TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Rustam Nadrshin 144840IVSM

FACIAL ANIMATION TOOL

Master's Thesis

Supervisor: Einar Meister
PhD

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Rustam Nadrshin 144840IVSM

NÄO ANIMEERIMISE TÖÖRIIST

Magistritöö

Juhendaja: Einar Meister
PhD

Tallinn 2017

Author's declaration of originality

I hereby certify that I am the sole author of this thesis and this thesis has not been presented for examination or submitted for defence anywhere else. All used materials, references to the literature and work of others have been cited.

Author: Rustam Nadrshin

09.05.2017

Abstract

People absorb information using all possible sensors of their body. Among all sensors, the audio-visual perception is the most important and helps to receive around 40 percent of overall information. That is why it is very important to improve human-computer interaction and see more human-like interaction in human-computer interfaces. The aim of this work is to develop an application of a virtual head which can visualize isolated phonemes and different facial expressions.

The current work explores existing solutions, which are presented by different implementations of virtual agents. Also, a comparison of different approaches to the implementation of the talking heads are done.

The main focus of this thesis is to find a way how to visualise speech articulation and facial expressions and provide software which enables to control the virtual head. As a result, the virtual head able to recreate movements of the lips, jaw and head during pronunciation of different phonemes and expressing emotions has been developed.

This thesis is written in English and is 34 pages long, including 6 chapters, 16 figures and 4 tables.

Annotatsioon

Inimesed hangivad ümbritsevast keskkonnast informatsiooni kõigi meelte abil. Kõige olulisemad on meie jaoks nägemine ja kuulmine, mille abil saame ca 40% teabest. Seetõttu on oluline parandada ka inimese-masina vahelist kommunikatsiooni ja arendada liideseid, mis võimaldavad kasutada inimesele omaseid informatsiooni hankimise viise. Käesoleva töö eesmärgiks on luua virtuaalse pea rakendus, mis võimaldab visualiseerida isoleeritud häälikuid ja erinevaid näoilmeid.

Töös uuritakse virtuaalsete agentide olemasolevaid rakendusi ja võrreldakse erinevaid lahendusi kõneleva pea loomiseks. Peamiseks ülesandeks on leida viis, kuidas visualiseerida kõne artikulatsiooni ja näoilmeid ning luua virtuaalse pea juhtimist võimaldav tarkvara. Tulemuseks on virtuaalse pea tarkvara, mis võimaldab animeerida huulte, lõua ja pea liikumisi erinevate häälikute hääldamisel ning visuaalselt esitada erinevaid emotsioone.

Magistritöö on kirjutatud inglise keeles ning sisaldab teksti 34 leheküljel, sealhulgas 6 peatükki, 16 joonist ja 4 tabelit.

List of abbreviations and terms

2D	Two Dimensional
3D	Three Dimensional
API	Application Programming Interface
BGE	Blender Game Engine
FBX	FilmBox
GUI	Graphic User Interface
HTS	Hidden Markov model based Text to speech synthesis System
ID	Identifier
MASSY	Modular Audio-visual Speech SYnthesizer
MPEG	Moving Picture Experts Group
OS	Operation System
SFAE	Simple Facial Animation Engine
TTS	Text To Speech
UI	User Interface

Table of Contents

List of figures

List of tables

1 INTRODUCTION

When people talk to each other and see the interlocutor's face, information is transferred and perceived quite differently, more precisely, information is conveyed by both auditory and visual channels. Also in human-computer interfaces we would like to see more human-like interaction which involves different modalities simultaneously. This paper introduces an approach how to present information in a more humanlike way. For this purpose, a virtual agent represented as an animated 3D head model has been developed.

1.1 Motivation

Starting from birth and throughout life, we absorb information in different ways using all possible sensors of our body. Among all these sources, the most important is the audio-visual perception of what is happening around us, because this is the way how we can understand the world. Moreover, we are so used to audio-visual perception that many people experience difficulties when they should do with vision or hearing only. For example, an interview by phone in a noisy environment can become a serious challenge.

It is quite difficult to evaluate the contribution of vision and hearing to the understanding of the surrounding world, because the part of the answers will be in the field of psychology. Since this work deals with visual representation of human articulation, it is worth mentioning that according to some studies, only through vision a human receives around 40[1] percent of information. That is why it is extremely important to improve human-computer interaction by introducing human-like virtual talking agents.

Actually, there already exists an application of Estonian talking head based on MASSY model[2]. [1] However, it has several limitations:

- it is for research and demonstration purposes only,
- it works in Internet Explore under Windows only,
- the head model cannot be used in other applications.

[1] http://www.creatingtechnology.org/papers/vision.htm
[2] http://massy-est.phon.ioc.ee/MASSY/peamudel.php

For the development of applications involving virtual talking agents it is necessary to have a new head model free of the restrictions mentioned above.

1.2 Problem Statement

During last decades, many resources and efforts have been directed to fill the gap between human and computer interaction. Different approaches and solutions have been suggested, which to some extent help to improve the human-computer interaction.

In virtual talking heads, one of the most challenging problems is visualization of the articulation in conversational speech, because the solution of this problem will be useful in different ways. For example, virtual talking heads can be used for educational purposes as animated tutors in order to improve the process of learning new languages, aids for hearing impaired, multimodal interfaces, and others.

The main purpose of this thesis is to develop a virtual head model for the visualisation of speech articulation. The software should run on different operating systems. In other words, the result of the work is a software containing a three-dimensional model of the human head able to animate the movements of the lips, jaw and head itself when:

- pronouncing different phonemes
- showing different emotions
- making head movements.

In this work, various approaches will be considered for visualization of articulation. The process of the animation of the three-dimensional model of the human head will be shown. It is very important to choose the right key points so that the animation is as close to a real human as possible. For these purposes, the animation will be implemented for each phoneme and emotion articulation with the utmost precision.

Last but not least, it is quite important to choose the most suitable development tools. Therefore, an analysis of existing programs and tools was conducted to identify the most appropriate environment which meet the following requirements:

- Possibility to create various animation types

- The solution should be cross-platform
- Open source programs should be used
- The ability to compile standalone application

1.3 Organization of the Thesis

Related Work. Literature review is done in this chapter. Main approaches are described and compared. Existing solutions are explored.

Methods and Tools. Various modelling methods overview is done in this chapter. Also some 3D tools are listed and presented.

Facial Animation. This chapter is presenting basic human emotions and phoneme description for Estonian.

Application. In this chapter is giving overview about making application in Blender. There is also installation and usage guides.

Summary. In this chapter gives a short overview and makes conclusion about thesis. There is also note about future work.

2 RELATED WORK

During last decade, a lot of time and effort has been spent on visualisation of human speech. A lot of research has been done in this area and number of articles have been published which try to apply different technics in order to solve this problem. In this chapter, we will discuss some of these publications. By nature, human speech is a combination of the two information streams – auditory and visual. In face-to-face communication humans use both information channels. For example, if speech is not heard clearly enough, then visual perception of articulatory movements can improve understanding of speech, by increasing the resistance to interference of data transmission [2]. The contribution of visual information in the perception of different speech segments is variable – for example, visual information enables the distinguishing of sounds like /t/ and /p/, but it does not enable to distinguish visually the sounds /p/ and /m/. Therefore, in virtual talking agents, it is extremely important to synthesize a human speech sound with adequate articulatory movements. The McGurk illusion is a well-known example of the complementary nature of multimodal speech perception – when an audio /ba/ is paired with a video /ga/ it is heard as /da/ [3].

2.1 Main approaches

Even though various implementations of the talking head have much in common in terms of visualization, it is possible to classify them as a parametric and an image based talking heads. In the case of parametric models human face geometry is represented as a polygonal mesh and in order to animate it, it is necessary to correctly determine the positions of the vertices of the mesh at a particular moment [4]. In the case of image based methods a large set of images or video clips is used and animation is produced by selecting the right frames for concatenation [5]. These approaches, despite of their obvious differences, can be used together and their correct combination can improve the final result [6].

2.1.1 Parametric talking head

A parametric talking head has a set of parameters which values are set at certain point of time. These parameters can control different articulatory movements of the face parts, for example:

- opening, widening and protrusion of the lips,
- jaw opening,

- tongue tip and body positions.

The usage of this parameters allows to set different positions of the face corresponding to different speech sounds. Then, in the process of facial animation a smooth transition from one position to another must be implemented which will create a realistic talking head.

2.1.2 Image based talking head

The image based virtual agents are based on the selection of the appropriate frame or the corresponding chunk of video. In order to be able to implement this approach, it is necessary to have a larger set of pre-prepared images or videos. For this purpose, the process of a pronunciation is recorded. Then the video frames are indexed by phonemes or phoneme sequences, and added to the database.

