Phenotype-Based Figure Generation and Animation with Props for Human Terrain

CIS 499 SENIOR PROJECT DESIGN DOCUMENT

Grace Fong
Advisor: Dr. Norm Badler
University of Pennsylvania

PROJECT ABSTRACT

A substantial demand exists for anatomically accurate, realistic, character models. However, hiring actors and performing body scans costs both money and time. Insofar, crowd software has only focused on animating the figure. However, in real life, crowds use props. A three-hundred person office will have people at computers, chewing pencils, or talking on cell phones. Many character creation software packages exist. However, this program differs because it creates models based on body phenotype and artistic canon. The same user-defined muscle and fat values used during modeling also determine animation by linking to tagged databases of animations and props. The resulting figures can then be imported into other programs for further stylization, animation, or simulation.

Project Blog: http://seniordesign-gfong.blogspot.com

1. INTRODUCTION

The package is a standalone program with a graphic user interface. Users adjust figures using muscle and fat as parameters. The result is an .OBJ model with a medium polygon count that can later be made more simple or complex depending on desired level of detail. Each figure possesses a similar .BVH skeleton that can be linked to a motion capture database which, in turn, is linked to a prop database. Appropriate motions and any associated props are selected based on user tags. These tags also define which animation matches to which body type. Because it uses highly compatible formats, users can import figures into third-party software for stylization or crowd simulation.

1.1. Significance of Problem or Production/Development Need

Human terrain and crowd simulation frequently used in the entertainment industry. When populating backgrounds of live-action movies, digital crowds can be a timely and cost-effective alternative to hiring extras. In the computer animation and video game industries, all landscapes and populations must be created digitally. Crowds are a necessary component in research fields such as evacuation simulations. However, every crowd which is simulated needs agents to populate it. This project aims to inexpensively fill this need for agents. Whether it is controlled by artist or script, the program will generate high-compatibility, anatomically-accurate, human terrains with user-defined physical parameters and animations.

1.2. Technology

For compatibility with the current crowd simulation software used by the computer graphics department at the University of Pennsylvania, the program will be written in C++. The figures' base models and interpolation curves will be constructed in Autodesk Maya. GUI development will be done in Qt and OpenGL. Gathering motion capture data will take advantage of the Vicon Blade system and Autodesk MotionBuilder, but any key frame animation will be done in Maya. For more direct compatibility with Maya, the program will be wrapped with Python and might be turned into a plug-in.

1.3. Design Goals

This tool minimizes artists' and researchers' work by generating anatomically accurate human while still allowing users control levels of variation.

1.3.1 Target Audience

This program targets users who need to efficiently generate human terrains. It minimizes work for artists who need to vary background figures for games and movies. It also targets crowd simulation researchers who need realistic agents for their projects but do not have the resources to hire artists.

1.3.2 User Goals and Objectives

The product is meant to be versatile enough for use in both artistic and research fields. A user with no technical background should be able generate a prop-populated, animated, human terrain entirely through the GUI. However, it should also be compatible enough that a script, such as a Python or MEL expression, can function as a "user." Using standard file types (OBJ, BVH) allows models to be used in a greater variety of simulation and modeling software.

1.3.3 Project Features and Functionality

The end product is a standalone application which generates figure models with a variety of phenotypes. Users, mostly artists, should be able to use a GUI of sliders and possibly numerical input (based on BMI calculations) to control phenotypes of resulting models. When the scene requires a large number of models, the same GUI can be used to specify a range for random values and make a random crowd based on how many agents the user needs.

The GUI also has a secondary animation-linking mode which allows artists to tag BVH files and link props to the animations. Users themselves do not attach the animations to the skeletons. The program decides which animations are appropriate to which body types based on the tags. It also automatically decides which props to include based on which animation is currently being used.

Also, programmers also should be able to automate “user” input with a script. It is also likely that the project may end up being a Maya plug-in which would greatly increase its usability with artists.

2. Prior Work

Few character creation programs take crowd simulation into consideration. For example, MakeHuman and Poser (SmithMicroSoftware) are considered the most anatomically accurate, but they are used for medical training and figurative art, respectively. Many massive, multiplayer, online games - such as World of Warcraft, Lineage, and Guild Wars - have some sort of 3-dimensional, avatar creator. Emote Games even sells "Participate" software which allows programmers to add avatar creation to their social networking sites. However, because the purpose of in-game, creators is to give the user a unique, customizable avatar experience, they limit themselves through high stylization, focus on clothing, and incompatibility with other software. The closest existing product is the Sims 3 game which allows users to adjust characters based on the same weight and fitness variables, while still being anatomically accurate within the style of the game. However, these models have very limited use - the Sims.

