turn off the automatically shut down every evening and morning. This could have lead to a loss of frames and rendering time.

Then the master had to be told to be the master:

-enter in console : R:software/Farmerjoe_0.1.3/farmerjoe –master

The slaves had to be told to be slaves:

-enter in console : R:software/Farmerjoe_0.1.3/farmerjoe –slave

And progress surveillance needed be activated:

-enter in console : R:software/Farmerjoe_0.1.3/farmerjoe –appserver

To submit a job one has to load the file called “farmerjoe_submit.py” into Blenders text editor and hit ALT + P.

Following the workflow is possible by entering http://[IP]:[port] into the browser.

Example: http://172.16.4.50:2007

HOW TO SAVE RENDER TIME

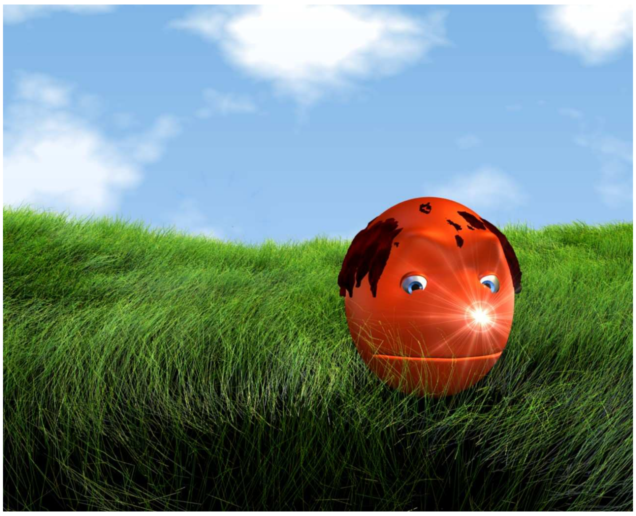


Figure 42 bug in scene 1 of the final animation

It is very important to chose jpg or another picture format as output, but not video! It may happen that in some frames constraints do not work properly (Figure 42). If you would have a video as output you needed probably to re-render a thousand frames just because of one or two wrong frames. In the end I had 6940 frames for my final animation and some 4000 frames of test renders. I needed to re-render about 150 frames because of such mistakes caused by bugs.

Of further importance is not to use ray tracing or radiosity settings[7]. They cause a huge amount of calculations and do not have a big impact on the image quality of such a project. One should also barely use the motion blur function. It gives fast motions a blurred effect to obtain a smoother motion. Thus it needs information from the previous and the next frame, leading to longer render times. I used motion blur in the second scene and I found out through tests that it may cause up to four or five times higher render times.

The outer most extreme of render time was reached with the material settings for glass. Such a frame may have run up to three or four hours on a single computer! The average render time was between 100 and 150 minutes. In comparison the average of all the frames not having glass settings was two minutes. That made it very profitable for me to look for frames being identical and copy them by myself.

In the end I needed about 1800 CPU hours to let the frames for the final animation render; Shared on 24 computers that means an overall render time of 75 hours, not having included the test renders. I have been setting up everything and rendering at RGS for two weeks in the summer holidays.

[7] Algorithms to follow reflected light beams, and let them be reflected again and again until the energy is used up, to get photo realistic renders.

3.11 POSTPRODUCTON

So far I just had thousands of rendered single images. To put them together to a video I used the software ImageToAVI. Then the single videos needed be compressed and converted. After that I imported them into Adobe Premiere Pro, one of the best video cutting software available. To still follow my goal of producing an animation free of costs I used a test version. I cut too long beginnings or endings of the scenes, speeded up or slowed down some motions and set different blend effects. Then sound could be added. To do so I used software called Mixcraft. It gives you the opportunity to compose your own songs and mix them with sounds of the included library. I ended up with two tracks for video and five tracks for audio settings.

The final video (Having A Ball) is on the CD accompanying my Paper. Also on the CD are some uncompressed samples of images out of the scenes and the whole development of my animation.

4 CONCLUSION

Making an animation needs a lot of patience. I started with not having known anything about the software I used neither about the process of making an animation. Especially in the beginning it was very hard to use Blender as it is quite unusual software. I spent months doing tutorials and reading manuals. After I knew how Blender is organized and the basics about modeling, I thought that each individual step of an animation is fairly easy. Which is actually true, the most difficult thing is to keep it together and bring it to an end. With my ideas for the story I could have extended the movie to an hour. The motions could have been smoothed and the timing brought to perfection. I could have worked on it for one more year without getting bored. But obstacles like bugs in the software did not allow me to dream of perfection. Especially when working on such a big project with a lot of libraries Blender's only weakness made working difficult. For example tracking or parenting constraints got lost during the rendering but also the animating part. Nevertheless I know there would not have been a faster and more comfortable way to animate. Blender is a powerful software which bears comparison with commercial 3D animating software. Looking back I am proud of what I have done and it makes me happy to watch the movie. I managed to produce an animation fully free of costs, except for my time. I am now able to use complex software and understand the way an animation is produced. I learned how to use gained knowledge in a big variety of ways. Aside from gained experience around computer animations I learned how to tackle a long-term project and how to get through it. I was not always convinced to have chosen the correct topic for my Matura work. In the end I can say I would do the same again. It was not always easy, but it was worth the pain and brought me some steps further in my own development.

5 ACKNOWLEDGMENT

My whole Matura work would not have been possible without the help of many people around me. In the following passage I want to thank all of them:

First of all I need to thank my mentor Clemens Wagner, who did not have it easy because of my extraordinary topic, for his faith in me and his effort for making a renderfarm at Romanshorn Grammar School possible. Further Marin Esteban for his help on software problems, story developing and spiritual succor and the whole IT staff of RGS for their effort and cooperation. Last but not least Christian Zanetti and his staff, who made it possible for me to render for two weeks in the summer holidays.

6 BIBLIOGRAPHY

Hess, D. Roland: *Animating with Blender*

How to create short animations from start to finish

Websites:

http://wiki.blender.org/index.php/Doc:Manual

http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro

Nystic.com

http://www.cgtextures.com/

25.07.09 for all websites

During the whole process of getting used to Blender and animating with it I gained a lot of knowledge which source cannot be clearly defined. Often it was a question of trying it out and applying learned things in different ways.

software used:

Blender 2.48:	modeling, animating, rendering
ImageToAvi:	jpg to video
GIMP II:	texture manipulating
Farmerjoe:	renderfarm
FormatFactory:	convert and compress videos
Mixcraft 4:	use of library sounds and composing own sounds
Adobe Premiere Pro:	cutting and adding music

7 TABLE OF FIGURES

All figures without quoted source were made by me.