Georgia Institute of Technology

School of Literature, Communication, and Culture

Computational Media

Designing a Character Avatar Model for the Mermaids MMO

12/15/2010

Samer Ead
Literature, Communication, Culture
Computational Media
Emergent Game Group
Mermaids

Celia Pearce
Assistant Professor
Literature, Communication, Culture
Computational Media
Emergent Game Group
Mermaids

Dr. **Shannon Dobranski**
Associate Director of Undergraduate Studies
Literature, Communication, Culture
Computational Media

ABSTRACT

This paper describes the technique formed for the creation of an efficient, simply rigged, customizable mermaid avatar model for the *'Mermaids'* massively multiplayer online game (MMOG). Our goal was to improve the in game customization of the player's 3D mermaid model, while maintaining rendering efficiency. We devised a procedure that utilizes the iterative nature of design without sacrificing the scientific and technical aspects of the project. Our procedure begins by a method known as "Partitioning" where we break down the model's body into distinct sub-models. During development, this partitioning allowed us to focus on smaller concise areas of interest, whereas during game-play this change granted the player greater strands of customization freedom. Since the model relied on a skeleton for its animations, it's partitioning required "Skeletal Reformations" to reassess the control scheme of the rig over the sub-models. In this method, individual sub-skeletons were designed to provide increased local control over their respective sub-models in contrast to the global control that the previous rig allowed. The sub-skeletons were then joined together forming a combined and complete skeletal rig for the mermaid model. We iterated through the previous methods refining their procedures in efforts of "Balancing Customizability with Efficiency", which in turn provided us with the results of our novel technique. Our technique utilizes innovative methods that localize skeletal control over respective sub-models in a novel way, which allows increased customizability with limited costs to efficiency.

INTRODUCTION

"Mermaids" is a massively multiplayer online game (MMOG) being developed by Georgia Tech's Emergent Game Group {EGG} directed by Digital Media faculty member Celia Pearce, around the principles of utilizing large-scale emergent social behavior as a design material. In summary of the student-based development team's research mission, we work to develop specific games, techniques, and design features that promote large-scale emergent social behavior in multiplayer games. This paper focuses on a subgroup that researched an innovative design procedure to introduce greater in-game customizability features within the mermaid's model while maintaining efficiency.

Traditionally, game developers resorted to the cheapest and quickest known ways of creating 3D models that enabled basic player customizability through developer pre-configurations. Such models had low efficiency costs, and supplied all the necessary elements of aesthetic and customization features that the players expected. With low customer expectations the developers were able to get away with a limited number of low quality fake animations, as well as preconfigured customization features that restricted players to minimum capabilities of character personalization. Thus, a player was capable of choosing the kind and possibly the color of the hat they wish their character to wear, but they were not capable of determining the way the character wore it – for example soldiers tend to wear their formal hats in different slanted ways to portray certain honors, credentials, or affiliations. Our technique utilizes innovative methods that localize control in a novel way, which allows increased customizability and player freedom with limited costs to efficiency.

In traditional MMOGs, game developers allowed players' limited customization of their main game characters. Such features were available on two broad levels, at the beginning of the game and during game play. During the first level, players had to choose a main character model from a set of pre-designed models – determining their age, sex, race, and other features that the player could never change. Whereas during the second level, avatar customization was limited to predefined locations and utilized secondary models – changing clothing, weapons, and jewelry. Conversely, our technique allows players to change any part of their main character's model at anytime during the game, including both primary and secondary levels[1]. Though, since secondary models – such as hair, fins, and jewelry – are modeled and animated separately from the primary ones, our technique also grants the player control to maneuver the placement locations with minimal restrictions – for aesthetic reasons hair should to be placed on the head. Figure 1 is an example of a seemingly normal mermaid concept model that is ready to be placed within a gaming engine. Although the elements of Figure 1 are very similar to one that could be created by traditional methods of modeling, rigging, and animating, the sublevels of this model are in fact five distinct 3D

Figure 1: The concept image of a 3D model of the merman avatar. Concept by Rose Peng, and 3D model by Samer Ead

models. These fundamental differences in the design and the foundation of our technique have brought new ways to allow higher levels of interaction and personalization between players and their characters.

