**How to build a Character Animation System**

1. **Introduction/Background**

Animating virtual humans is a complex task. Many diﬀerent aspects of human behavior need to be modeled in order to generate a convincing result. The behavior and appearance of a virtual character needs to be recognizably human in expression, although photorealism is not necessary. People are adept at recognizing movement and human-like behavior, so the actions and appearance of a virtual character must match the expectations of the human viewer. This means that not only must the character’s movements be natural, but they must be contextually appropriate, such as responding with the appropriate reaction and in the proper time frame to stimuli. Research has been done on various aspects of character animation, such as locomotion and facial animation. However, the integration of all these aspects leads to complexities. For example, coordinating locomotion with path ﬁnding, or coordinating reaching with gazing. At ﬁrst glance, it appears that modeling an entire animated character can be achieved by combining individual areas, and then reassembling the ﬁnal character as a combination of each individual part. For example, locomotion can be combined with a lip sync animation. This combination works since there is little relationship between locomotion and the movement of a character’s lips and face. Serious problems arrive when the areas overlap and directly or indirectly impact each other. For example, performing a manipulation with your hands while simultaneously looking at another object in the virtual world. The looking behavior might engage parts of the character’s body that disrupt the manipulation. Thus, although manipulation and gazing are distinct problem areas in animation research, they can interact with each other in unexpected ways.

**1.2. Problem Statement**

This project will address the challenge of simulating the different aspects of human behaviour and the difficulties that arise as we try to integrate all these aspects. This is what we need if we are to synthesize a highly realistic, interactive character.

Many game engines provide robust solutions to many real time simulation problems such as mesh rendering, lighting, particle eﬀects and so forth. However, game engines generally do not handle complex character animation. They often provide a general framework for replaying animations on a hierarchical skeleton, as well as a providing mechanism for blending between animations or looping an animation. However, the intricate and speciﬁc motion commonly associated with humans must be constructed by the game engine programmer and designer.

**1.3. Aim and Objectives**

**1.3.1. Main objective.**

The main goal of this project is to develop a character animation system allows the realization of common behaviors that are used in real time games and simulations. These behaviors include: synthesizing speech, responding to speech, moving, touching, grabbing, gesturing, and gazing, breathing, emotional expression and other non-verbal behavior.

**1.3.2. Specific Objectives**

* To analyze the existing character animation systems and identify the weaknesses thereof.
* To design a character animation system that addresses most of these weaknesses.
* To implement and document the improved system.
* To test and validate the improved character animation system.

**1.4. Scope**

Scope

**1.5. Literature review**

Literature review

**1.6. Methodology**

Basing on locomotion, the System can be implemented and experimented with two different locomotion systems; a semi-procedural system based and an example-based system. The semi-procedural locomotion algorithm uses two example motions; a forward walk and a strafe (sideways walk). The procedural nature of the algorithm allows the use of inverse kinematics to place the feet at the desired position as well as control the angle of the foot on uneven terrain. Since only two example motions are used algorithm, the motion can be parameterized along two axes, each representing either forward or sideways movement. Turning is controlled by setting step targets and orienting the body in line with the footsteps. The drawbacks to using such this semi-procedural system is the lack of ability of the algorithm to allow for differing styles of movement. Because the foot placement is established by using inverse kinematics, the nuances of the leg movement from the animation data are replaced with the results from the IK algorithm. The forward movement animations consist of two walking or running cycles, and the turning animations consist of a character turning around at various speeds; turning in place, turning in a tight circle while walking, turning in a tight circle while running and so forth. The size of the turning Building a Character Animation System 5 circle limits the amount that a character can turn while moving at a given velocity. The animations are parameterized in three dimensions; forward velocity, turning angle, and sideways velocity. Thus, it is possible to emulate any number of foot positions through a combination the various parametric animations. The parameter space of these dimensions can be modeled using tetrahedrons. The parameter values are automatically extracted from the example motions, such as determining the average forward velocity of a motion. We have found that the example-based animation produces a more realistic result, although this method can be susceptible to foot skating if the motion data is not consistent.

**1.7. References**

1. Thiebaux, M., Lance, B., Marsella, S.: Real-time expressive gaze animation for virtual humans. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1. pp. 321–328. AAMAS ’09, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2009), http://dl.acm.org/citation.cfm?id=1558013.1558057

2. Johansen, R.S.: Automated Semi-Procedural Animation for Character Locomotion. Master’s thesis, Aarhus University, the Netherlands (2009)