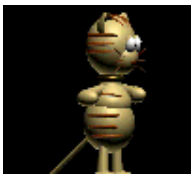

What A Cat Can Do ?

6.837 Team 31 Final Project Report

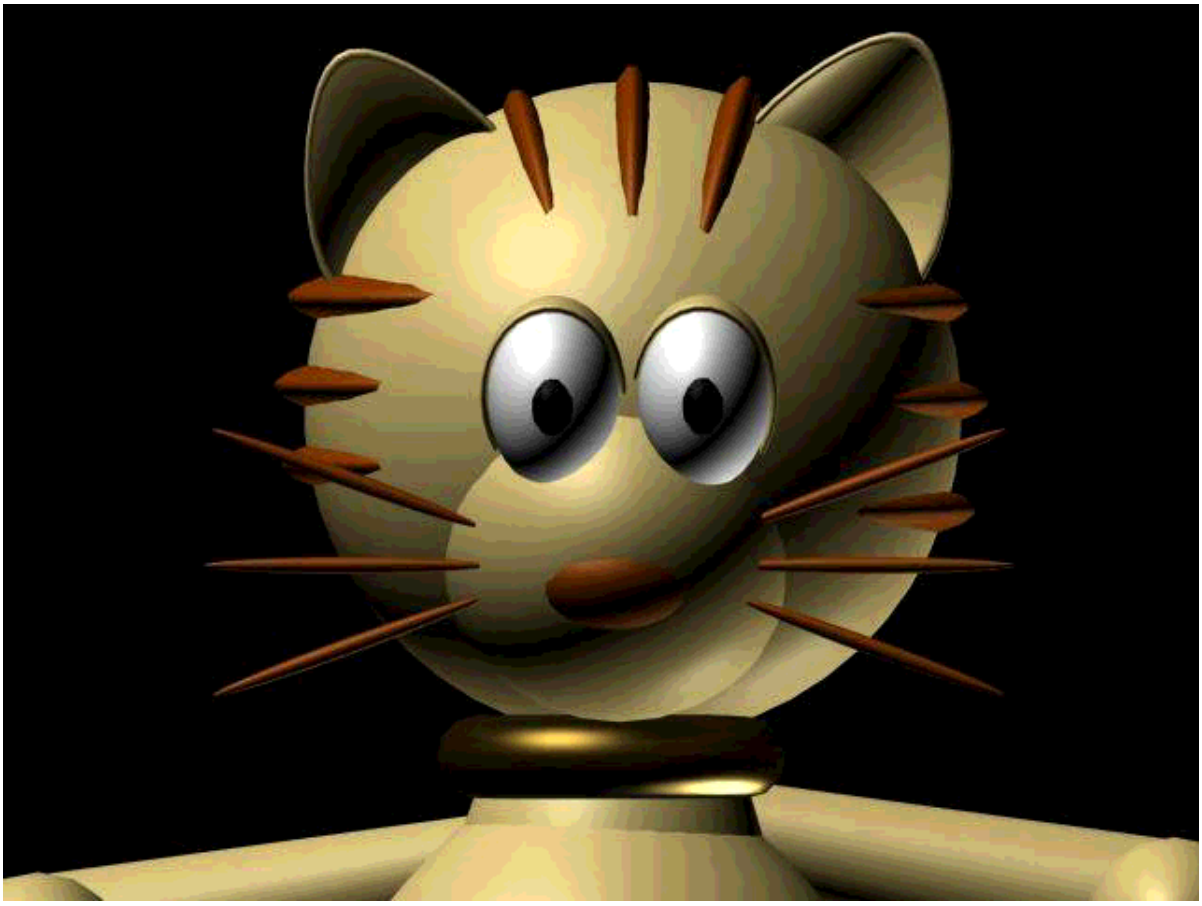

Fig.1 Cat modeled by 3D MAX

Abstract

"What A Cat can Do ?" is a project combines a short animation of one day life of a cat mainly produced by 3D Studio MAX with a prototype of real-time 3D interactive user interface which simulates some character behaviors driven from user's input via keyboard.

Some music effects are added into the interface. The real-time rendering and music playback are generated by Microsoft's Direct3D and DirectSound engine.