3. PROJECT DEVELOPMENT APPROACH

3.1. Algorithm Details

Male and female bodies differ in too many ways (ribcage-pelvis ratio, breasts) to use one mesh to capture both. Therefore, there will be a base model (an OBJ) for each sex. Using a mid-detail mesh allows users to increase or decrease level of detail on the model, suited to their purpose.

Nevertheless, in both situations, the overall superficial contour of the human body is defined by muscle and fat. The mesh will interpolate between wire deformers, placed at strategic points through knowledge of human anatomy. Each pair of deformers represents a muscle or fat pad, one wire for minimum mass, one for maximum mass. The deformers end up dividing into two sets: one for muscle and one for fat. Because fat develops and sits on top of muscle, fat deformers will be parented relative to the muscle deformers and move as muscle deformers move. The same theory can also be applied to other, more specific, features such as shoulder width, head size, breast size, etc.

Skin and eye color are adjusted believably lightening and darkening the base of a layered shader. Further layers adjust other variables - such as ruddiness for more reds in fleshy areas. A freckle layer and a body hair layer could procedurally generate markings based on which areas realistically receive more growth (first on the face, then limbs, then chest, etc).

Many crowd simulations require models with the same skeleton. If aspects like height, weight, and sex vary too greatly between many models using the same skeleton, the animation may become unbelievable. The program lets user import their own skeleton and automatically constrains the deformers based on what has been provided. On the other hand, if the user does not have a BVH skeleton they would like to use, the creator can make one for them by calculating joint placement relative to the mesh. The linker selects appropriate animations based on the mesh and skeleton.

Models can be exported as just skins (OBJs), or they can use the linker to connect to a database of BVH files. BVH files are tagged so that the linker knows which animations to use based on the model's anatomical aspects. For example "heavyweight walk" would only be used on characters with high obesity. Some actions are associated with OBJ props. For example "overweight man on cell phone" utilizes the "heavyweight walk" and parents a modeled cell phone to the skeleton. Animations and props should be user-created and user-tagged. Letting users add to the databases allows for greater variety of use. For example, a crowd for a movie set in 1910 uses different props and animations than one set in 2010. However, the user must ensure animations will be compatible with the previously selected skeletons. For example, the program will not check to make sure the number of joints from a motion capture system matches the number on the generator-created skeleton.

3.2. Target Platforms

3.2.1 Hardware

The program will be created on a PC. The resulting standalone should work for Windows, and possibly Apple, operating systems. The plug-in version should be compatible with any computer that runs Autodesk Maya.

3.2.2 Software

The software will be created in Microsoft Visual Studio on a machine running Windows XP. Therefore, the standalone will be first produced as an .EXE. An Apple version can be compiled later. Users should not need any specific software to run the standalone program. However, the Maya plug-in version requires Autodesk Maya.

4. WORK PLAN

4.1.1. Project Milestone Report (Alpha Version)

By the milestone (October 16, 2009), I will have the models completed. All wire deformers will be placed. I will also have the GUI completed.

4.1.2. Project Final Deliverables

The final deliverable should be a standalone application which produces OBJ models and BVHs. There will also be a small database of test animations (from motion capture) and props. Any other versions (Apple OS, Maya plug-in) will be tackled after the initial standalone is completed.

4.1.3 Project Timeline

- Finish additional research.
- Finish base meshes.
- Place curves relative to curves to base meshes.
- Design and code GUI.
- Paint skin textures.
- Create freckling/body hair algorithm.
- Create layered skin shader.
- Finish BVH uploading.
- Record motion capture data for animations.
- Finish modeling props.
- Finish linker between skeletons and tagged animations.
- Finish linker between tagged animations and props.
- Convert to plug-in and other formats.

4.1.4 Gant Chart

Note, dates represent week ending that day.

	9/26	10/3	10/10	10/17	10/24	10/31	11/7	11/14	11/21	11/28	12/5	12/12
Additional Research	■											
Base Meshes		■	■									
Curves			■	■								
GUI				■	■	■	■	■	■	■	■	
Skin Texture					■	■						
Freckling						■	■					
Skin Shader					■	■	■					
BVH Import/Export							■	■	■			
Motion Capture							■	■				
Link Animation								■	■			
Props									■	■		
Link Props										■	■	
Convert												■

5. REFERENCES

MakeHuman
Poser (SmithMicroSoftware)
World of Warcraft
Participate(Emote Games)
The Sims 3