Figure 1. Segmentation face [7]

At first glance, the image-based approach looks very understandable and easy to implement. However, in order to use this approach it is necessary to solve one extremely complex problem – how to combine the sequences of images or videos so that the result looks smooth enough, like an entire animation. Because, even the smallest change in the facial expression or the angle of the head will be easily noticeable if they the change from frame to frame is not smooth. To

solve this problem, different methods of image and video processing, such as normalizing the orientation and position of the face, have to be implemented.

Another way to solve the problem of a "jumping" frame can be the segmentation of the face into different parts and the subsequent isolated work on them. This method was used in the Video-Rewrite project, where only the mouth area was processed among the whole face [8].

In alternative approach the face is divided into several areas (as shown in Figure 1) and stored as separate images, and later were combined to create a final visualization [9].

Figure 2. Example of the morphing between two images[3]

One more way of solving the problem of transition from one image to another, can be the use of morphing technology [10]. To create an effect, at least two images are used, on which the artist sets the key points (so-called markers or marks) that help the computer to perform the correct morphing. As a result of applying the morphing, a set of images about intermediate states are created by interpolating the existing data. Figure 2 shows a classic example of morphing.

2.1.3 Comparison of different approaches

In spite of the fundamental differences, both approaches can be used to solve various problems. Table 1 presents the main characteristics of these visualization methods.

[3] https://en.wikipedia.org/wiki/Morphing

Parametric	Image based
synthetic face	natural images
one instance of the face	many instances of one face
(articulation) parameters	database indexed by (classes of) phonemes
co-articulation (mostly) outside the face model	co-articulation (hidden) inside the face model

Table 1. Comparison of different approaches [6]

2.2 Examples of parametric models

2.2.1 MASSY

MASSY is a solution based on the parametric approach. [11] Simple text is used as input. Phonetic articulation module converts text into audio stream data, and generates information about the sequence of phonemes, pauses and their duration. Based on this data, the audio synthesis module generates an audio stream for subsequent playback, and visual articulation module generates information about face articulation motion. The face module visualizes information about the positon and adds an audio signal to create a complete audio-visual voice output. Figure 3 shows an overview of the system.

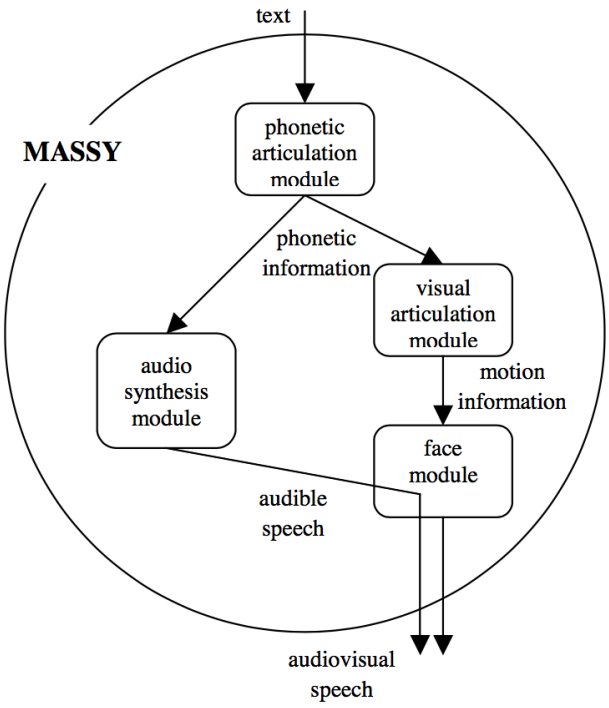


Figure 3. Schematic system overview of the MASSY [12]

MASSY uses parametric approaches for speech visualization. Thus, the result of visualization of speech can be photo realistic. It is possible to produce joint articulation, which considers all potential preceding and following speech segments. The separation of articulation and visualization enables the synthesizing of different speech styles.

2.2.2 Greta

Greta is a project which contains a 3D face model that corresponds to the specifications of MPEG-4[13]. The main goal of this project is the implementation of an animated model that can realistically visualize the dynamic changes of the human face.

Figure 4. The joy of Greta [13]

In this project, the Simple Facial Animation Engine (SFAE) was created, which contains 3D model of young woman head, which was called Greta. The Greta facial model consists of 15,000 polygons. Such a large number of polygons allows to achieve a very high level of detail. Such detailing is especially useful and shows good results particularly around the mouth and

eyes. This is not surprising, because these parts of the face play a crucial role when one person communicates with the other, as well as it is crucial when expressing emotions (see Figure 4).

A distinctive feature of Greta is the high quality of the 3D model. Another distinguishing feature is the generation of real-time wrinkles, as a result, the resulting face looks more realistic. Also, because of to the generation of various bulges and furrows, the look of the face seems more expressive.

2.2.3 Lucia

LUCIA is a talking head for the Italian language. [14] This project uses audio information from the Italian version of the Festival TTS. [15] The structure of the project is shown in Figure 3.

The LUCIA graphics engine is an engine for facial animation that is similar to other previously implemented MPEG-based projects. The main advantage of this project is a good coarticulation model used for face animation, which is trained on real data. Accuracy of reproduction of kinematic movements of articulatory parameters is very high. If we will examine the trajectory of articulatory parameters, we can see that the difference between the real and generated trajectories is, on average, no more than 0.3 mm.

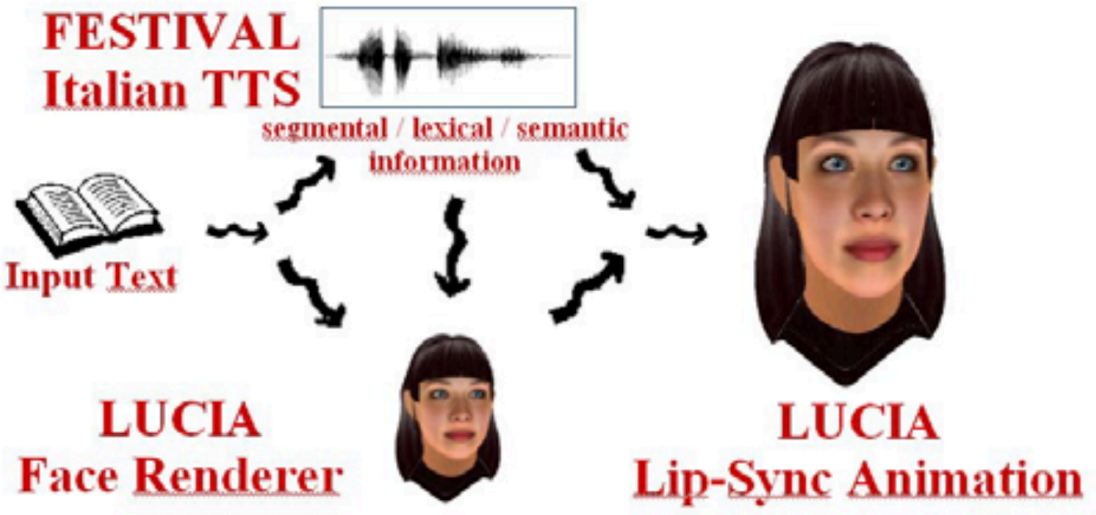


Figure 5. Lucia, an Italian talking head [14]

2.3 Video Rewrite

Video rewrite automatically creates new video based on the training data, where a person says words which he was not speaking in the original video. Phonemes in the new video and in the training set are labelled by Video Rewrite automatically. Then in the training set the mouth images are rearranged so they will match with the phonemes from the new track. If some phonemes required for the new video are missing in the training set Video Rewrite picks the closest approximations. The final sequence of the mouth images is attached to background video. [8]

In Video Rewrite computer-vision techniques are used for identifying on the training video the points on the speaker's mouth and morphing technique is used to create final video compilation based on the mouth gestures.

