How to use abdnthesis.cls

Timothy J. Norman

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of the
University of Aberdeen.

Department of Computing Science

2010

Declaration

Abstract

An expansion of the title and contraction of the thesis.

Acknowledgements

Much stuff borrowed from elsewhere

Contents

Chapter 1

Introduction

The game industry now is bigger than ever before and still growing. Along with technological advancements as well as rise in popularity, games themselves become bigger and more polished. With increasing size and quality, the number of man-hours rises drastically. A lot of work is being put into creating tools that enable faster creation of content. However, there is still a lot left to be optimized and automated.

One domain of game development that suffers that drains lots of man-hours into monotonous processes that could potentially be automated is animation. Most games will rarely animate everything by hand as there is too much content to cover. While some animated content needs to be very polished - usually called cutscenes (action sequences, parts of game that greatly influence the plot development), some animation might be cruder (dialogue sequences). Games like RPGs will feature a lot of dialogue - during the dialogue the characters cannot stand still as it would negatively impact player's immersion in the game world. The characters must move and perform gestures that naturally underline their speech. These animations cannot be all done by hand because of the sheer amount of content needed. For instance:

- Mass Effect Andromeda and Fallout: New Vegas features 65,000 lines of dialogue[1][2].
- The Witcher 3 features roughly 35 hours of dialogues[3].

The main challange is to make the dialogue scenes (and other automatically generated animation) look indistinguishable from cutscenes. In many games, player will be shown good, well-polished animation that immediately switches to poor, clunky and unrealistic animation. The less perfected dialogue scenes break the immersion of the player and negatively impact the overall experience. Easier, faster and better quality methods of generating dialogue scenes would be a great asset to the gaming industry.

1.1 Motivation

The purpose of this project is to develop a tool that helps generate animated dialogue scenes and minimizes the amount of manual work by using natural language processing. Generating the scenes directly from script would pose several benefits:

[1]https://www.pcgamer.com/mass-effect-andromeda-has-over-1200-speaking-characters/
[2]https://www.pcinvasion.com/fallout-new-vegas-will-have-65000-lines-of-dialogue
[3]https://www.pcgamer.com/most-of-the-witcher-3s-dialogue-scenes-was-animated-by-an-algorithm/

- The script is written to outline the plot of the game. The same script could be fed into a program to generate the animations.
- The script is semi-structured natural language. By allowing natural language, the program helps shorten the gap between artists and writers and animators and technicians.
- The program can be used for prototyping scenes quickly and easily.
- The program can be used by people who know nothing about animation.

The program I propose would create prototype animation with almost no amount of work required. This can be use to either save time or to use the saved time to manually adjust the animations.

1.2 Objectives

The Projects Objectives are as following:

Develop a tool able to interpret a natural language script

The tool must be able to read a semi-structured script and recognize dialogue lines, emotions of characters and actions performed by characters.

Develop a tool able to blend a final dialogue scene

The tool must be able to output a fully editable dialogue scene. The scene is assembled using pre-made motion capture clips.

The scenes created by the software will be very crude and unpolished. Scenes generated by the tool will need to be polished manually, the amount of polish can be decided by assessing the importance of a given scene. However there is a chance that the scene quality will be much inferior to scenes generated by similar tools that use different approaches. Therefore an important question this project tries to answer is whether taking the NLP approach to animation is feasible in the games industry given current technology.

Chapter 2

Background and Related Work

2.1 Background

Manually crafting every animation in the game is unrealistic due to cost and time requirements. Many games have employed various approaches to computer generated animation in order to generate hours of realistic content. No game however has succeded in making the dialogue animation indistinguishable from manually animated cutscenes.

2.2 Existing Systems

There has been a variety of approaches featured in games. Many of them ended up generating dialogue scenes that are higly repetitive, not very realistic and in general not mathing the sentiment and emotion of the speech with movement. The only system that did not seek to find cheap workarounds around the issue and instead embraced the full complexity of the problem is the dialogue sene generator used in The Witcher 3. The system developed by CD Projekt Red made computer generated dialogue sequences in many cases barely distinguishable from those made by an artist, allowing less important scenes to be left completely untouched by a human animator. [4]

The tool created by CD Projekt Red takes information on initial state of involved characters (position, stance, emotions, etc.) and audio recording of the dialogue lines. The tool chooses matching premade animated clips and outputs a fully editable animated scene of characters conversing with one another. The tool uses audio recordings to aid the animated clips (analyzing the audio waves may help decide when characters accent or underline some information). This tool is the current state-of-the-art and has produced the best effects in terms of amount of work to quality of animation ration. However, there are some serious drawbacks to this project. It still requires a fair amount of work as for every scene the initial state must be specified manually. Moveover, the system requires audio to be recorded first. Moreover, the tool is not released to be used commercially outside CD Projekt. [10]

The tool I propose would hardly be able to compete with that of CD Projekt Red, however it would have some significant advantages. It would make generating the scenes even faster (requiring less manual work and preparation) and would in general be more appealing to small developers and people who are not animation experts.

2.3 Related Work

The main focus of this project is the usage of NLP for generating animated sequences. While this project puts particular emphasis on generating scenes of dialogue, there exist a multitude of projects that explores the usage of NLP in animation in a variety of ways.