Introduction

Digital Creature has become a more and more considered topic of future research direction. By means of faster PC 3D accelerators and more sophisticated graphic development engines provided by various companies, the basically unfeasible development of real-time, quality, 3D virtual environment on PC platform has become possible recently. There are already some fascinating demonstrations or products on the market such as real time motion capture, real time motion simulation performed on powerful workstation level machines. However, due to speed issue, most PC level real time 3D computer games or applications are still confined to simple models with less polygon numbers or reduced texture mappings and colors.

As bandwidth and CPU capability problems will gradually decrease. The line between personal computers and workstation class machines is rapidly disappearing. More and more multimedia enriched products emerge to the market, as a result, users will become more critical toward the overall graphic quality and performance of an application.

The motivation of this project divides into two directions: one, to familiar with 3D modeling tools in order to have the ability of producing well designed models or animation for stand alone applications or 3D based game spirits and scenes. Two, to explore the possibility of generating real time 3D simulations using pre-designed objects and animation with different graphic APIs, to find out what can be done at current stage and what can't.There are several APIs developed to complete 3D rendering pipeline (light, transformation, rotation, etc.) which can access to the hardware acceleration capabilities available on different performing platforms. As hardware and software environment mature, 3D real time simulated applications with high resolution, accurate texture mapped, colored characters and scenes will definitely become the main stream in many aspects. This project is a first step for future possibilities as well as a milestone to demonstrate techniques learned in 6.837 lectures through out this semester.

Goals

The goal of this project is to explore 3D character animation as well as the procedure and technique needed to complete a 3D based real time rendering application.(a virtual environment, a game, or graphic demonstration tools) Within the limited time frame, we hope to gain an overall understanding of the mechanism by implementing a simple prototype of 3D real-time rendering interface on PC and devote also enough time for producing a short 3D character animation. At the same time, we want to find proper tool (or tools) to export the animation and modeling works into controllable file format and achieve some interactive controls using graphic APIs. Something we want to learn and achieve from this project:

- Familiar with 3D modeling tools. Build enough understanding toward necessary 3D animation skills.
- Build a good story line.
- Build an immersive 3D character and scenes.
- Produce a short animation with complete story line and sound effect.
- Learn digital video formats and nonlinear editing techniques.
- Export some animation clips or 3D modeled objects into controllable file formats.
- Find a suitable graphic API which can handle animated file with reasonable complexity.
- Programming user input interface and link it with proper character motion simulation.
- Add some sound effects to the interface.
- Achieve Real-time rendering with reasonable complexity if time permits.

Achievements

A.Character Animation Part

Being able to produce one short character animation is one of the two major goals of this project. The first two to three weeks were used to familiar with the available modeling tools, go through some examples provided by the reference books listed in the last part of this report. Concerned about the learning curve with available time, I chose 3D Studio MAX R2.5 as the main modeling tool instead of more powerful modeling software - Alias on Athena SGI machines. Together with 3D Studio MAX R2.5, I also use Metacreation's character modeling tool Poser 4, Scene modeler Bryce 4, as helpers for some behavior animation assessment and outdoor scene building. As for detailed picture editing, I used Photoshop as main graphic edit tool. After rendering out from 3D MAX, the avi files of clips were put together with sound effects and edited using nonlinear video editing tool Adobe Premiere. Major efforts and difficulties can be divided by the followings:

- Animation storyboard

  The story idea was firstly sketched out as Fig.2. It lays out what happens in one day for a cat's life. The animation will serve as the main background story as most of the current games and link out some dance animation, different behavior modeling which can be exported into controllable file format

for the 3D real time rendering engine.