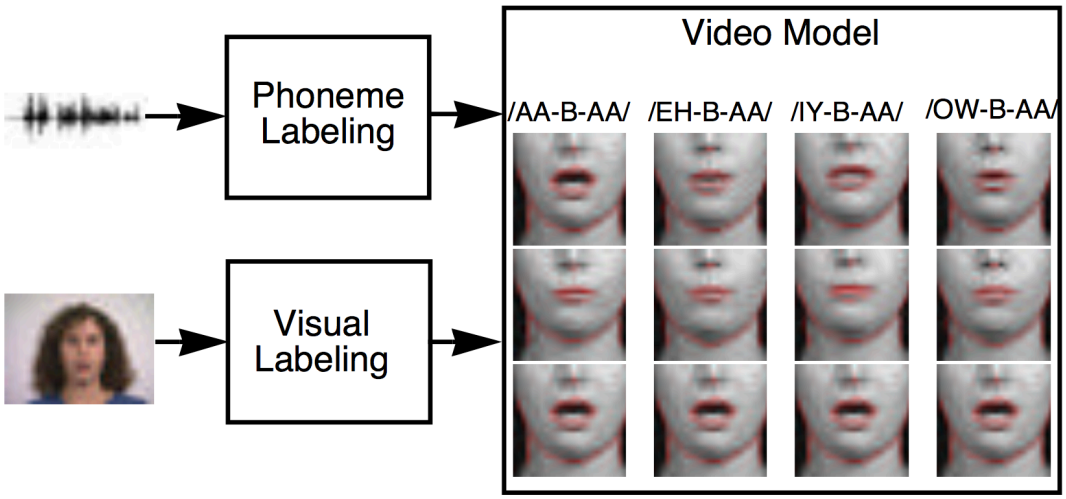


Figure 6. Overview of analysis stage [8]

New videos in Video Rewrite are created based on two steps: analysis of a training data and synthesis of a new video. Training dataset is automatically grouped into phonemes by Video Rewriter in the analysis stage. Facial features are automatically tracked in those groups. So, in the training data the visemes are entirely described by the phonemes and their group labels. This video database is used in the synthesis stage with the new utterance. In this stage the system extracts the corresponding viseme sequences automatically and uses morphing images to stitch them into a background scene. A video with jaw and lip movements is a result of this phase which is synchronized to the new audio. Figure 6 shows the steps of the analysis phase and Figure 7 shows the steps of the synthesis phase.

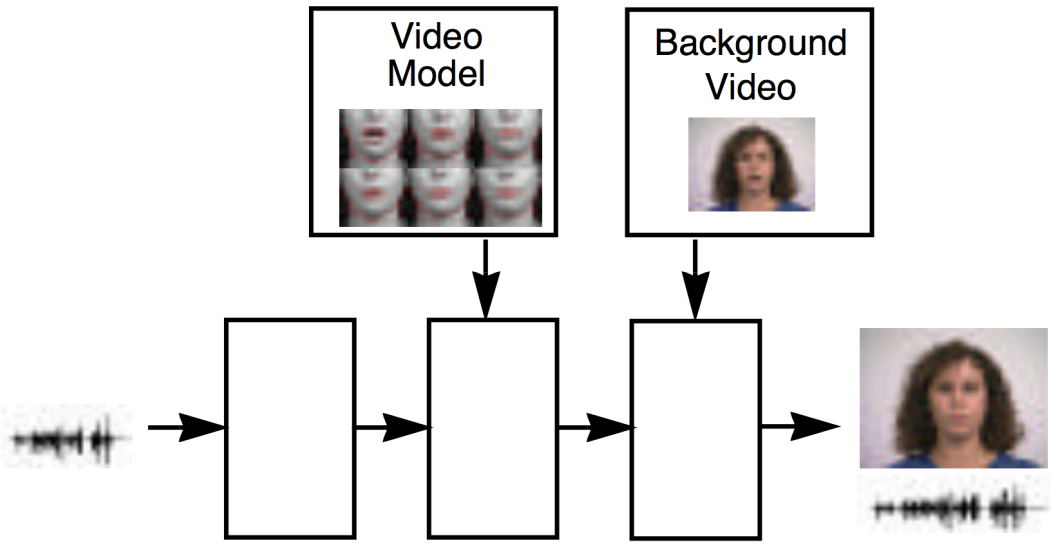


Figure 7. Overview of synthesis stage [8]

A small number of labelled images is required to model a new person in Video Rewrite. Based on the given dataset Video Rewrite learns the mimics of the person's mouth when they speak normally taking into account each detail of the mouth movement. Then database of video clips is created based on the dynamics of the mouth when person speak. This helps to present differences of individual speakers instead of using a general model which is a common practice in most facial-animation systems.

3 METHODS AND TOOLS

In traditional hand-drawn animation used in cartoons, the painting is done in 2D on paper, canvas, wood, etc. and only one of the sides of the object is presented. The picture itself is flat and if we want to get an idea about all sides of the object, then a number of pictures of the object should be drawn. Another method is called puppet animation. In this method, the puppet is made only once, then it is photographed from different angles and poses, which produces a series of flat pictures. A 3D visualization of the same puppets in digital form is possible with special tools like Blender, 3ds Max, Maya, Cinema 4D, etc. in which a 3D image, for example, a model of human body, can be created. In this chapter, an overview of modelling and animation will be presented.

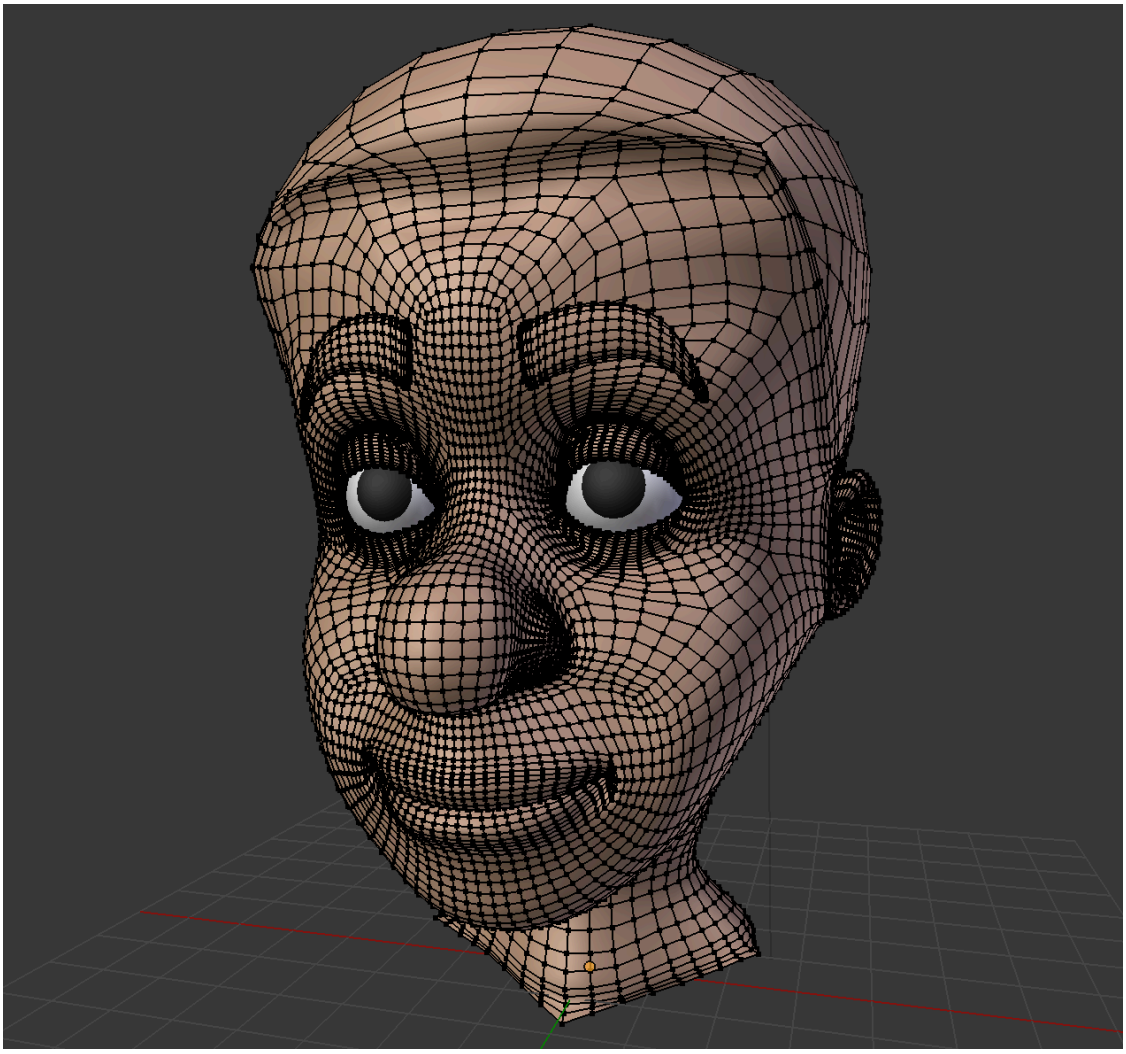


Figure 8. Example of the polygon mesh

3.1 Modelling methods

There are several ways of modelling, but the most popular one is the polygonal modelling. The main idea is that the topology of an object is represented in the form of simple geometric primitives called polygons (typically triangles or quadrangles). The surface of a 3D object is represented as a mesh consisting of quadrangular polygons (see Figure 8). If necessary, the quadrangles can be turned into triangles when exported to the game engine, but if antialiasing or tessellation is needed, the model consisting of quadrangles is preferred.