A very early research (1991) explores usage of NLP for creating animations that would help engineers demonstrate tasks in an easy and safe way (demonstrating tasks personally might be unsafe, reading manuals might be insufficient to understand the task in full) [6]. The system would take as input a set of natural language *directives* or *commands* (e.g. *move cup to table*). The system would interpret such an instruction into a series of steps (tasks) that are carried out in a given order. Based upon that sequence an animation would be generated.

The project hovewer seems to have a few significant problems. Most importantly, the end results was not editable. In my research I believe that the end results will not be immediately satisfactory without any manual improvements and I believe that the outputted scenes should be fully editable. The other issue with this project is that the end result is not realistic or immersive (this was not a priority of that research, but is important for me). The animations were automatically generated in full, which I do not believe to be a viable approach for my project. To improve realism of the scenes, the animation should be created using motion capture clips.

A lot more research has been done in this area with certain degree of success. Another interesting paper explores a topic more similar to the focus of this report. `GeneratingAnimationfromNaturalLangua` proposes using a database of pre-recorder motion data instead of filly generating the movements [7]. The paper argues that motion synthesis requires too much manual setup (which a normal user might find too hard). The paper also proposes an architecture of a database - a way to store motion capture clips and their associated metadata. The system described in this paper is however essentially different from what I propose in this report, as the described report focuses on action-driven animation, while my tool would focus on dialogue-driven animation.

There exists research that also focused on animations that feature dialogue. [9]

So far, the main focus of similar research has been generating animated scenes from text with focus on actions and sometimes motion synthesis. There exists no record of employing sentiment analysis to create animated dialogue scenes.

2.4 Emotion Analysis

The task of emotion analysis is a subset of natural language processing. The task of analysing natural language text in search of subjective information such as sentiment or emotion is known as sentiment analysis.

2.4.1 Sentiment Analysis

Sentiment Analysis can be broadly defined as a computational approach for discovering opinions and attitudes expressed in text by opinion holders. In its most basic form it focuses on binary classification of the sentiment of the opinion holder (positive or negative) [8], but can be extended to mine for more complicated opinions such as emotions or detecting sarcasm. One of the popular uses of sentiment analysis is predicting stock market behaviour, as well as getting immediate feedback on products, political campaigns, decisions by monitoring social media [3].

Approach to sentiment analysis might differ depending whether focus is on document-level analysis or sentence level analysis - where document-level analysis assumes the entire document to have one clear area of focus, while sentence level analysis analyzes each sentence individually. Apart from those, there are more types of sentiment analysis such as aspect-based analysis (which is use when the document or sentence does not focus on a single entity), or comparative analysis (when direct opinion is not desired, but it is needed to contrast opninions with each other) [3]. For the purposes of this project, it seems that the sentence-leve approach is the most suitable, as dialogues comprise of mostly single-sentence lines (rarely more than three sentences per line) and emotional payload may change as the dialogue progresses.

Traditionally sentiment analysis is performed by various classification mathods (usually supervised learning). Naive Bayes classifier and support vector machines were proven to yield pretty acurrate results (over 80% accuracy in general) [8]. The major constrait of those methods is that their quality is tightly linked with the quality of the lexicon used and size and quality of training datasets.

One state-of-the-art solution to emotion analysis problem has recently became publicly available. Created by IBM, Watson Tone Analyzer is a tool capable of accurate emotion labeling (fear, joy, anger, etc.) as well as tone labeling (analytical, confident, tantative, etc.). The system is based on the *Big Five personality traits* model [2]. Thanks to this approach, Watson is capable of much more detailed analysis of natural language than most other tools.

2.4.2 Emotions

To fully understand how to approach the problem of animation generation from emotional natural language it is important to look at how can emotions be understood and represented in computational terms. Emotions are complex and subjective in nature, so it is important t deconstruct emotions into something that can be worked with. A traditional approach to this problem is using the six basic emotion model. Specified by Paul Ekman and Wallace V. Friesen, who by studying nativa papua-new guinean tribes have discovered six universally recognized and understood emotions: [1]

- Joy
- Anger
- Surprise
- Disgust
- Fear
- Sadness

Although various research proposes changes to the six basic emotions model (such as reducing it to just four basic emotions [5]), this model is widely accepted and used in literature and software (such as IBM Watson tone analyzer, or various lexicons such as the NRC Word-Emotion Association Lexicon [1].

Psychologists theorize that the basic emotions can be used as building blocks to achieve more complex emotions. The specific emotion can be described as a mixture of basic emotions and their intensities. Robert Plutchik, a supporter of that theory, has represented emotions on a wheel-like diagram (2.1) that depicts how two basic emotions combine to create a more complex emotion [1].