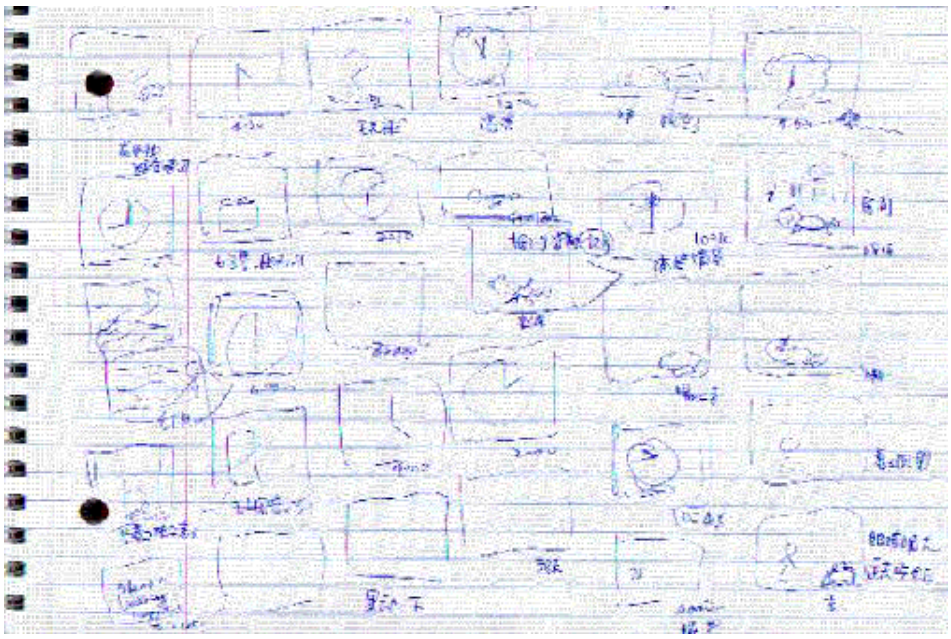

Fig.2 Main storyboard sketch

- 3D character modeling

Character modeling is one of the hardest but funniest parts of 3D modeling. I've done some basic character modeling using Metacreation's character tool Poser. Poser's pre-designed character models have well built skeleton, Inverse Kinematics Setting and joint limitation setting which is very easy to produce character animation with reasonable familiarity to the software. However, it also limits some imagination and you can only control part of the existing models. 3D Studio MAX is a powerful modeling tool. With it's plug-in Character Studio, we can also produce compelling character animation with a lot ease. Unfortunately, I still haven't got access to this powerful tool until writing this report. Therefore, I decided to do everything from scratch and get myself a chance to learn the essence of Hierarchy and IK setting. Some difficulties occurred alone the way:

- Correct geometry design- the cat model actually come from the snoopy model designed for assignment 3. I changed the head part the modified body, arm, legs by taper, rescale, rotate little by little.Here is the cat evolution course in Fig.3. The whole model takes about two whole day's work evolving from a dog to a cat.