Our technique utilizes the standard character creation procedure based on the following methods: "Modeling, [2]" "Rigging, [3]" and "Animating, [4]" to provide levels of in-game customization that were relatively non-existent previously. This technique can be summarized within three methods: "Partitioning," "Skeletal Reformations," and "Balancing Customizability with Efficiency." If used properly, the combination places limitations on the models to preserve the dynamics of ordinary looking simulations while making future potential updates, upgrades, and additions much simpler. Thus, and in addition to the previous example, instead of having to change the entire model, or even requiring a back-end designer to change a partition of the mermaid, such capabilities can be granted to the user or player. The player would be able to select from premade models, not only to change secondary-level models, but also to make changes to the main top-level model. For example, a simple yet significant addition could be placing a shark's fin on the tail of the mermaid model, thus not only changing the entire outlook of the model itself, but also better reflecting the player's intended persona within the social structure of the game.

In conclusion, the goal of our research was to find the proper balance for our technique to provide the highest possible level of customization while maintaining the highest level of rendering efficiency. Our results proved that our methods provided increased customizability with maintained efficiency on various design levels. They also showed that the procedure helps programmers simulate the models during game-play with increased algorithmic control and organization.

RELATED WORK

Research from the modeling, rigging, and animation fields that combine into the complete design of a working model have proven vital for the compilation of our technique. Research on tools for beginner and casual designers surprisingly resulted in our deeper understanding of the fundamental requirements to build an efficient model without completely depending on intricate techniques.

Research on exploratory modeling, a modeling method for the sake of exploration and experimentation without a final goal in mind – usually the general approach of beginners and the approach of professionals during brainstorming phases – has brought about great techniques for developing easier but equally efficient tools. [8] Talton et.al. expanded on this method by employing user-generated content and surveys to build a greater library of systematically modeled designs that in turn could serve a wider range of users. In this project, we utilize their approach to the complexity of manual modeling, and carefully consider their results in building easier tools, in order to devise a relatively easier procedural technique that would enable us to achieve greater understanding and expertise in a shorter amount of time.

Figure 2: One skeleton template - with limited acceptable alterations - was used to fit and control all of the models. The Figure was taken from "Automatic Rigging for Animation Characters with 3D Silhouette." [5]

Many researchers have experimented with the generality of skeleton rigs, a practice that enables the use of the same skeletal rig for multiple 3D models of similar applicable nature – example illustrated in Figure 2. For example four legged animals (size into consideration) pretty much could all utilize the same animation rig. Both Baran et.al. and Pan et.al. provide automated solutions that aid the user in the construction of seemingly simple rigs [1, 5], though neither is capable of handling complex models, nor provide higher levels of control over their models to handle a high level of secondary customizability features[5]. We have been able to utilize this research to aid us in the simplification of the previously mentioned procedural technique over the multiple levels of building a model. This has resulted in the creation of the "Partitioning" method that ultimately provided complex models in plain, distinct forms that have direct structural relations between the models -to- the rigs -to- the animations.

On the other hand, the papers by Lee and Hodgins and their colleagues respectively [3, 7], provide great insight into how the model reacts in simulation of the boundaries of the skeleton rig, and its direct effects on the animation and deformation of the model throughout the procedure. Lee’s eloquent presentation of the direct relation between real human skeleton and the modeled skeletal rig [3] inspired experimentation with rigs that resemble real-life skeletons within the range of the model’s extent – specifically the tail of the mermaid with a skeletal rig that resembles a fish-tail skeleton. The direct results of the experiments provided us with the idea for the “Skeletal Reformations” method. This method was the key to maximize control over both primary and secondary models, while maintaining smooth animations, and algorithmic efficiency.