[1]http://www.saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

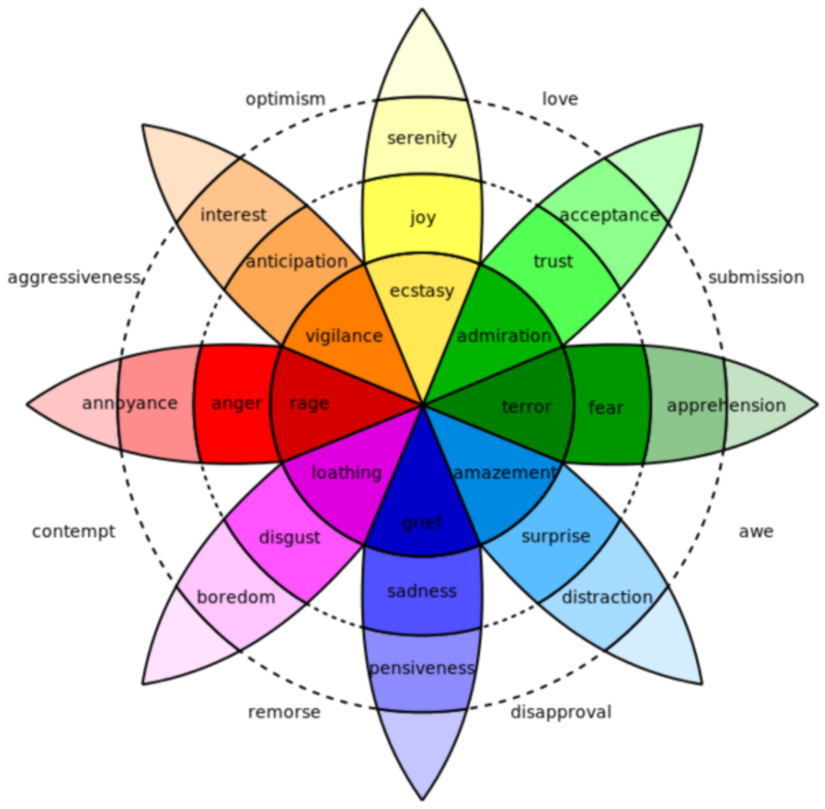


Figure 2.1: Plutchik's wheel

It seems that it is enough to extract the very basic emotions from the text, as more complex emotions can be understood as combinations of those basic ones.

2.5 Animation Software

The output scene must be created in some software able to play and use the scene. Developing a tool for this would be too time consuming - such software is not a small undertaking and with the time constraints this would result in a very rudimentary framework that would be very limiting. Therefore a choice must be made between existing frameworks.

The animation software must satisfy the following requirements:

- Flexibility and editability - As the scenes created by the generator will not be perfect, the software must provide powerful animation editing features.
- Exporting - It is desirable for the animation to be able to be exported into variety of different formats that can be used by other frameworks useful within game development.
- Availability - As the proposed tool is developed with small developers in mind, it is most desirable for the tool to be available without any unnecessary fees or licensing. The proposed tool shoul also provide help for people unfamiliar with animating, who would be unwilling to pay additional fees for something they are not familiar with.

- Familiarity - Although most animation software is based on similar concepts, due to time constraints my previous experience with the software is also important.

2.5.1 Maya

Maya is a 3D computer animation software developed by Autodesk. It supports modeling, rendering, simulation, texturing, and animation. This software is an industry standard and has been used for such projects as the Halo franchise. It supports all the animation editing and exporting features needed for this project, however it comes with a pretty harsh pricetag of $180 per month [2]; a price which would make the potential reach of the proposed tool much smaller. Moreover, although I am familiar with animation concepts, I am not familiar with Maya.

2.5.2 Blender

Blender is an open source animation software. Similarly to Maya, it supports modeling, rendering, simulation, texturing and animation. Due to its open source nature the software may be less usable or stable at times however it is still very powerful and recognized within the industry. Blender's rendering engines are not as sophisticated as those of Maya. It supports exporting the animations to Collada, Alembic, 3D Studio, FBX, Motion Capture (.bvh), Wavefront, X3D and Stl file formats. This means that the final animation could be exported and used by pretty much any other tool. Blender is completely free to use and available to anyone. I am familiar with the tool

2.5.3 Unreal Engine

Unreal Engine is by far the most popular and most powerful publicly accessible game engine. Since the proposed tool would find most use in games it would make sense to create the scenes directly in a game engine. This approach however has some disadvantages - animation editing features in game engines are much more limited than those of a software dedicated to creating animation. Moreover, any game engine will not support the same exporting features (however, since the animation would already be in the engine, the need for exporting is arguable). Unreal engine is free to use, however Epic Games will seize a portion of income generated by a product developed with Unreal Engine. [3]

2.5.4 Unity 3D

Unity 3D is the most popular freely available engine after Unreal. It suffers from the same drawbacks regarding animation editing features and is in general less stable and sophisticated. The only reason why Unity would be more suitable than Unreal for this project is my familiarity with the Unity 3D framework.

2.5.5 Final Choice

Upon taking a closer look on the available software, I conclude that Blender is the most suitable tool for the task as it satisfies the requirements best. It provides all the necessary editing exporting and editing features, is free and easily available and I already have experience with using it.

2.6 Motion Database

The dialogue scenes in this project are going to be assembled from existing short motion capture clips. While a potential user of the tool proposed by this paper (a game studio) would most

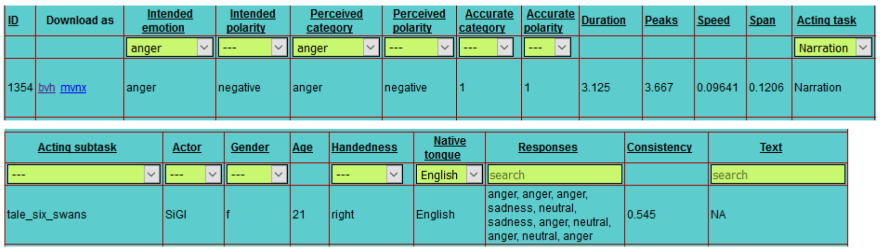
[2]`https://www.autodesk.com/products/maya/subscribe`
[3]`https://www.unrealengine.com/en-US/faq`

likely possess resources to create their own motion capture clips and be in posession of legacy motion capture data recorded for past projects, I must resort to use other resources. There exists a multitude of motion capture data freely available online[4]. Among them, the most famous is the Carneigie Mellon University's CMU Graphics Motion Capture Database[5]. It contains thousands of recordings of people walking, dancing, running, performing everyday activities, playing sports and performing gestures.