What is a tessellation? If an object is represented as polygons (especially organic objects, for example, a human), then it is obvious that the smaller the size and the larger the number of polygons, the closer the model will be to the original. This is the basis of the tessellation method: first a rough model is made from a small number of polygons, then tessellation is applied and each polygon is divided into 4 parts. So, if the polygon is quadrangular (and even better, close to a square), then the tessellation algorithms will give better a result.

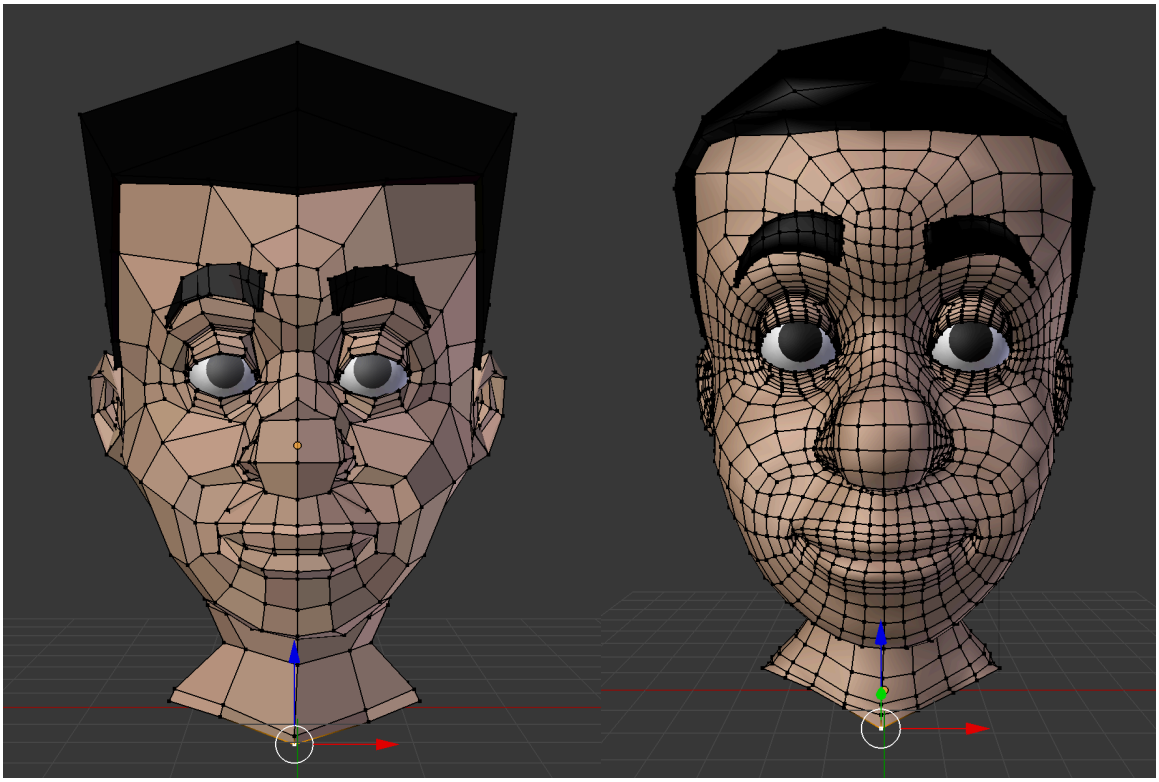


Figure 9. Low and high poly models

As stated above, the more polygons the closer the model can be to the original. But a large number of polygons has a negative effect which is a lower performance in animation. The more

polygons, the more points on which they are built, the more data need to be processed by the processor. Therefore, in 3D graphics always a compromise between detailed model and performance have to be found. In this regard, even the following terms arose: high poly and low poly, a highly polygon model and a low polygon model, respectively (see Figure 9). In games, low polygon models are used because the visualization is done in real-time. By the way, the models in games are represented by triangles in order to increase the productivity – graphical processors are able to quickly process hundreds of millions of triangles per second on the hardware level.

3.2 Animation methods

There are two main types of animation: keyframe based animation and skeletal animation. These types of animation are used in different situations and each has its own advantages and disadvantages.

3.2.1 Keyframe animation

In keyframe animation, the static meshes of the model are stored for each frame of the animation. If you will animate the model as in Figure 10, you will need to export six different static meshes and change the drawn mesh during each frame. Such animation is called keyframed animation because only key animation frames are exported. For example, in the animation in Figure 10 between each successive stage can be many intermediate frames which will make the animation smoother. However, you do not need to export them, because they are obtained by interpolating the successive key frames. For example, in linear interpolation, the location of each vertex of the mesh is linearly interpolated between the first and second frames.

One of the advantages of keyframe animation is the speed, because during the animation nothing needs to be calculated. All animation frames are stored in memory and during animation you only need to change the model each time. The disadvantage of this method is the need to store all model meshes in memory so that they can be quickly drawn. If the model has hundreds of animation frames, its mesh has to be saved hundreds of times. However, in scenes with hundreds of animated models that use the same animation, the keyframe method can be extremely useful.

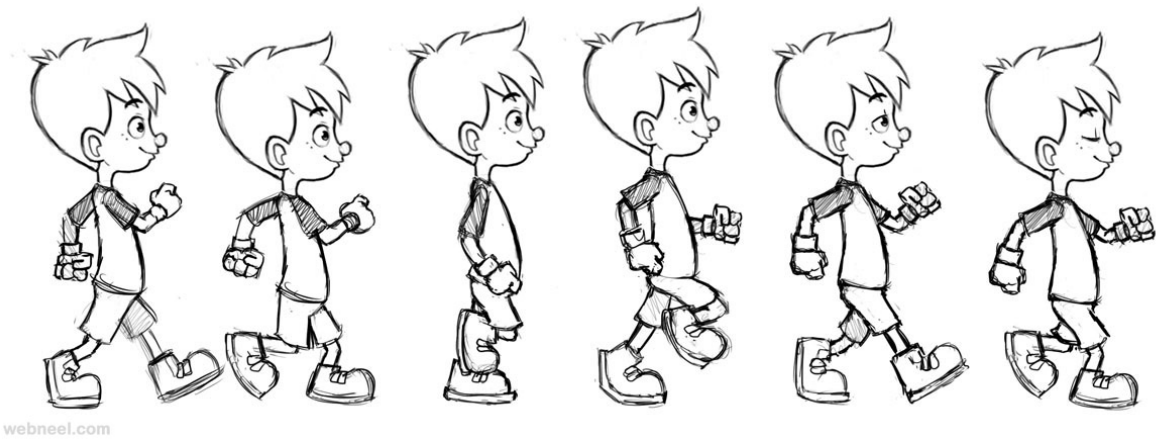


Figure 10. Key frame animation[4]

3.2.2 Skeletal animation

Another way of animating models is the skeletal animation. For this method, the skeleton of the model needs to be build which consists of several bones, and then each vertex of the mesh is attached to the bone of the skeleton. As a result, when animating the skeleton, the mesh connected to it follows the animation of the skeleton.

To build a mesh of the model with skeleton and its animation, various modelling programs can be used that support skeleton (or bone) animation, such as 3ds Max, Maya, Blender, etc. After the model is created, it is necessary to export it in a format that supports skeletal animation. Among the model formats, skeletal animation is supported by X (DirectX files) and FBX (Autodesk) files. It should be noted that skeletal animation is also animated with keyframes, which means that only key frames of skeletal animation are exported. As in the case of the keyframe animation, skeleton animation frames can be interpolated.

Skeletal animation has several advantages compared with keyframe animation. It allows to easily mix animations, allowing to apply different animations at the same time to the model. For example, two different animations can be applied to the model, when one animation will make the model walk, and the other look around (by turning head). Skeletal animation also allows to connect the bones of one object with the bones of the other. For example, if there is a character who wields a sword, a bone for the sword can be attached to the character's arm bone,

[4] http://webneel.com/walk-cycle-animation

which will make the sword to move when the character's arm moves. Nowadays, skeletal animation is used more widely than animation with key frames.

3.3 Modelling tool

Nowadays, a lot of software is created that allows you to make and work with 3D scenes and objects. Among them, there are both commercial and open source applications. These applications cover almost all aspects of human life. At first glance it is extremely difficult to determine which modelling tool will be the best option. So, in order to avoid the situation that after weeks or even months, choice was wrong and time for studying a 3D editor was wasted, you need to be well aware of the strengths and weaknesses of each particular program. Some of them will be described in more detail in this chapter.

3.3.1 Blender

The existence of this program proves that a free does not necessarily mean bad. Creating an application for 3D modelling and animation is an incredibly difficult task. And added to this that for such a titanic work the developer of the application will not receive any reward, other than the gratitude of 3D artists, makes the creation of a free 3D professional editor impossible. Nevertheless, the Blender project was not only started, but is also actively developing like any other commercial analogue.