Controlling deformations to maintain efficiency was made possible by the cumulative knowledge thus achieved. But the unique results obtained by Park and Hodgins’ [6] experimental research on motion capture distinctively improved our understanding of the difficulty of the issue. If properly assessed with Lee et.al.’s methods, it would directly bring to attention ways of identifying weak spots, those with increased possibilities of deformations [3]. In general, models with the greatest complexity tend to be the most vulnerable, and the most deficient – i.e. the upper body portion of the mermaid, which is the human torso model. Even though the experiment was based on different fundamental modeling simulation and animation agents, the results indirectly motivated placing restrictions on simulated models to limit unreasonable shortcomings.

Though none of the other works singularly solved all the proposed problems within modeling, rigging, and animating, their combination, or more thoroughly the combination of their successful results provided insightful knowledge to conduct further experimentation. While automated techniques for all three processes have proven potential pros, their combined cons overwhelm the idea of a fully automated model. Thus, devising techniques to simplify the complexity of manual modeling labor prevails as the most logical solution.

Finally, none of the previous researches attract changes to models during real-time simulations in game-play. All their solutions rely on the models and rigs to be virtually static and edited outside the game. However, it's this paper's goal to address this problem, as it remains an open issue for designers to address.

METHODS

Based on the traditional processes of modeling, rigging, and animating, our technique entails three methods: "Partitioning" "Skeletal Reformations" and "Balancing Customizability with Efficiency" with each method uniquely corresponding to a distinct set of combinations and iterations of the processes. Provided with a high-level model of the avatar's biological compilation, the avatar was partitioned into a set of sub-models made of the most basic forms of coherent body systems – head, arm (though there are two of them), torso, and tail. While aesthetically easy to merge, the real challenge was creating a skeleton that is capable of providing all the different partitions with pleasing animations without distorting the remainder of the body. Thus, skeletal reformations were made for each sub-model based on its respective resemblance to a real-life skeleton, a process that enabled primary level customization – i.e. changing the tail with legs. Finally, in order to provide the most efficient secondary level customization – i.e.

wearing clothing – the rigs governing the singular sub-models had to be subjected to further testing and balancing in order to re-situate the skeletal joints in best possible algorithmic pattern.

The following subsections will delve into the fundamentals and details of each of the methods.

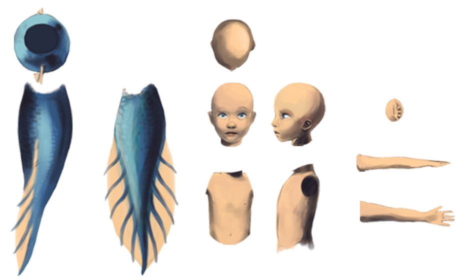


Figure 3: The partitions of avatar model, the tail, head, torso, and arm. Even though Figure 3 shows only the left arm's model perspectives, the right arm model is a duplicate, flipped horizontally. Concepts by Rose Peng

Partitioning

Partitioning is the method devised to split a model of high-level complexity into a combination of smaller models. The sub-models must maintain the attributes and the fidelity of the original model, while simplifying its makeup allowing users to make changes to selected sections without affecting others. Partitioning itself is conducted onto two different levels: the 3D model and the skeleton rig, which are directly correlated and dependent on each other to maintain efficiency and smooth animations.

3D Model partitioning splits the main 3D model into a number of distinct individual sub-models. In our experiment, Figure 3 shows how partitioning of the mermaid 3D model was split into 5 segments, where each inherited the attributes of the primary model – distinct skin, skeletal rig, and animations. With this novel perspective of the models, the designer is at liberty to introduce a number of interventions to update upgrade or even entirely change specific portions during game-play with minimal acceptable deformations. Traditionally, such minimal skin deformations would be easily covered up with spoof animations, or combinations of white light and other classic magic effects. Lately, MMOs utilized these cover up effects as leveling animations though no direct model changes necessarily occur. Conversely, our technique exploits the sub-models to allow the designer to incorporate realistic transition animations that increase levels of immersive content. For example, while a certain player chooses to train his or her character into a shark hunter, with larger muscles and thus larger arms. A traditional spoof animation would glow the thin arms with bright white light and magically turn them into muscular arms. On the other hand, a realistic transitional animation would be designed to pop the muscles out in a sarcastic manner before the model is restored to its new figure. These animations are made possible mainly due to their limited to no affects on other sub-models, which help maintaining minimal efficiency costs. The animations also increase player satisfaction, creating an entirely new dimension for the player's immersive experience in MMOs. Despite the fact that the newly formed sub-models are completely separate entities, they are still tied to together by their individual bonds to the combined skeleton.