Only a small portion of that database would be useful to this project. There is however another database that focuses on matters more aligned with this project - the Emotional Body Motion Database [6]. The database was created by the Max Planck Institute for Biological Cybernetics and features solely recordings of people performing gestures associated with some emotion. The recordings feature all the basic emotions listed earlier as well as other emotions such as pride, relief, shame and more. There are three main types of motion capture recordings in this database:

- Narration - Actors were recorded while reading a story - their gestures were captured while narrating a part of the story that emphasizes some emotion.
- Nonverbal - Actors were recorded performing a gesture associated with some emotion in a nonverbal setting.
- Sentence - Actor were recorder while performing gestures when talking.

The database contains over 1400 clips, each labeled with what emotion it is supposed to represent, and what emotion would an observer assoiate the recording with [11] [12]. An example entry in the EBMD can be seen in 2.2.



ID	Download as	Intended emotion	Intended polarity	Perceived category	Perceived polarity	Accurate category	Accurate polarity	Duration	Peaks	Speed	Span	Acting task
		anger	---	anger	---	---	---					Narration
1354	bvh mvnx	anger	negative	anger	negative	1	1	3.125	3.667	0.09641	0.1206	Narration

Acting subtask	Actor	Gender	Age	Handedness	Native tongue	Responses	Consistency	Text
---	---	---		---	English	search		search
tale_six_swans	SiGI	f	21	right	English	anger, anger, anger, sadness, neutral, sadness, anger, neutral, anger, neutral, anger	0.545	NA

Figure 2.2: An example e

[4]http://jeroenvanboxtel.com/MocapDatabases.html
[5]http://mocap.cs.cmu.edu/
[6]http://ebmdb.tuebingen.mpg.de/index.php

Chapter 3

Analysis

The project methodology, development tools and technologies and project risk assessment are defined in this following chapter.

3.1 Methodology

The activities required in order to develop the project are listed below:

- Develop a project plan.
- Review literature. Review existing software, learn about other approaches, analyze existing research.
- Find and assemble a set of relevant motion capture animated clips.
- Prototype a module that uses processes the script and outputs a structured file that can be used to generate the animated scene.
- Tag and classify animation clips. Categorize clips by emotion, length, etc.
- Prototype a Blender extension that uses the created structured files to output an animated scene.
- Iterate on the existing software adding new features.

My goal is to first build a very minimal prototype of the program. The prototype will focus on emotions, work with a small amount of animations and keep the characters gesturing, but not moving or performing other actions throughout the scene. Afterwards I will focus my effort on making the outputted scenes more realistic and intricate.

3.2 Technologies and Resources

The first module of the project focuses on NLP. This module should be extract emotions and actions from the script. The most important tool used will be IBM Watson Tone Analyzer. If that tool turns out to be for any reason ineffective or imperfect, a customary tool (naive bayes classifier or a keyword classifier) can be built for that task. Actions can be extracted using a variety of information extraction software such as MITIE or Ollie.

The motion capture clips will come from two sources - the Carneige Mellon University motion capture database and mocapdata.com. I will hand-pick relevant animatio clips and preprocess them so that they can all be easily used with a character model. I will create a SQLite database that will store metadata about the animations (associated emotions, length, etc).

The last part of the project - assembling the final animated scene - will be done using Blender; an open source 3D modelling and animation software. The first module will create a json file that specifies a sequence of dialogue lines and animation clips accompanying them. A Blender extension will read that file and import necessary character models and animations to create a fully editable animated scene.

3.3 Risk Analysis

Risk	Mitigation	Level
Time delays caused by workload/illness	Follow the project plan closely. Focus on developing a minimum working prototype first.	High
Natural Language Approaches are too inaccurate	Use more structured software. Try to find other approaches to this problem. If there seems to be no solution, I can use the findings and existing software to argue that the technology has not yet reached a level that would make this project viable.	Low
Motions obtained from the EBMD are not expressive and unambiguous enough to create realistic and convincing scenes	Even with bad quality animations it should be possible to determine whether this project has future potential, given a better motion capture database is used.	Moderate

Table 3.1: Risk Assessment

Chapter 4

Requirements Specification

This section describes functional and non-functional requirements of the proposed software. The software must be developed accordingly to the requirements and must satisfy all of them in order to be truly successful.

4.1 Functional Requirements

1. **Analysis of a semi-structured script** - The software takes a semi structured script as input. The structure of a script must resemble a structure of a movie script. The script provides 2 kinds of information - characters involved and lines of dialogue spoken by them. The software must be able to extract emotions from the dialogue lines.

2. **Store motion capture data with regards to emotions** - The software must store and tag the motion capture clips with relevant metadata about what kind of emotions they represent. The database must be easily and quickly searchable.

3. **Find relevant animation clips** - The software must take information about character's emotions and actions and choose animation clip that best represent's characters behaviour. The chosen clips must reflext characters emotions, but also need to be of correct length to match the speed of the speech, as well as not repeat too often. The system must be compatible with Emotional Body Motion Database published by Max Planck Research Institute.

4. **Assemble the final scene** - The software must be able to output the final animated dialogue scene. The outputted scene must be fully editable, enabling various adjustments before rendering. The final scene must also be exportable to other formats so it can be used with game engines or other editing software.