The secret of success is the fact that anyone can work on Blender. Many of the tools that appeared in this program were added by completely different people who created certain functions to solve their own problems. However, it is worth mentioning, that Blender started as a commercial project, but was later closed and relaunched as an open source project.

One of the main advantages of the application is that it is cross-platform. Blender works equally well and stably on Windows, Linux and MacOS. In addition, the application has very low hardware requirements and can work even on devices with very low resources. The disadvantage of this application is that it can be hard to learn, because Blender can be used widely and can help to solve a lot of different tasks.

3.3.2 Maya

This application has been in the leading position for many years in the 3D modelling software market. It is very popular among professional 3D artists. This 3D editor is used by big and well-known studios such as Pixar, WaltDisney, Dreamworks and others.

The program has everything that is needed for creation of 3D graphics. Maya supports all the stages of creating 3D – from modelling and animation to texturing and rendering. This 3D tool can simulate body physics, emulate fluid effects, allows to fine-tune the character's haircut, animate hair, etc. The application's main feature is the PaintEffects module, which allows to draw with a virtual brush 3D objects such as flowers, grass, etc. The program is quite difficult to learn, which is compensated by a user-friendly interface.

3.3.3 3ds MAX

In 3ds Max, there are a lot of tools needed for modelling a variety of projects starting from the simplest primitives to the most complex 3D models. In addition this 3D-editor has tools for analysing and adjusting the lightening of 3D space. Also, photorealistic visualizer was integrated into the program, which allows to achieve high realism of the calculated image.

Despite of its complexity, 3ds Max is easy to learn, and the lack of any specific tool is compensated by a large base of plug-ins that significantly extend the standard capabilities of the application. For example, HairandFur plug-in has a lot of tools that look like a barber's arsenal, for instance, virtual hair can be combed, cut and even stylized based on the given form. Before this plug-in has been created, only professionals who have extensive experience with 3D and who know the secrets of imitating wool using textures and additional scripts written by them could work on hair animation.

3.3.4 Cinema 4D

Over the past decade, this 3d editor has become as popular as other successful and popular commercial editors, such as Maya or 3ds Max. Programmers of the German company MAXON Computer managed to precisely guess the niche, which remained free for a long time. The fact is that the most professional applications have always cost thousands or even tens of thousands of dollars. But the concept of Cinema 4D was built in such a way that the price of the application turned out to be democratic and at the same time the application remained interesting for

professional 3D designers and it was constantly getting better. The architecture of the application is very logical, and it is quite easy for a newcomer to understand it.

The toolkit of the program was constantly improving and expanding with very useful additives. Nowadays in Cinema 4D allows to create character animation, powerful photorealistic visualization system and, of course, it has convenient modelling tools. In the latest versions of Cinema 4D, the visualization algorithm has been substantially redesigned and the possibilities for processing 3D scenes have been expanded. The application allows to calculate the effects of global illumination and takes into account the dispersion of light.

3.3.5 Comparison

All the above-mentioned applications offer a wide range of capabilities for modelling and animation of 3D models. For this thesis, selection of the application has been done based on the following requirements: cross-platform, integration with third-party applications, free availability and existence of courses to learn how to use the application.

Cross-platform is extremely important requirement which makes sure to not be tied to a specific operating system. Shown below is the data regarding which operating systems a particular 3d tool works on. From Table 2 it can be seen that the Blender is the most accessible 3D editor, which works on practically any operating system. The least available editor is 3ds max, which is supplied by Autodesk company only for Windows operating systems.

	Windows	**MacOS**	**Linux**	**Other**
Blender	YES	YES	YES	YES[5]
Maya	YES	YES	YES	NO
3ds Max	YES	NO	NO	NO
Cinema 4D	YES	YES	NO	YES[6]

Table 2. Operation system support[7]

Integration with third-party applications is quite important, because in the future it will be necessary to use the model in the game engine in order to be able to build standalone

[5] Blender provide source code, which can be compiled for various OS
[6] AmigoOS
[7] https://en.wikipedia.org/wiki/Comparison_of_3D_computer_graphics_software

application. All selected editors support the most common data transfer formats, such as OBJ and FBX. So, it is not possible to identify the most suitable editor based on this requirement, because they all fulfil this requirement. However, in general, the integration with the game engine will be very important for choosing the 3D editor.

Among the programs which were described above, three are commercial and only Blender is an open source project. Therefore, if we consider accessibility, then Blender is the favourite, because it is totally free and has no limitations, unlike commercial products. However, due to the fact that blender is open source and performs a wide range of tasks it can be hard to learn, because unlike commercial products it is not supplemented by a sufficient number of high quality tutorials.

3.4 Game Engine

The game engine is the main software component of computer and video games or other interactive applications with graphics that are processed in real time. The game engine noticeably simplifies development process, because it provides basic functionality, such as, interaction with the user through various ways of data entry. One of the most important characteristics of the game engine is the ability to run on multiple platforms, respectively, the application built on that kind of game engine will be cross-platform.

In modern game programming, game engines are used as a foundation for the games. Game engines consist of many component-modules that implement the game functionality such as, displaying and processing graphics, sound, artificial intelligence, etc. All remains to do in order to create a game is only to build the engine functionality that will already correspond to specifically developed application. The modular design of game engines allows programmers to easily replace its parts, modify them to create new games with new models, improved graphics, sounds, a different script, modify existing functionality and add a new one. Because of this functionality, on the basis of existing engines a lot of new applications are created, while the effort spent to create that new applications is significantly reduced.

Within the scope of this diploma, several game engines were considered as an option for the implementation of the application which could visualize the 3D model of the human head. The

main requirement for the game engine was that it has to be open source. Below is a list of reviewed engines:

- BGE[8]
- OGRE3D[9]
- Irrlicht[10]

All this game engines are an open source projects. Comparison of these game engines with each other shows that BGE is the most developed and progressing project. The reason of it is that BGE game engine initially was integrated with Blender's 3D editor which significantly increases the possible number of users (software engineers) who chose to use this game engine. Also, it worth mentioning that the distribution of the final application is incomparably easier on BGE than on the other two game engines. Looking further, it can be also noted that BGE as a part of the Blender project has a clear development strategy, which makes it a better choice for long term usage.

3.5 Our choice

As a result of the analysis of available applications and taking into account the programs advantages and disadvantages, Blender and its game engine BGE were chosen as a development tool because it is the most suitable application for the requirements that were crucial for this diploma. An important factor of this choice is the initial integration of 3D editor and game engine, which will make the development process quicker. The main disadvantage of chosen application is that being an open source, a widely-used application and having a very complicated user interface makes it extremely difficult to learn.

[8] https://www.blender.org
[9] http://www.ogre3d.org
[10] http://irrlicht.sourceforge.net

4 FACIAL ANIMATION

In order to be able to create a facial animation for the 3D head model, it is important to understand how the facial muscles behave in different situations. Since the goal of this work is the visualization of the human face when it shows different emotions and the pronunciation of phonemes, it is necessary to understand from the theoretical point of view how exactly these processes occur. This can be achieved by understanding basics of the human face due expressing emotions. The pronunciation of phonemes is dependent on the language which is chosen for visualization. For this work, the Estonian language is chosen, and data on phonemes of this language will be given below.

4.1 Emotions

The initial study of this topic was conducted by the psychologist Paul Ekman, who started to study the recognition of emotions is 1960s. [16] His team of scientists was showing to their subjects photos of human faces that show different emotional states. Subjects had to classify the emotional states that they saw in each photo, from a given pre-determined list of possible emotions. Other studies conducted throughout many years used similar methods.

The Ekman's initial studies identify that there are six basic emotions, which he called universal emotions(see Figure 11). Universal emotions are listed below.

1. Joy (sometimes called "Happiness") – is symbolized by raising the corners of the mouth (an obvious smile) and squeezing the eyelids

2. Surprise – is symbolized the curving of the eyebrows, the eyes are wide open and reveal more white, the jaw is slightly lowered

3. Sadness - is symbolized by lowering the corners of the mouth, eyebrows coming down to the inner corners, and lowered eyelids

4. Anger - is symbolized by the lowering of the eyebrows, the lips are pressed hard and the eyes stick out

5. Disgust – is symbolized by the raising of the upper lip, the wrinkling of the bridge of the nose and the raising of the cheeks

6. Fear - is symbolized by the raising of the upper eyelids, opening of the eyes and the horizontal lips

There is also a seventh emotion, which is sometimes considered universal.