While partitioning the 3D model was literal, the partitioning of the skeleton rig has both literal and metaphorical parts. Theoretically, the skeleton partitions fit the same five distinct sections as those of the 3D model partitions. Though technically, in order for the body to act as one, the sub-skeletons remain attached to each other via extremity joints. Thus, the sub-models are indirectly attached to each other through their bonds to their respective rigs. The actual method of achieving the partitioned sections of the rig is a bit tricky, but if done properly should prove to be the basis of all customizability features provided by our technique. Figure 4 shows the wireframes of the model and the rig of the mermaid's tail and torso attaching section. While the tail's 3D model had been bound to the tail rig, and the torso 3D model had been bound to the torso rig, the highest central static joint on the tail and the lowest central static joint on the torso skeletons are joined. This attachment places the tail rig below the hierarchy of the torso rig and through similar skeletal joints the sub-models are reattached to form the complete composition of the main character model. Seemingly complicated, this approach to the skeleton bonds grants us two distinct customization approaches. On one hand, the working skeleton can now be preserved along with all its animations, thus only having to replace the 3D model as discussed previously. On the other hand, the 3D model could be preserved, and changes would be made to designated portions of the skeleton thus changing the animations to be played during simulation – as long as the base shape and number of joints are similar the bonds are preserved. Combining the two previous approaches, both model and rig could be easily replaced during game play enabling the replacement of the entire mermaid's fish tail with octopus tentacles for example. This technique provides a massive bank of possible future innovations from such simple outcomes.

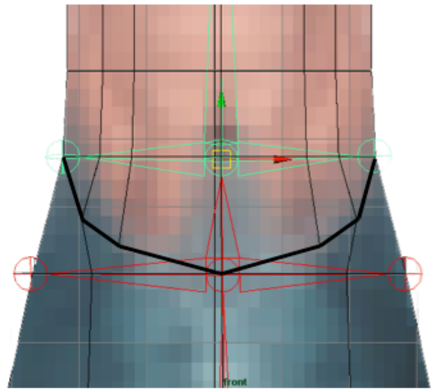


Figure 4: Image of the wireframe of the 3D model fitted to the concept art of the avatar along with the attachment section of the 2 partitions' skeletal rigs. Concept by Rose Peng, and 3D model by Samer Ead

Skeletal Reformations

Inspired by Lee's [3] research on the relation between the real human skeleton and human modeled skeletal rig, we decided to conduct our own experimentation on such relations. The hypothesis stated that rigs which resemble real-life skeletons, within an acceptable range of modeling limitations, would provide the most possible control over the skin, as well as would be the most efficient. In the method Skeletal Reformation, two different reformations to the sub-skeletons within the mermaid's rig were explored. Initially, a reformation of interest was the replication of a real human skeleton onto the mermaid's hand skeleton, a part of the arm sub-model, in order to increase the level hand gesture and grip animation algorithms. Later, the replication of a fish's tail skeleton onto the mermaid's tail skeleton was added to the experiment, in order to provide smoother tail animations, as well as better algorithms for tail customization.

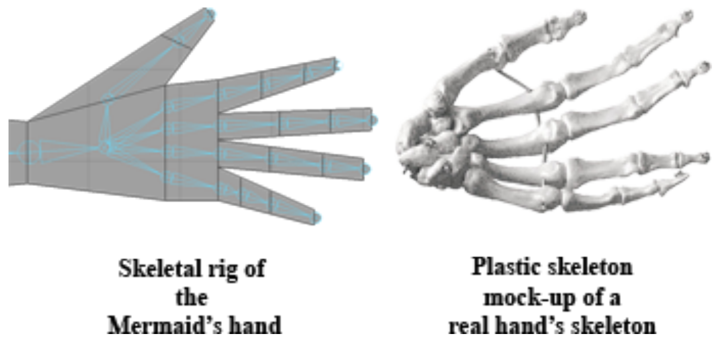


Figure 5: Image of a plastic mockup of a real human hand skeleton (Right), and of the wireframe of the skeleton of the Mermaid's hand (Left). Computer skeleton by SAmer Ead