4.2 Non-Functional Requirements

- The user should be able to use the software with a custom motion-capture database (Usability).

- The user should be able to assign differenct character models to different characters (Usability).

- The user should be able to fully customize and edit the final scene (Usability).

- The software is designed to automatically create big amounts of animated scenes. The time of generating a scene is not a high priority, but it must be reasonable (Peformance).

- The software must support common animation file formats (fbx, bvh) (Portability).
- The software must be modular enough so that different parts of it can be replaced with ease .It mustbe possible to replace emotion analysis software and 3D animation editing software with other software (Portability).
- The output animation must be perceived well by the audience and judged to be acceptable for use in a game given minor adjustments (Quality).

Chapter 5

Design And Architecture

This chapter describes the design of the system. This includes both the underlying decisions of the system as well as the user interface design. The chapter also presents on a more detailed view of the system's architecture.

5.1 System Design

The system has to work in two major steps. The first step is to take semi-structured natural language script and analyze it. This means that the system must analyze each dialogue line and infer some emotional values from them. Using those emotional values and the length of the dialogue line, the system must find best matching animation clips from the motion capture database. The results of this step is a file that specifies which characters say which lines of dialogue and it also specifies which animated clips accompany those dialogue lines.

After this step is completed, the file must be intepreted into a final scene. The importer must be able to read the file, import required models and animations and generate the final scene. As every animation software and game engine is a little different, each of them would need a custom importer. Those would be very similar in principle, but differ slightly because of the implementation of given software. For my project I have created the importer for Blender. This is because Blender is open source and available to anyone, as well as I am the most familiar with this software.

5.2 Emotions

For my project I have decided to only use the following emotions: joy, fear, disgust, anger and sadness. Tradinionally, surprise is also part of the six basic emotions model by Paul Ekman and Wallace V. Friesen. However, IBM Watson Tone Analyzer, that I am using for sentiment analysis, is unable to detect surprise in the text (more on that in section 5.5) - therefore I had to abstain from using this emotion in this project. The emotions are expressed as decimal numbers between 0 and 1 - this identifies whether a given emotion is present in a specific motion clip or text and how intense that emotion is. It also allows to create a mixture of emotions in order to represent more complex emotions.

5.3 Architecture

The architecture of the system is essentially a pipeline. The modules process resources and pass them onto the next module. They can be completely unaware of each other. This allows for a lot of flexibility and helps achieve some of the requirements. The emotion analysis API can be replaced with a different one, the user can use a custom motion capture database, and a different importer can be created if the user wishes to use software other than blender.

There are three main modules:

- Text Analysis
- Animation Clip Matcher
- Animation Generator

The Text Analysis module parses the text and analyses emotion using some API. It takes a script as an input and passess the parsed and analyzed text to the Animation Clip Matcher.

The Animation Clip Matcher uses the motion capture database to find the best animation clips to accompany the speech. The Animation Clip Matcher must take into account the emotion analysis of the text and find animations with similar emotional score. Another important constraint is the time of the animation. The animated clips have different lengths and a chosen clip must not be significantly longer than the spoken/read text. In case the animated clip is shorter, the Matcher must choose more than one matching clip. The Matcher outputs a JSON file which specifies all dialogue lines; it states which characters perform which emotional gesture actions and where is the animation file located.

The Animation Generator reads the JSON file and imports all needed animations. The user is now able to assign character model for each character and run the generator. The generator will assemble the animations in correct order, focus the camera on a currently speaking character and add subtitles. The output is an animated scene that can be either manually edited, exported to a file or rendered.

The full diagram of the system can be seen in figure 5.1.

Python programming language is used to implement the modules. This is mainly due to the fact that Blender only supports extensions created using Python. For modules that do not rely on Blender, Python was chosen due to high development speed.

5.4 Input

The input of the system looks as follows:

The input is a file that represents a dialogue. Each character that participates in the scene must ble clearly stated. Each dialogue line must have a dialog line must have a character clearly associated with it. Character names are specified after five tabs. Dialogue Lines are specified after three tabs. Each file must end with "ENDSCRIPT" with no indentation.

I used this format as this format is often used to represent movie scripts. One can find hundreds of scripts saved in this format on The Internet Movie Script Database (`www.imsdb.com`). The example input can be seen in figure 5.2.

5.5 Emotion Analysis

As aforementioned, there are many ways to perform emotion analysis of the text. I have chosen to use IBM Watson Tone analyzer for this task as it offers a state-of-the-art service accessible for prototyping (a lot of tools offering high quality emotional analysis are bult for commercial purposes and not available without paying high fees).