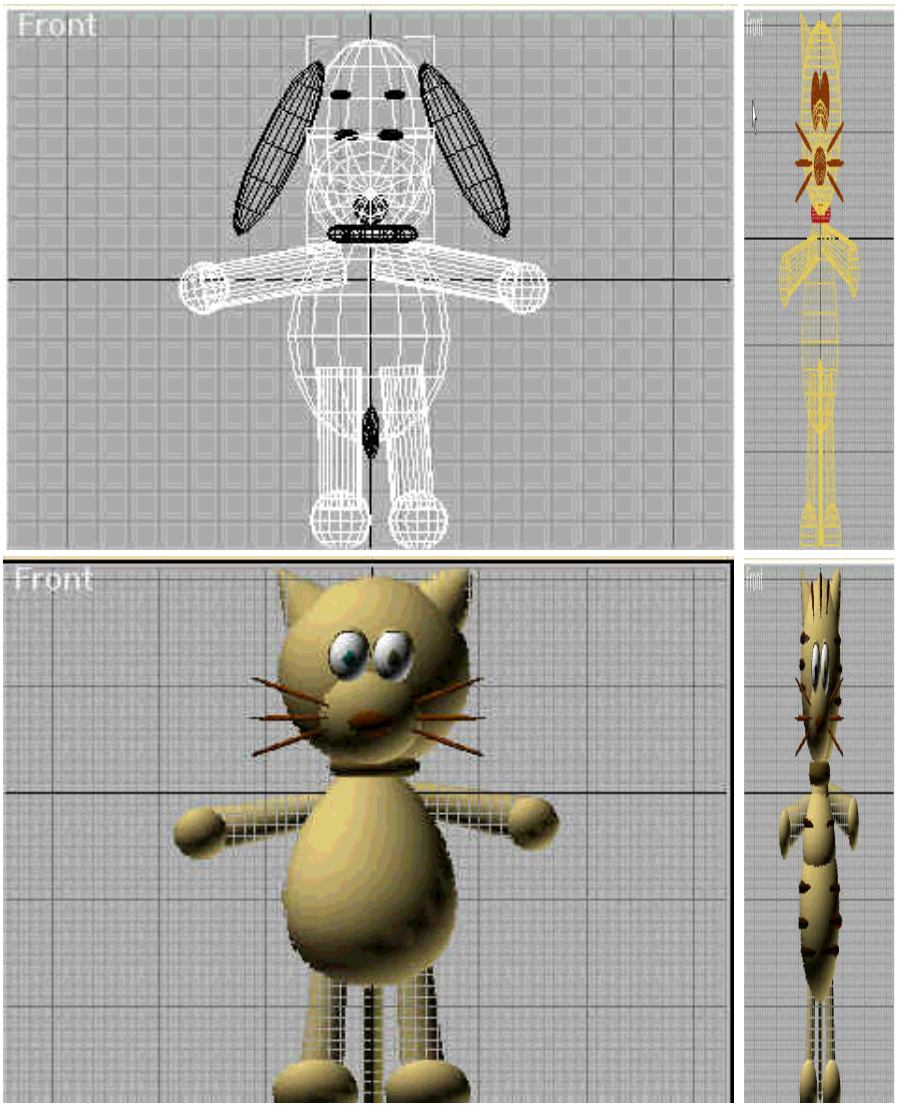

Fig.3 Correct geometry design course

- Eye design - one of the very important things in this animation is modeling a sleepy cat. In order to make it appealing, eyes with movable eyelids should be modeled. I learned the modeling trick for cartoon like eyes with future animatable ability from Maestri's great book [2]. With the book's help, it still takes about one whole day's work for two closable eyelids and movable pupils.

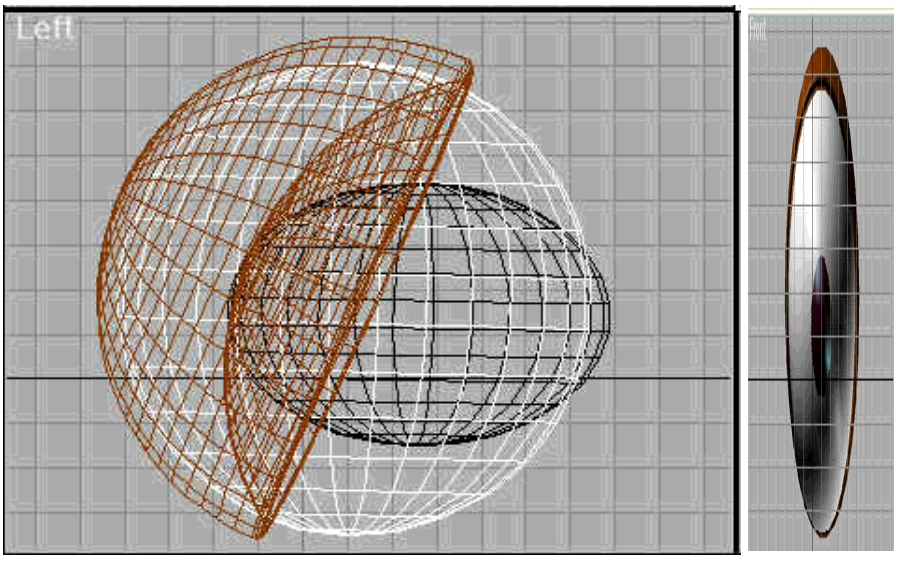

Fig.4 Eye modeling

- Ear design - besides eye, ear is the second hardest part for the cat model. I tried several possible solutions like cone, triangle, sphere etc., but the results weren't satisfactory. The final solution of a convincing ear comes from the eyelid with some taper and bend operations! A sudden inspiration, but works pretty good!

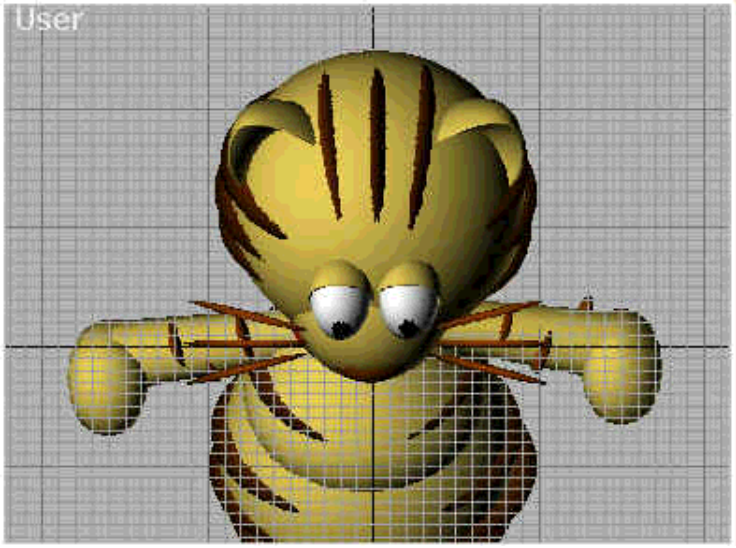

Fig.5 Ear modeling

- Hierarchy setting- the cat model doesn't contain too many different objects, however, to be able to turn the cat's head without leaving it's eyes behind. A correct and effective hierarchy setting can make life a lot more easier in the future.