To increase the level hand gesture and grip animation algorithms, the skeleton hand was reformed. Figure 5 compares the images of a virtual to a mockup skeleton of a real human's hand. The joints in real hands are situated in a very specific pattern to increase levels of gripping power and control. The weight and size of the hand are balanced to equate the force applied on the palm of the hand, which in turn frees the fingers to grip. Base on this information the avatar's hand was designed to adopt the number of joints and their placements on the skeleton's hand. The respective locations of the joints tipped the virtual weight of the hand to the location of the thumb, so the skeleton got out of sync with the algorithm controlling it, and the grip animations of the hand over secondary models were horrible. Balance in the virtual hand was restored by placing aligned joints down the upper half of the palm, a step that emphasizes greater weight and thus control from the palm on the hand. That step also required the reorganization of the hierarchy of all of the hand's joints.

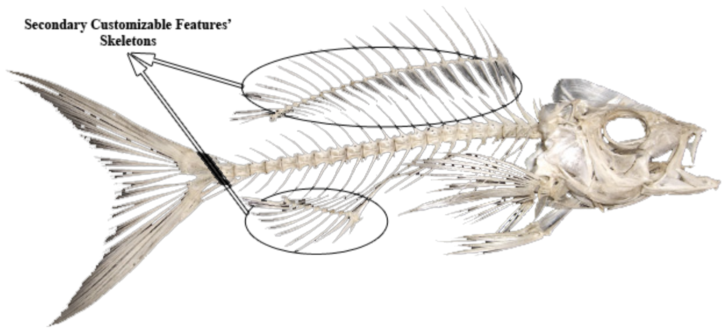


Figure 6(a): An image of a real fish's skeleton, along with real secondary skeleton that control fins, and other ornaments.

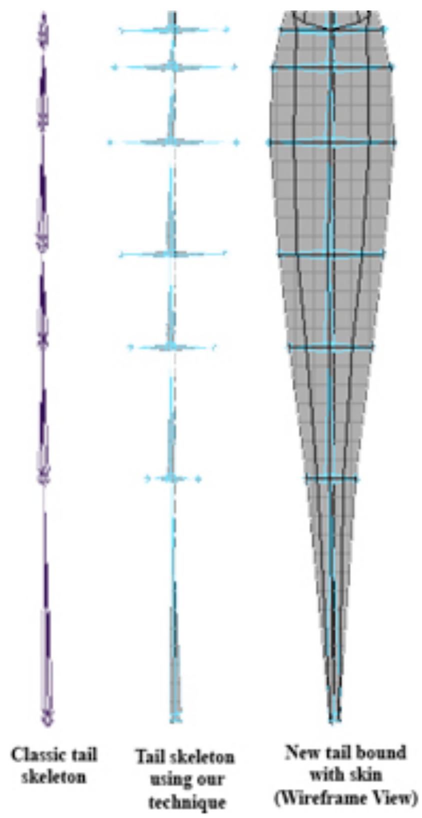
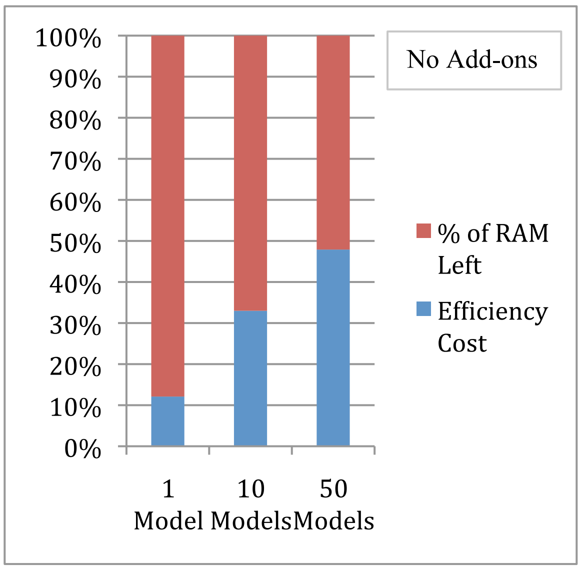