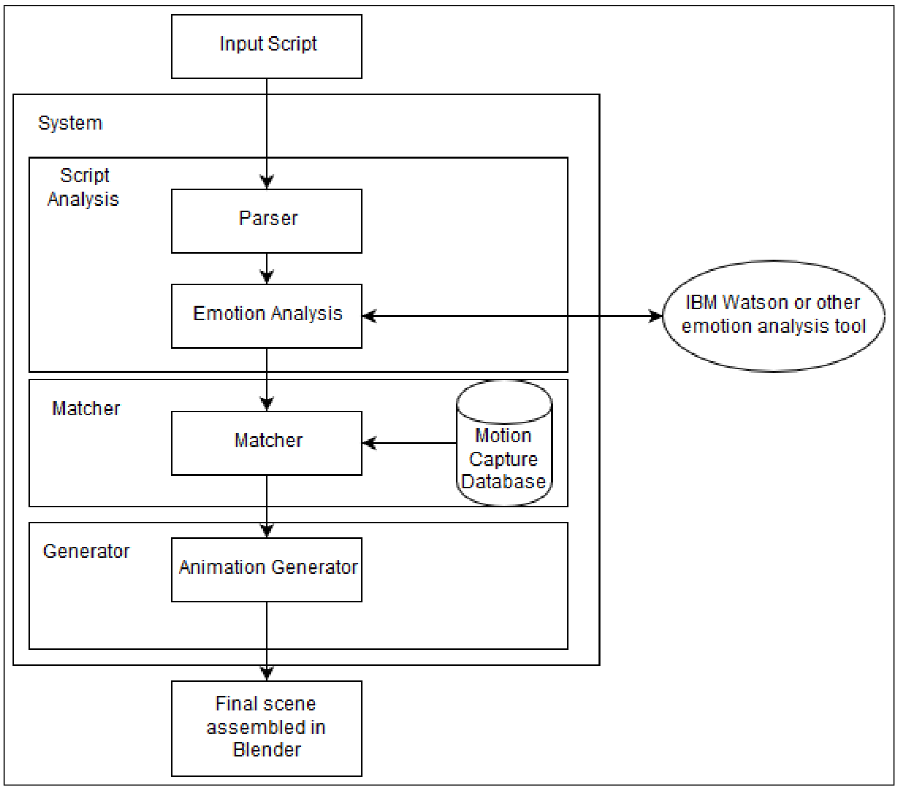
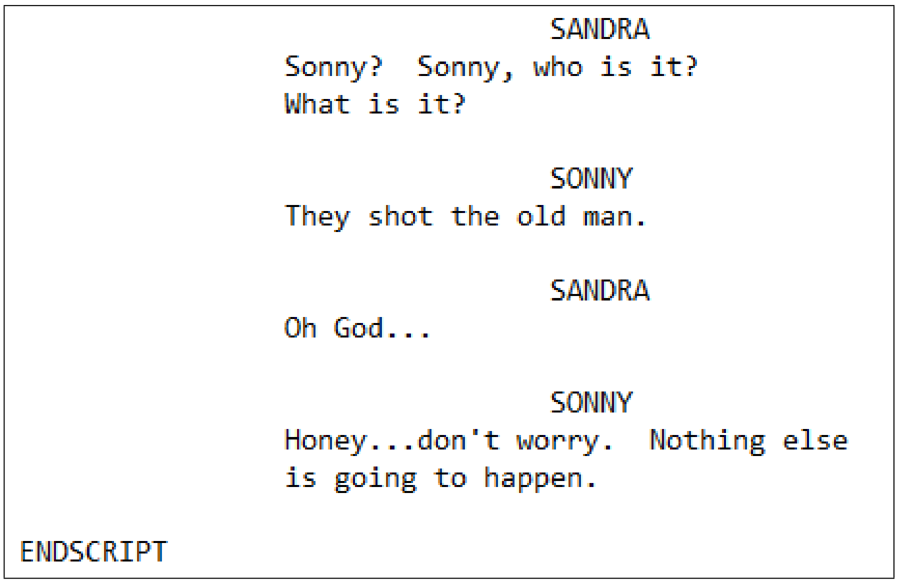


Figure 5.1: The pipeline architecture of the system



```
                        SANDRA
          Sonny?  Sonny, who is it?
          What is it?

                        SONNY
          They shot the old man.

                        SANDRA
          Oh God...

                        SONNY
          Honey...don't worry.  Nothing else
          is going to happen.

ENDSCRIPT
```

Figure 5.2: An example of system's input

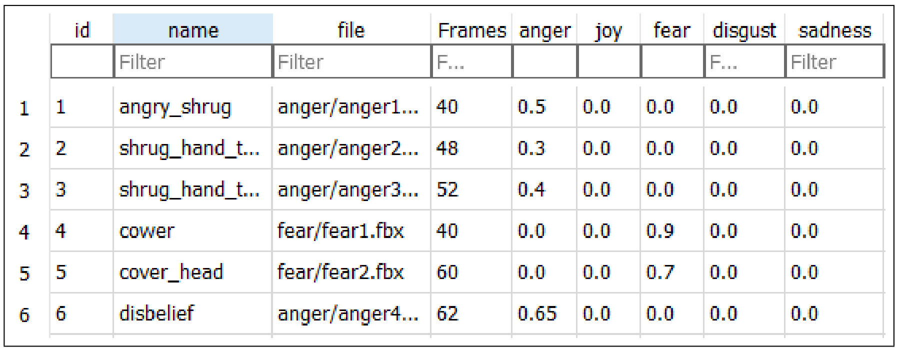
The system takes each dialogue line and sends to IBM Watson Tone Analyzer for analysis. Watson's REST API is used to accomplish that. Watson returns all the emotions with values between 0 and 1 found in the text. Each dialogue line now has emotional values assigned to it.

5.6 Database

The motion capture database consists of two main parts: the files that contain the animated clips and a database that holds the metadata about the animations.

Most animation come from the Max Planck's Research Institute Emotional Body Motion Database. This database is a set of motion capture clips performed by actors. The clips are supposed to represent emotions in various settings. Most of these animations need minor changes in order to be compatible with my system (e.g. a lot of clips were recorded while the actor was sitting, so the leg keyframes must be removed to make the model stand). The animations are exported to FBX files (proprietary file format owned by Autodesk, widely supported by many different frameworks) one animated clip per file.