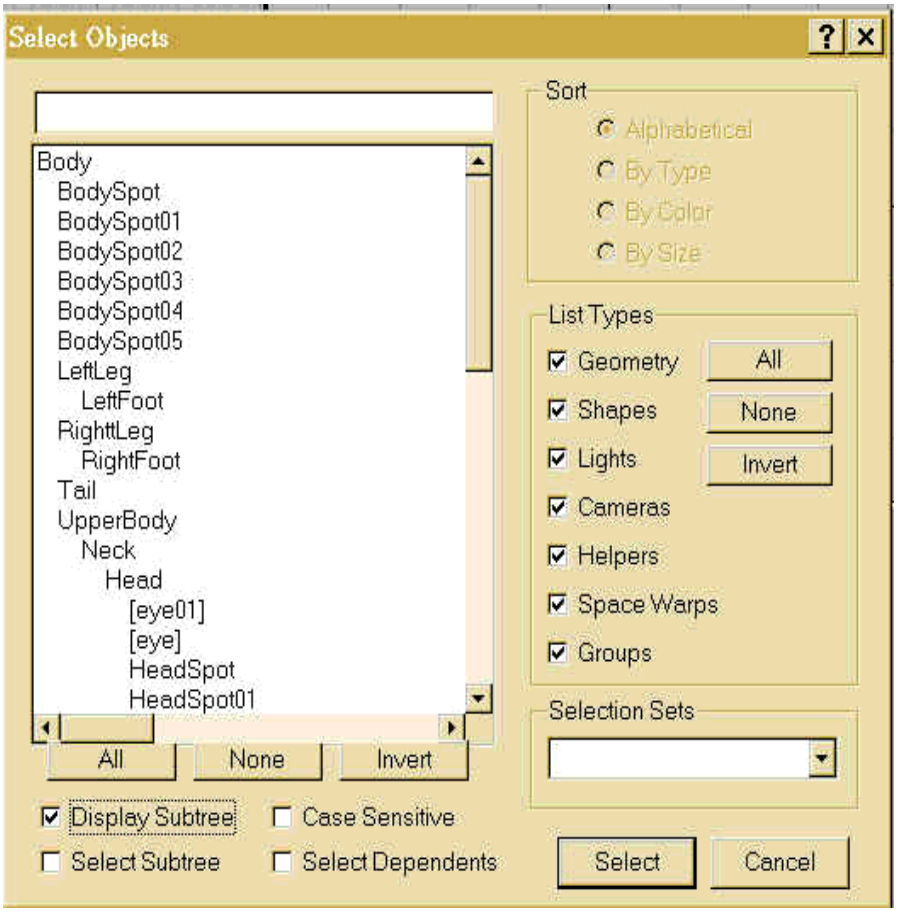

Fig.6 Hierarchy setting

- lnverse-Kinematics - Once we've set the right hierarchy, inverse kinematics can make posture easier. Also, we should set reasonable joint limits to make posture realistic. This procedure needs a lot understanding of how our body moves, try and error. The total time for setting the right hierarchy and inverse - kinematics is about two day's work. Thanks for Maestri's great book [2] again!

- Posture - Above parts are the real hard parts! After everything above works well. Posture is easy as long as you have enough observation toward the target character. However, since I tried to model the cat in a cartoon like way, some real life motions might not be able to achieve with this model, instead, we can build some human like behaviors. That's a compromise.