Figure 6(b): Traditional mermaid's tail skeleton (Right). Newly designed tail that is based on our technique (Center). Our design bound to the tail's model skin (Left). Computer skeletons by SAmer Ead

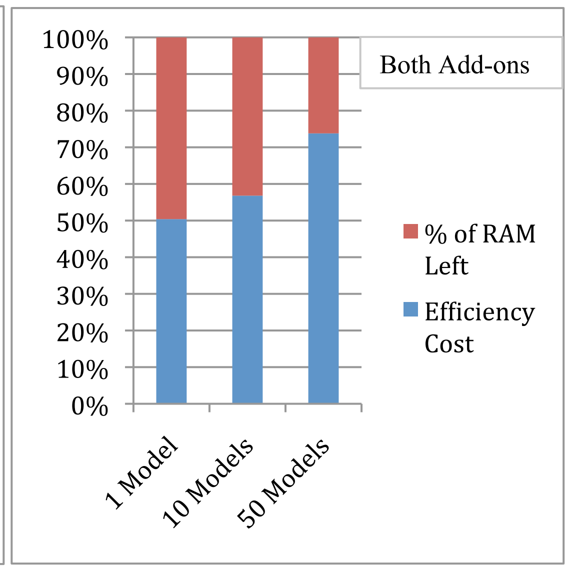
The second experiment focused on the replication of a fish's tail skeleton (Figure 6(a)) onto the mermaid's tail skeleton (Figure 6(b) – Left). The goal was to provide smoother tail animations, as well as better algorithms for tail customization. The process was initiated by careful studies of the skeletons of real fish tails, along with their influence, control, and limitations in real life. But based on the results of the first experiment it was apparent that real to virtual conversion might not be exact. Access to the traditional version of fish tails (Figure 6(b) – Right) provided a control specimen to the experiment that conveyed the least amount of acceptable control over the tail's skin model. Such classic, linear, tails are very poor at referencing the locations of secondary models that would be placed on top of the tail in a three-dimensional environment. Experiments on classic tail skeletons showed that secondary models are referenced with their respective location from the origin point of nearest joint, with limited distances from that joint, which constricts customization severely. Some of the most apparent constrictions are the acceptable locations of placement for secondary models have to be predefined. Those models have to also be static to minimal animations in order not to have rendering lapses, lapses that would throw the secondary models off of the main body creating great aesthetic deformations. Utilizing the results obtained from the studies on real fish tales, reformations to the classic mermaid tale were made to incorporate joints that reached the horizontal extremities of the tail. With the newly found tail skeleton, it became possible to devise an algorithm that triangulated the location of a secondary customization model that greatly improved rendering efficiency. The extremity joints also provided space for secondary models to attach themselves to the skeleton of the main body thus adding to the five sub-models, and inheriting all the attributes. Even thought the skeletal reformation experiments had both been successful on the micro level, further testing was necessary in order to confirm their validity on the macro level of a massively populated server.

Balancing Customizability with Efficiency

The technique has thus far been successful in increasing customizability using Partitioning and Skeleton Reformations, while successfully maintaining efficiency on the client scale. On the other hand, efficiency readings, shown in graph 2 in versus graph1, were extremely high considering a server scale of a massively multiplayer online game.



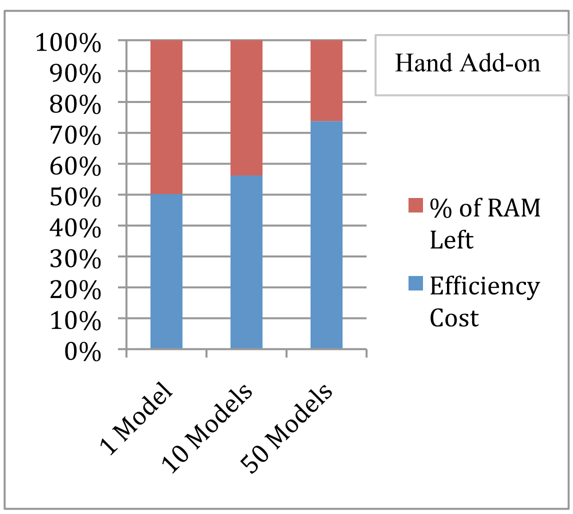
Graph 1: The efficiency readings of the server using the older traditional models.