The database is implemented using SQLite. It is the perfect tool for this task as the database needs to be simple, relatively small and easily searchable. The animation metadata is held in a table that look like this FIGURE. The emotional values of each animated clip need to be manually adjusted. When all values are set to zero, it means that the clip carries no emotional impact (neutral). An example of a few database records can be seen in figure 5.3.



	id	name	file	Frames	anger	joy	fear	disgust	sadness
		Filter	Filter	F...				F...	Filter
1	1	angry_shrug	anger/anger1...	40	0.5	0.0	0.0	0.0	0.0
2	2	shrug_hand_t...	anger/anger2...	48	0.3	0.0	0.0	0.0	0.0
3	3	shrug_hand_t...	anger/anger3...	52	0.4	0.0	0.0	0.0	0.0
4	4	cower	fear/fear1.fbx	40	0.0	0.0	0.9	0.0	0.0
5	5	cover_head	fear/fear2.fbx	60	0.0	0.0	0.7	0.0	0.0
6	6	disbelief	anger/anger4...	62	0.65	0.0	0.0	0.0	0.0

Figure 5.3: Database of animation clips

5.7 Matcher

does some stuff. explain matching algorithm here.

5.8 Animation Generator (Importer)

As a forementioned, in this project I have used Blender for creating the animation. The generator is a Blender addon script. The generator first reads the JSON file prepared by previous modules and loads required animation clips from files. Each action is assigned to a corresponding character model and then pushed on a separate NLA strip (figure 5.4). Subtitles are added as Video Editor Sequence Strips spanning between frames that correspond to the animation (figure 5.5). The camera is animated to focus on a character that is currently talking.

The final animation can be edited and adjusted in Blender (figure 5.6) or rendered (figure 5.7).

5.9 User Interface

The Animation Generator module is the only module that features a Graphical UI. The module is embedded into Blender and uses extends Blender's interface. The UI is minimalistic and simple to use. It requires the user to specify an input file, as well as animation directory. Upon execution, the extension will pre-load the animations and determine characters involved in the scene. Now it is up to the user to specify which model is supposed to be used for each character. Upon

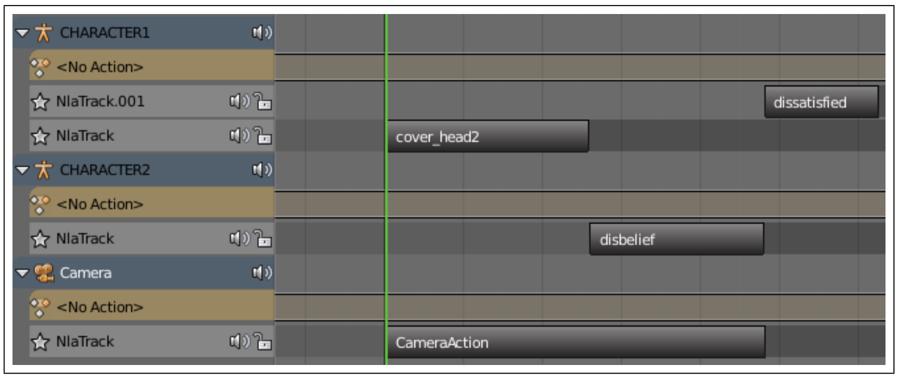


Figure 5.4: NLA strips of the final animation

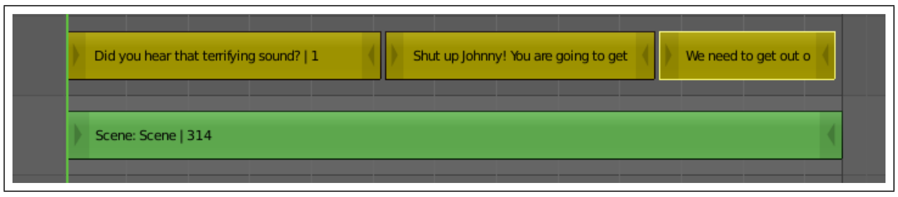

Did you hear that terrifying sound? | 1
Shut up Johnny! You are going to get
We need to get out o
Scene: Scene | 314