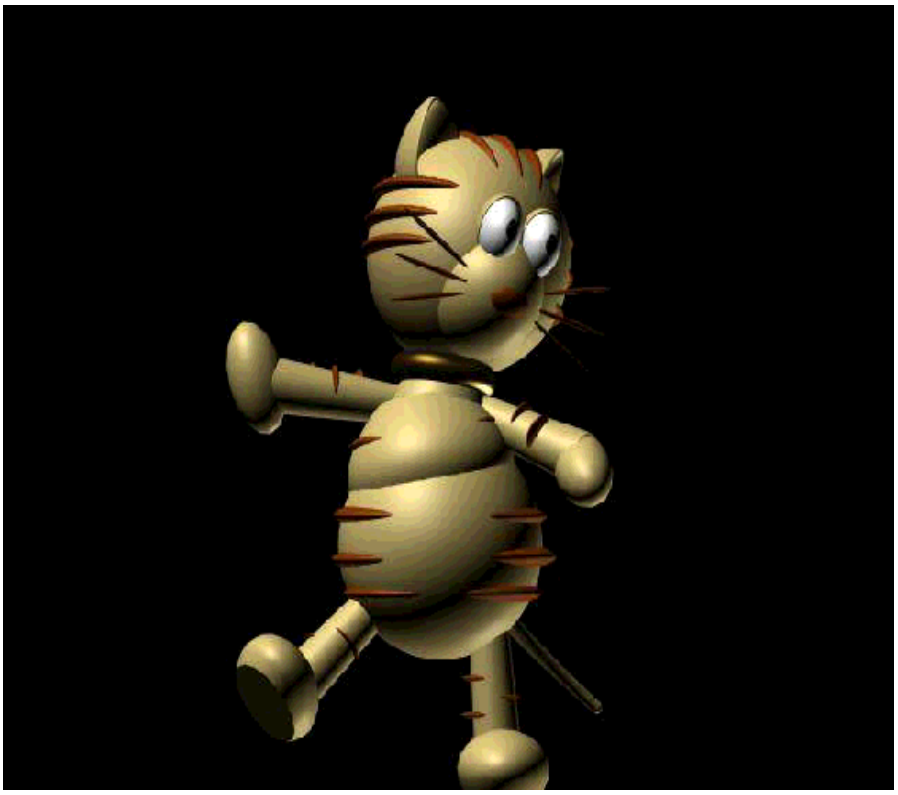

Fig.7 One posture design

- Scene Modeling

Scene modeling needs to import some existing object's with correct spatial coherence. It seems to be an easy job - just putting things together. Without enough experience and sense of scaling, it still needs a lot of time and experience. Also, a sense of space is needed. There are some memory issues: When the scene becomes complicated, it needs a lot time to be able to rotate or transform the whole scene and choose the right viewing angle. For the interior decoration, I modified some existing example models produced by Dei Jang Company to

save some time from doing all the furniture from scratch.

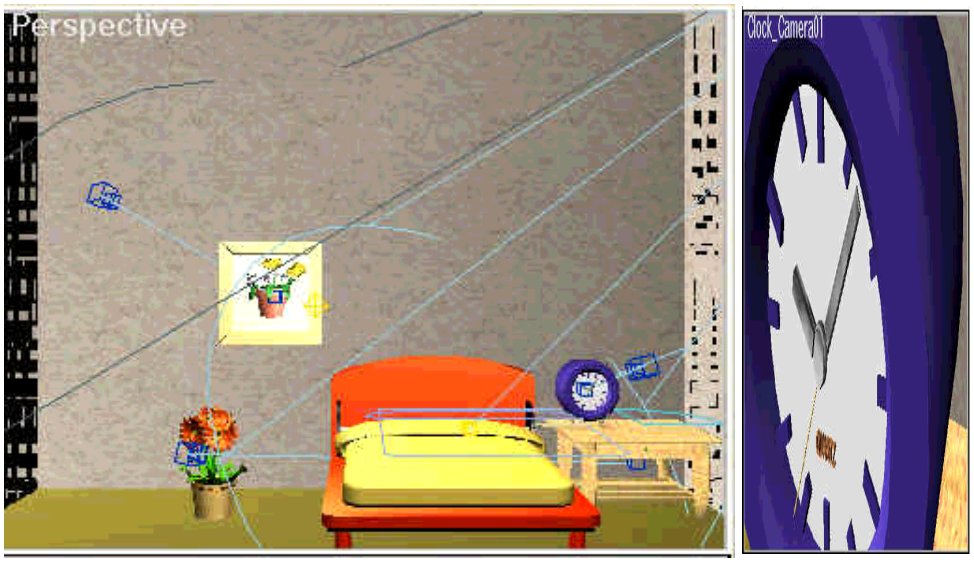

Fig.8 Two scene designs

- Lighting- the light setting of one scene might not look proper when we import other objects. The color of object will look really different when the light system is different. That's also something I didn't think about first.