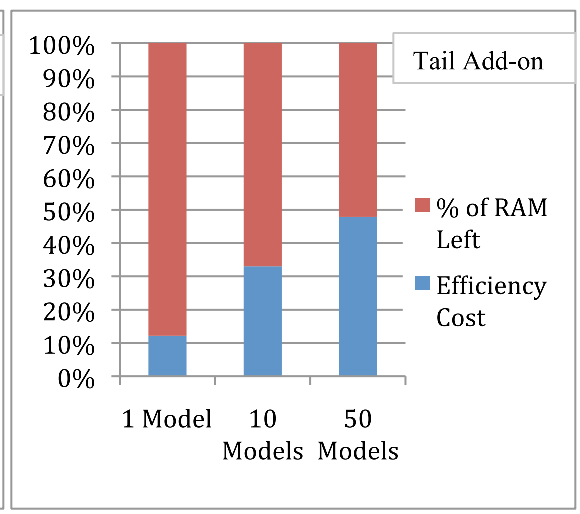
Graph 2: The efficiency readings of the server using our new models.

In order to address the efficiency question for the server side, an algorithm was devised that generates a large amount of random mermaids into the world while efficiency rates were logged.[6] After conducting over five trials that resulted in server failures, connectivity lags, local communication failures, and local computer crashes, it was determined that the server was running extremely slowly and that the technique has failed at a certain unspecified point. Since it was up to us to discover where the experiment failed exactly, it was decided that the previous methods had to be revisited testing the system with different variations of their reformations disabled. Once efficiency was restored to full, it became apparent that the reformations done during our experimental phase were the source of the problem experienced during initial balancing phases.

At this point, the results from graph1 had become the control scheme, and the system had to be tested while rendering a single reformation at a time. During the first simulation in this phase, the server with the hand reformations was enabled to be tested for efficiency costs. The results shown in Graph 3 were almost identical to the results in Graph 2 from our first test of the server, which undoubtedly means that this reformation was the glitch. Tracing over the reformed hand, it was realized that the ratio between the number of joints in the hand and the size skin model of the hand was very small. This small ratio resulted in a bottleneck of excessively large number of algorithms trying to process in relatively short time spans.



Graph 3: The efficiency readings of the server using the reformed hand model reformation only.



Graph 4: The efficiency readings of the server using the tail model reformation only.

This bottleneck was reflected in two defects, a hick in the renderer that constantly froze, and an extremely slow processor speed with a high level of the server's memory usage. For increased assurances that the problem was fully detected, a second simulation was done that only had the tail reformations enabled. The efficiency costs of that simulation, shown in Graph4, in comparison to Graph1 which was the control simulation were barely noticeable. Thus, the final model includes the partitioning, and the tail reformations with very basic traditional avatar hand skeleton.

CONCLUSION

This paper provides an overview of the procedure formed for the creation of a highly customizable, yet efficient mermaid avatar model for the *"Mermaids"* massively multiplayer online game (MMOG). We have outlined our technique through the development of three methods known as "Partitioning," "Skeletal Reformations," and "Balancing Customizability with Efficiency." These methods work together to split the main character model and main character skeleton into sub-models each with unique sub-skeletons, a process that grants the capability to change any of the sections during game-play without affecting the other sections. This technique succeeded in turning the avatar model into a customizable model on both primary and secondary levels, thus being able to exchange parts of the body as well as add ornaments to them.