Figure 5.5: Video strips featuring subtitles

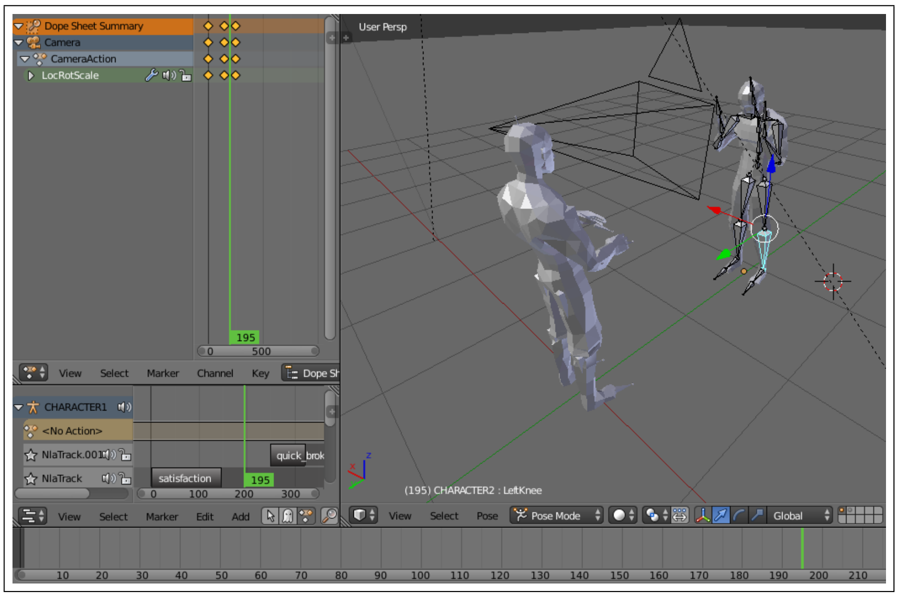

Dope Sheet Summary
Camera
CameraAction
LocRotScale
User Persp
195
View Select Marker Channel Key Dope Sh
CHARACTER1
<No Action>
NlaTrack.001
NlaTrack
quick_brok
satisfaction
(195) CHARACTER2 : LeftKnee
View Select Marker Edit Add
View Select Pose Pose Mode Global

Figure 5.6: Editing the final animation in Blender

pressing finalize, the animation will be created. From now on The user can continue to use Blender normally.

The JSON file must be created first outside of blender by running a script. This is because the modules need to be kept as separate as possible so that importers can be easily implemented for other platforms.

They key aspect of the interface is its simplicity as it allows users to generate relatively complicated animation sequences by essentially using two buttons, with no knowledge about animation and little Blender expertise required. The UI also provides flexibility as the user can decide upon which character models to use. The final UI is presented in FIGURE.

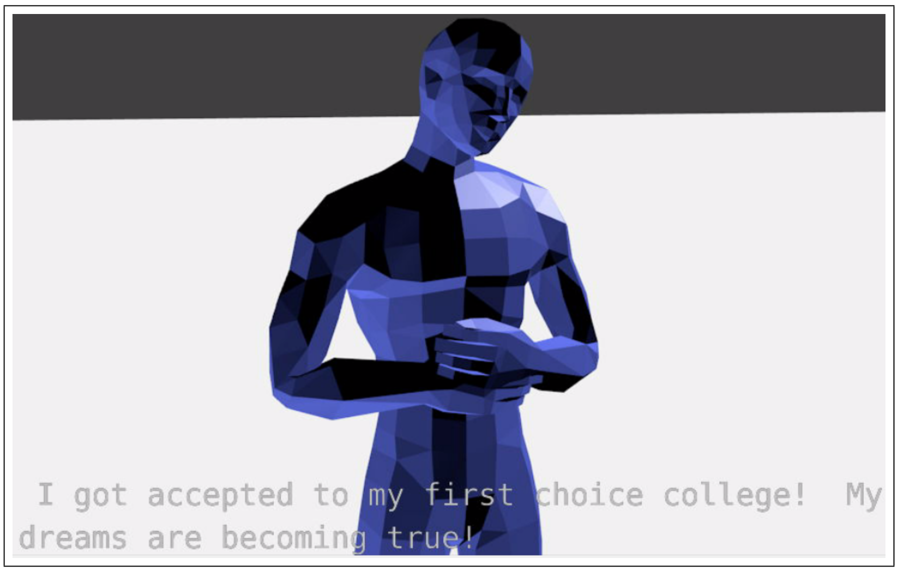

I got accepted to my first choice college! My dreams are becoming true!

Figure 5.7: Rendered final animation

Bibliography

[1] Burton, N. (2016). What are basic emotions? `https://www.psychologytoday.com/blog/hide-and-seek/201601/what-are-basic-emotions`.

[2] Chang, L. (2015). IbmâĂŹs watson may now be able to tell how snarky your email is with its new tone analzyer. `https://www.digitaltrends.com/cool-tech/watsons-new-tone-analyzer-can-tell-if-your-email-is-too-snarky/`.

[3] Feldman, R. (2013). Techniques and applications for sentiment analysi. *Communications of the ACM*, 56:82–89.

[4] Fenlon, W. (2016). Most of the witcher 3's dialogue scenes were animated by an algorithm.

[5] Jack, R. E., Garrod, O. G., and Schyns, P. G. (2014). Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. Technical report, University of Glasgow.

[6] Levison, L. (1991). Action composition for the animation of natural language instructions. Technical report, Universirt of Pennsylvania.

[7] Oshita, M. (2009). Generating animation from natural language texts and framework of motion database. Technical report, Kyushu Institute of Technology.

[8] Rohit, W. and Jagdale, R. (2018). An approach to sentiment analysis. Technical report, Department of CS & IT, Dr. BAMU, Aurangabad, Maharashtra, India.

[9] Sumi, K. and Nagata, M. (2006). Animated storytelling system via text. Technical report, Proceedings of the International Conference on Advances in Computer Entertainment Technology.

[10] Tominski, P. (2016). Behind the scenes of cinematic dialogues in 'the witcher 3: Wild hunt. `http://www.gdcvault.com/play/1022988/Behind-the-Scenes-of-Cinematic`.

[11] Volkova, E., Mohler, B., de la Rossa, J., and Bulthoff, S. (2014a). The mpi database of emo- tional body expressions common for narrative scenarios. Technical report, Max Planck Research Institute.

[12] Volkova, E., Mohler, B., Tesch, T., and Bulthoff, S. (2014b). Emotion categorisation of body expressions in narrative scenarios frontiers in psychology. Technical report, Max Planck Research Institute.