7. Contempt - is symbolized by shrinking of half of the upper lip and often the head leans slightly back.

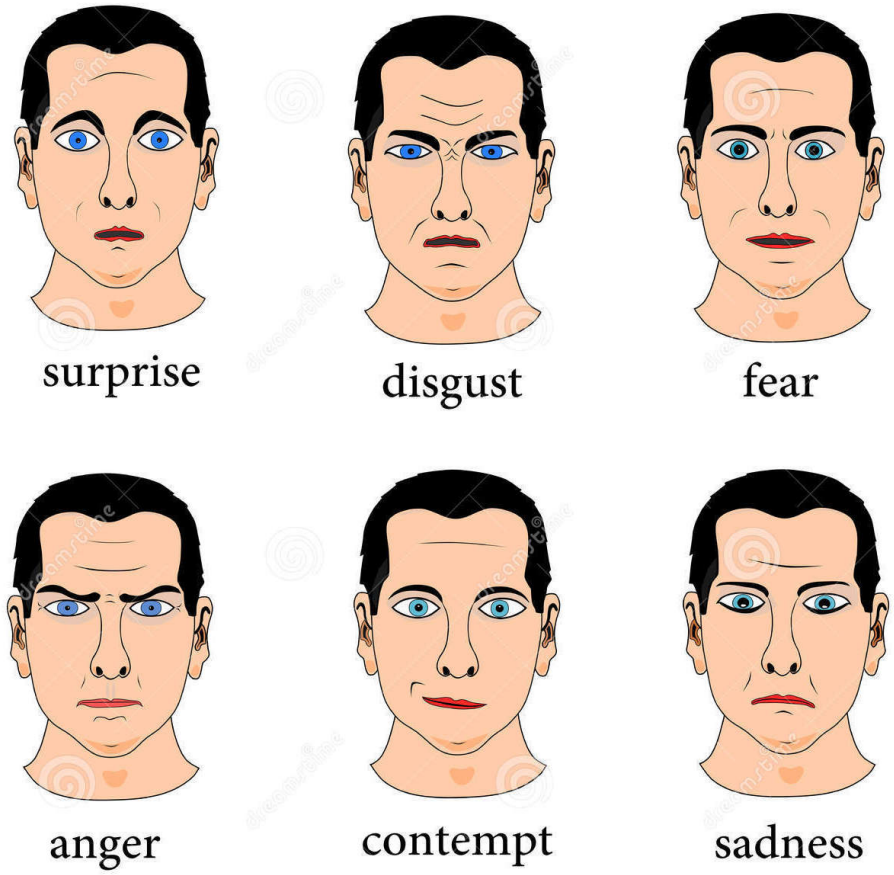


Figure 11. Example of Universal emotions[11]

Regarding this last emotion, there is some disagreement whether to consider it as a universal emotion or not. Because, to some extent, the contempt emotion is combination of anger and disgust emotions. However, despite this Paul Ekman in the 1990s added this additional disdain emotion to the list of universal emotions. [17]

[11] https://www.dreamstime.com

What about the other emotions that we feel, such as feelings of guilt, shame, jealousy and pride? Although we sincerely consider them as emotions, we do not demonstrate clear and obvious expressions for them. Probably, that's why many people can hide these emotions from others – because usually these emotions do not appear on their faces.

4.2 Estonian

Articulatory gestures are specific movement actions, through which targeted articulation is realized in speech. They are complex and include the coordinated movement of all the speech organs that participate in articulation, such as lips, tongue and jaw. Estonian has in total of 26 different phonemes. Among this, 9 phonemes are vowels and 17 are consonants. [10] Table 2 describes the articulatory parameters of all Estonian phonemes.

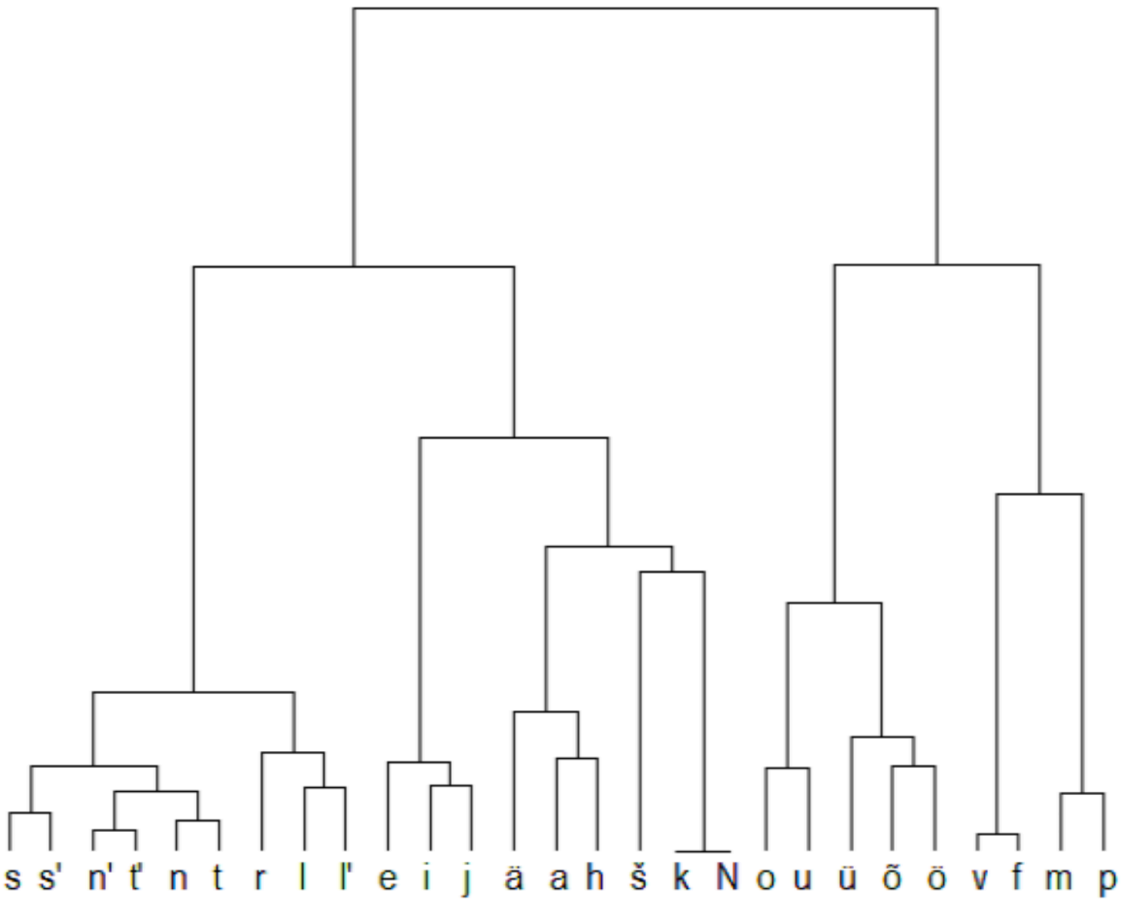


Figure 12. Cluster analysis of Estonian phonemes on the basis of articulatory features [1]

Phoneme	Description[12]
i	illabial high front vowel
e	illabial mid-high front vowel
ä	illabial low front vowel
ü	labial high front vowel
ö	labial mid-high front vowel
u	labial high back vowel
o	labial mid-high back vowel
õ	illabial mid-high back vowel
a	labial low back vowel
p	bilabial voiceless plosive
t	denti-alveolar voiceless plosive
t'	denti-alveolar voiceless palatalized plosive
k	palato-velar voiceless plosive
f	labiodental voiceless fricative
v	labiodental voiced fricative
s	alveolar voiceless fricative
s’	alveolar voiceless palatalized fricative
š	postalveolar labialized voiceless fricative
h	glottal-oral voiceless fricative
m	bilabial voiced nasal
n	alveolar voiced nasal
n’	alveolar voiced palatalized nasal
l	alveolar-postalveolar voiced lateral
l’	alveolar-postalveolar voiced palatalized lateral
r	alveolar voiced trill
j	palatal approximant

Table 3. Articulatory description of Estonian phonemes [1]