ACKNOWLEDGMENTS

We would like to thank the Mermaids research director and lead creative game designer Professor Celia Pearce for providing us with this great opportunity to expand the horizon of our knowledge through research and experimentation. We also would like to thank Calvin Ashmore, the lead systems developer, for all time and his help. I would also like to thank Rose Peng, the lead concept artist from the character design team, for all her contributions to the research generally and the hand-drawn assets provided in this paper specifically. Rose is my partner and has had crucial contributions, specifically but not limited to, the conceptual design of the look and feel of the model. Also special thanks to Prof. Jeffery Donnell for his continuing guidance on this thesis paper.

GLOSSARY:

Within perspective of relative context, generally the word 'model' refers to the entire make up of a '3D model' (skin), its 'rig' (skeleton), and appropriate 'animations'. 'Modeling' is the art of creating a '3D model' (skin) and thus does not entail the remaining practices. Finally, when we state '3D model' we are actually referring to the skin without the rest of the model's attributes (the rig, and the animations).

[1]: Please note that the Mermaids team originally created all variations of customizable models, and that our research is to enable such capabilities, not to incorporate them into game-play, where designers have decided to grant players different levels of customization capabilities at different levels during the game.

[2]: "Modeling" is the art and process of creating 3D skin models.

[3]: "Rigging" is the art and process of creating a 3D skeleton for a skin model.

[4]: "Animating" is the art and process of animating previous modeled and rigged characters.

[5]: Secondary customizability features – models attached to models for customization purposes. In example, a fin attached to the tail.

[6]: In order to quantify the efficiency rate of the server and the system, we monitor changes in the rendering speed of the game – we perform the same action repeatedly and look for rendering freezes or hiccups. We also monitor the CPU processing speed and the memory usage levels of the server.

REFERENCES

[1] Baran, Ilya, and Jovan Popović. "Automatic Rigging and Animation of 3D Characters." *The ACM Portal*. MassachusettsInstituteofTechnology, 05 Aug. 2007. Web. <http://portal.acm.org/citation.cfm?doid=1275808.1276467>.

[2] Hodgins, Jessica K., Wayne L. Wooten, David C. Brogan, and James F. O'Brien. "Animating Human Athletics." *ACM*. College of Computing, Georgia Institute of Technology, Aug. 1995. Web. <http://doi.acm.org/10.1145/218380.218414>.

[3] Lee, Sung-Hee, EFTYCHIOS SIFAKIS, and DEMETRI TERZOPOULOS. "Comprehensive Biomechanical Modeling and Simulation of the Upper Body." *The ACM Portal*. University of California, Los Angeles, Aug. 2009. Web. <http://portal.acm.org/citation.cfm?doid=1559755.1559756>.

[4] Orvalho, Veronica C., Ernesto Zacur, and Antonio Susin. "Transferring the Rig and Animations from a Character to Different Face Models - Orvalho - 2008 - Computer Graphics Forum." *Wiley Online Library*. Universitat Politecnica De Catalunya, Barcelona, Spain, May 2008. Web.

<http://www3.interscience.wiley.com/journal/121403002/abstract>.

[5] Pan, JunJun, Xiaosong Yang, Xin Xie, Phillip Willis, and Jian J. Zhang. "Automatic Rigging for Animation Characters with 3D Silhouette." *Wiley Online Library*. 22 May 2009. Web. <http://onlinelibrary.wiley.com/doi/10.1002/cav.v20:2/3/issuetoc>.

[6] Park, Sang Il, and Jessica K. Hodgins. "Capturing and Animating Skin Deformation in Human Motion." *The ACM Portal*. Carnegie Mellon University, 23 July 2006. Web. <http://portal.acm.org/citation.cfm?doid=1179352.1141970>.

[7] Pearce, Celia, and Calvin Ashmore. "Principles of Emergent Design in Online Games." *The ACM Portal*. Georgia Institute of Technology, 04 Aug. 2007. Web. <http://portal.acm.org/citation.cfm?doid=1274940.1274955>.

[8] Talton, Jerry O., Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. "Exploratory Modeling with Collaborative Design Spaces." *ACM.org*. Stanford University, Dec. 2009. Web. <http://doi.acm.org/10.1145/1618452.1618513>.