[12] https://en.wikipedia.org/wiki/Uralic_Phonetic_Alphabet

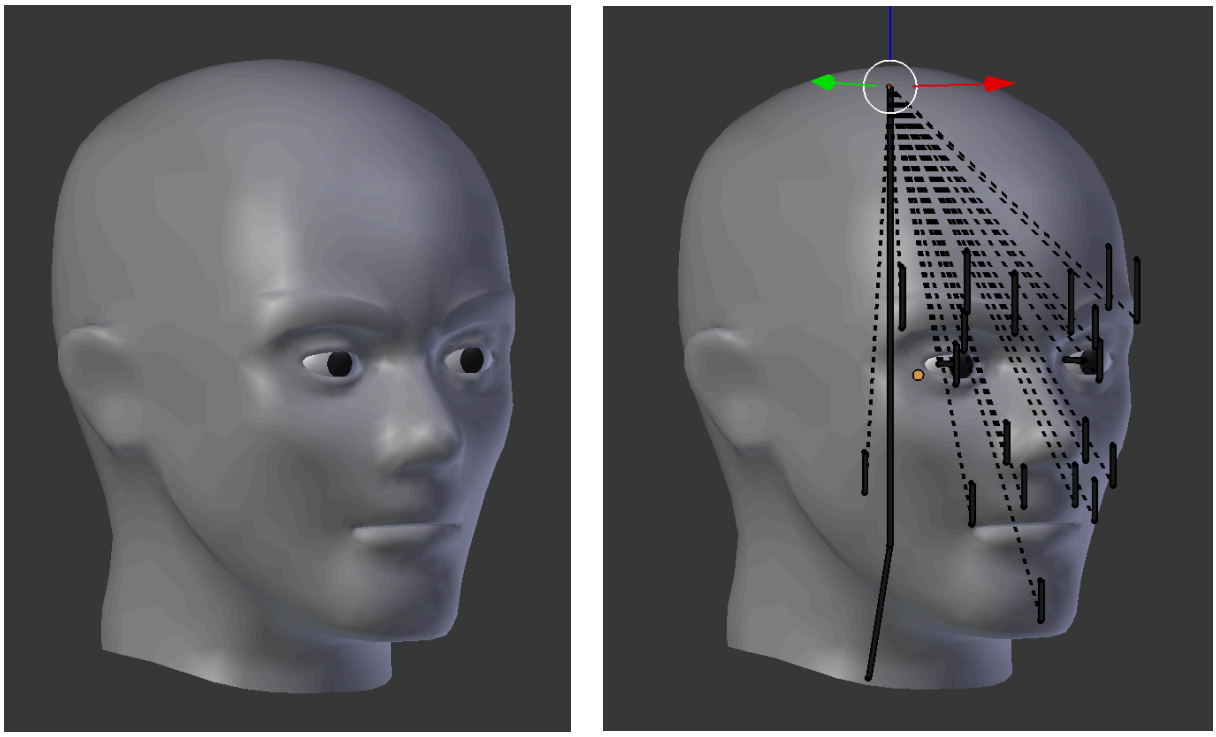
As was stated earlier, the unique sounds used to create words are called phonemes. The shape of the mouth and the location of the tongue at which the sound is created are called visemes.

In order to determine the parameters of the visemes that correspond to the phonemes of the Estonian language, an analysis of the articulatory parameters has been done for the pronunciation of various phonemes. During the initial visual analysis 12 basic visemes were identified which can represent all the phonemes of Estonian language. As a result of further various experiments on the pronunciation of individual sounds as well as phrases, a more detailed model of classification of phonemes has been created. Figure 12 shows the final cluster diagram.

5 APPLICATION

Taking into consideration all the advantages and disadvantages of different approaches of implementation of the virtual agent, it was decided to choose the parametric approach as a basis. The main element of this speech visualization approach is the model of a 3D head. For this purpose, a 3D head model from a tutorial[13] has been used. This model can be animated quite realistically, which allows to visualize various human emotions and speech animation. This chapter describes the process of the building an application in Blender.

Before talking about making application in Blender is worth mentioning one extremely useful resource, which is the Blender Essential Training[14] course from study portal Lynda. As it was mentioned before, Blender is extremely complicated to learn, so it will take a lot of time if you try to learn it by yourself, without using any tutorials. But the course mentioned above will help to understand how to work with Blender more quickly.



(a) (b)

Figure 13. 3D head model mesh(a) and armature(b)

[13] http://cgi.tutsplus.com/tutorials/create-a-facial-animation-setup-in-blender-part-2--cg-32333
[14] https://www.lynda.com/Blender-tutorials/Blender-Essential-Training/87088-2.html

5.1 Animation

Blender allows creation and animation among different forms of a single mesh by using shape keys. There are many use cases for shape keys, among them one of significant task is setting up facial animations. A 3D model (see Figure 13a) which will be used in building a demo application that has a set of shape keys. These shape keys are controlled by drivers, which are shown in the Figure 13b.

Application should visualize isolated phonemes and facial expressions. For this purpose we should create an Action with keyframes for each particular phoneme or expression visualization using defined drivers. These actions are containers for the data when objects and properties are animated. So, when an object is animated by changing location, rotation or scale factor, the data about it is stored in Action. It was decided to make all animations with the same duration equal 60 frames. Figure 14 shows one of the Actions.

Figure 14. Joy emotion animation Action

In order to create animation for each phoneme at the first stage, the information gained during adjusting MASSY model to Estonian will be used. [1] Table 4 shows data about articulatory parameters for each phoneme. Narrow Lips column defines the width of lips in horizontal plane, it varies from -1.0 (means maximally wide) to 1.0 (means maximally narrow). Open Jaw column defines position of the jaw and varies from 0.0 (means close position) to 1.0 (maximally open position). The second stage is to experimentally make an animation more similar to Estonian talking head based on MASSY model.

Emotion animations have been done manually based on description provided in Chapter 4.1. Afterwards they have been adjusted in experimental way in order to be similar to human emotion expressions.

Phoneme	Narrow Lips	Open Jaw
i	-0.7	0.4
e	-0.6	0.6
ä	-0.3	1.0
ü	0.9	0.1
ö	0.9	0.4
u	1.0	0.3
o	1.0	0.5
õ	0.5	0.5
a	-0.3	1.0
p	0.0	0.08
t	0.0	0.2
t'	-0.2	0.2
k	0.0	0.3
f	0.0	0.2
v	0.0	0.25
s	-0.3	0.4
s’	-0.4	-0.4
š	0.4	0.3
h	0.0	0.5
m	0.0	0.06
n	0.0	0.2
n’	0.0	0.3
l	0.0	0.55
l’	-0.2	0.5
r	0.0	0.7
j	-0.3	0.4

Table 4. Articulatory parameters of Estonian phonemes [1]

5.2 BGE

When model and animation for it has been created, the main question is about how to allow the end user to manage these animations playback. To solve this problem, we will need a game engine, in this case, the BGE has been chosen as a game engine, which is one of the components of 3D tool Blender. The main purpose of BGE is a game development, but it can be also used to build any interactive 3D programs, such as the talking head in this case.

The main component of BGE is the Game Logic Panel. Because of the Game Logic it is possible to perform any actions in the application. This is a powerful high-level tool that allows to create application logic through a graphical interface, which means that there is not necessity to write code.

The main elements of Game Logic are Logic Bricks. A logic brick is an elementary function that contains a set of customizable parameters depending on the task. Accordingly, by configuring and combining Logic Bricks, it is possible to create an application of different levels of complexity. Logic bricks can be in three types: sensors, controllers and actuators.

Sensors allow the application to perceive various external "irritations". All logical actions start from the sensors. When the sensor is triggered, it becomes positive, for example, when the keypad key is pressed, a positive signal is sent to all controllers connected to this sensor.

Controllers perform processing of signals received by sensors. It is possible to perform 8 different logical operations:

- AND
- OR
- XOR
- NAND
- NOR
- XNOR
- Expression
- Python

Actuators can perform various actions on this or that objects. Actuators perform actions, such as: moving and adding objects, playing sounds, playing animations. Actuators are activated when receiving a positive signal from one or more controllers.

For our needs sensor “Always” was used, which sending signal to python controller. Python controller in its turn controls all application process from code behind and connected to default actuator. Figure 15 shows application game logic panel.

Figure 15. Application game logic panel

5.3 Application logic

As it was mentioned in the previous chapter application processes are controlled by Python controller. It means that an application UI and playing an animation has been done from code behind.

Blender doesn’t provide necessary tools for building user interface. It means that if some buttons, tables, etc. are needed they should be modelled as separate objects. Such an approach is time consuming. But fortunately, there are a few open source projects that handle it and provide high level API for building UI.

BGUI[15] is an open source framework written in Python that handle GUI in the BGE. Widgets are the main elements of the framework presented in the form of buttons, labels, etc. During application development the layout was implemented which contains UI elements. Image widgets have been used in order to display images with list of phonemes and emotions, Text widget for allowing user to enter data and Button widget to enable user interaction with the application by playing particular animation.

[15] https://github.com/Moguri/bgui

When user presses Play button, the callback method is called. This method is looking for an Actuator connected to Python controller (see Figure 15). After this, Python script is making setup of an Actuator with following parameters:

- Action name based on user input
- Frame start position
- Frame end position
- Playback mode

When the actuator setting is completed, the script activates it and user can see the animation playback of chosen animation.

5.4 Installation

Since one of the main requirements for application was convenience in installation and use, the following instruction can be useful.

Demo application can be found on the attached media drive for this paper or can be downloaded from temporary link.[16] The structure of the both sources is the same. There are three folders:

- MacOS
- Windows
- Project

In MacOS and Windows folders, you can find the version of the application for the appropriate platform. Both folders contain ZIP archives, which should be extracted and copied to Application/Program Files directory in operation system. Because an application is not signed by appropriate certificate, operation system may block it. In this case you will need to do the following procedure for each platform:

- MacOS: Open System Preferences, choose Security & Privacy settings, allow application downloaded from Anywhere
- Windows: Run as administrator, under more menu allow 3rd party applications

[16] https://goo.gl/oeVmsH

The application was tested for operating systems macOS Sierra (version 10.12.4 or higher) and Windows 10.

Project folder contains the blend file as well as the Python script and BGUI library. Blender 2.78 recommended for use.

5.5 Usage

The application is very easy to use. Figure 16 shows application interface. In the upper left corner, there is a table which shows Estonian vowels with given IDs. Next to it there is a list of emotions with their IDs. Estonian consonants with corresponding identifiers listed in the top right corner. There is a field in the lower left corner, where could be entered phoneme or emotion ID from the lists. After entering appropriate number and pressing the Play button head which is in the center of the screen displays corresponding animation.

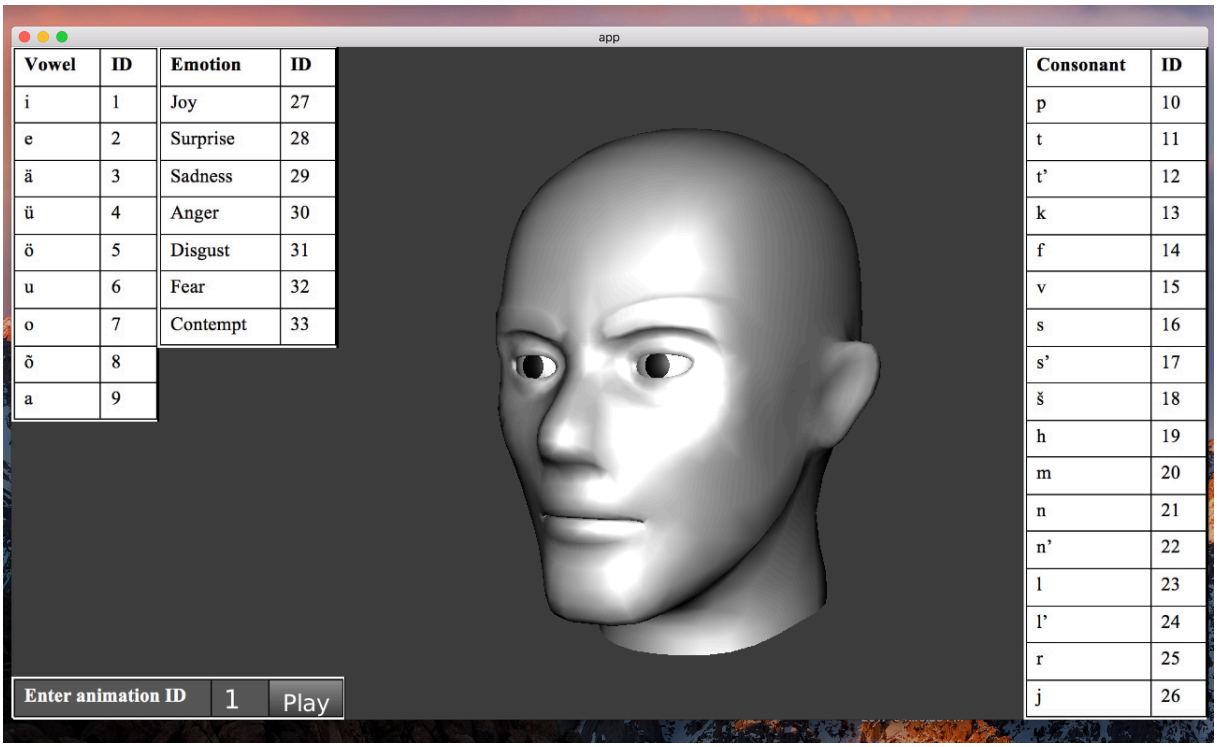


Figure 16. Application interface

6 SUMMARY

This thesis investigated a promising area in human-computer interaction. More concretely, it means that human-computer interaction can be improved by adding human-like behaviour. It is done using virtual agents which are represented by various implementation of talking heads. In order to understand the current state-of-art, different implementations of talking heads have been explored.

In face to face communication, people perceive information not only through hearing, but also by vision. In order to replicate the visual experience, it is important to have a knowledge about the details of the visual image of the interlocutor. To understand these fundamentals, we considered the basic human emotions suggested by Paul Ekman, and also examined the articulatory movements of the lips and jaw when pronouncing different Estonian phonemes.

In the current thesis we created an application that shows visual 3D representation of human head. This application displays different emotion expressions and isolated phoneme articulation. Application has been developed in Blender environment. This way was chosen because Blender has all necessary tools, starting from 3D editor and ending with integrated game engine that allows to build a standalone application without redundant dependencies.

6.1 Future work

In the future, we would like to improve the application, so it can show not only isolated phoneme but also visualize continuous speech with simultaneous audio playback which has a huge potential in different areas, such assistant for deaf people, learning purposes, etc. For this purpose, the virtual head will be integrated with a text-to speech synthesizer which produces audio output and the information necessary for simultaneous visual animation (phoneme sequence and duration of each phoneme). One possible way to do it is to use HTS engine[17].

In addition, the head model should have a tongue which will contribute to better visual distinction of consonants.

[17] http://hts-engine.sourceforge.net

During the Skype University Hackathon 2017[18] my team attempted to implement human speech visualisation. Due to limited time and resources we didn't manage to complete the project. Nevertheless, the project won the nomination in one category.

[18] http://aka.ms/unihack2017fb

Bibliography

[1] E. Meister, S. Fagel, and R. Metsvahi, “Towards audiovisual TTS in Estonian,” *Front. Artif. Intell. Appl.*, vol. 247, pp. 138–145, 2012.

[2] W. H. Sumby and I. Pollack, “Visual Contribution to Speech Intelligibility in Noise,” *J. Acoust. Soc. Am.*, vol. 26, pp. 212–215, 1954.

[3] H. McGurk and J. MacDonald, “Hearing lips and seeing voices,” *Nature*, vol. 264, pp. 746–748, 1976.

[4] J. Beskow, “Rule-based Visual Speech Synthesis,” in *4th European Conference on Speech Communication and Technology*, 1995, pp. 299–302.

[5] T. Ezzat and T. Poggio, “Visual Speech Synthesis by Morphing Visemes,” *Int. J. Comput. Vis.*, vol. 38, pp. 45–57, 2000.

[6] S. Fagel, “Merging methods of speech visualization,” *ZAS Pap. Linguist.*, vol. 40, pp. 19–32, 2005.

[7] J. Beskow, “Talking Heads. Models and Application for Multimodal Speech Synthesis,” 2003.

[8] C. Bregler, M. Covell, and M. Slaney, “Video rewrite: Driving visual speech with audio,” in *SIGGRAPH*, 1997, pp. 353–360.

[9] E. Cosatto and H. P. Graf, “Photo-realistic Talking Heads from Image Samples,” *IEEE Trans. Multimed.*, vol. 2, no. 3, pp. 152–163, 2000.

[10] T. Beier and S. Neely, “Feature-based image metamorphosis,” *ACM SIGGRAPH Comput. Graph.*, vol. 26, pp. 35–42, 1992.

[11] S. Fagel, “MASSY - a Prototypic Implementation of the Modular Audiovisual Speech SYnthesizer,” *English*, pp. 2553–2556, 2003.

[12] S. Fagel and C. Clemens, “An Articulation Model for Audiovisual Speech Synthesis - Determination, Adjustment, Evaluation,” *Speech Commun.*, vol. 44, pp. 141–154, 2004.

[13] S. Pasquariello and C. Pelachaud, “Greta: A Simple Facial Animation Engine,” *Soft Comput. Ind. - Recent Appl.*, pp. 511–525, 2002.

[14] P. Cosi, A. Fusaro, and G. Tisato, “LUCIA a New Italian Talking-Head Based on a Modified Cohen-Massaro’s Labial Coarticulation Model,” in *8th European Conference on Speech Communication and Technology*, 2003, pp. 2269–2272.

[15] P. Cosi, F. Tesser, R. Gretter, C. Avesani, and M. Macon, “Festival Speaks Italian!,” in *7th European Conference on Speech Communication and Technology*, 2001, pp. 509–512.

[16] P. Ekman, E. R. Sorenson, W. V Friesen, N. Series, and N. Apr, “Pan-cultural elements in facial displays of emotion,” *Science (80-.).*, vol. 164, pp. 86–88, 1969.

[17] P. Ekman, “Basic Emotions,” *Handbook of cognition and emotion*. pp. 45–60, 1999.