- Camera setting - different camera setting can eliminate some unnecessary details for the "not seen" objects. I also learned a lot when I change the camera position to make story goes smoothly.

- Rendering problems

  I've rendered several really great still shots and short animation clips before doing this project. Which went OK. However, I don't realize that I was using 6 frames per second with lower resolution. And what I rendered was only one object without background scene. I spent a lot of time doing the modeling, scene design, and story elaboration. It wasn't until I, actually my computer, spent 75 mins for a 10 seconds clip, and came out with a 75MB avi file which I, actually my computer, should use 20 times more than the the origin duration to play the clip, if not dead, I discovered that with the available time frame and available machine, I can't render that much frames as I planned.

It’s a dilemma to choose between resolution and time. I chose time in order to finish the animation before deadline, but I hope I could find faster machines in the future to re-render the whole animation. I am still looking for solutions for smooth playback without losing too much quality.

Fig.9 Rendered scenes with 320X240 resolution

- Non- linear editing and sound effects

Adobe’s Premiere is one very powerful and professional tool for nonlinear editing and sound effects. I found some wav files from various resources and edit them into the animation. It also takes a lot of time to reproduce the clip with sound effect. Fortunately, Premiere only re-compute the scenes being edited and sound exists in different channel. Such that it won’t talk another 75 mins for the same 10 second clip. It can playback in full screen mode and export to video. However, the huge file size is again the problem for smoothly editing and playback with machine available to me now.

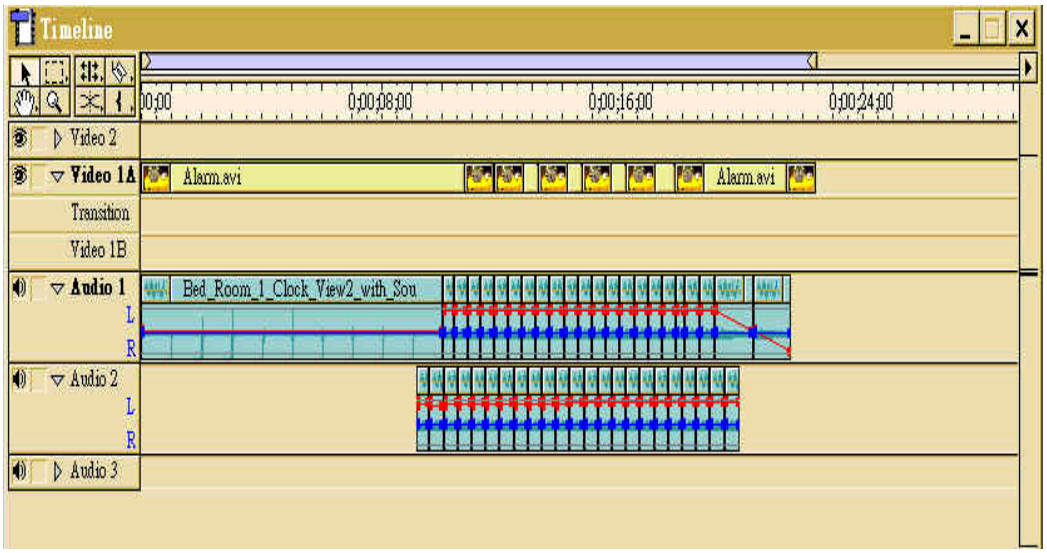

Fig.10 Non-linear edting environment

B.Control Part

The second important part of this project is the control part. However, this is also where we confronted significant obstacles in trying to realize the goals of this project. We spent a lot of time assessing possible solutions for the real-time 3D engine. Major efforts were put on choosing between Java3D and Direct3D from Microsoft. After reading limited reference books and online docs available for Java3D and Direct3D APIs. We discovered that Direct3D is more powerful and seems to have the solution for our goal- converting pre-designed animation with keyframe, texture mapping information. However, we don't have experience with Direct3D APIs and many people I consulted with warmed me that Direct3D is not that trivial to implement even if for an experience C++ programmer. Java is relatively easier to implement within the time frame, but again, we haven't found real time rendering solution for complicated 3D animation sets yet. Therefore, we divided our efforts into two directions:

- Java

  Java 3D API seems not have the right solution for importing our animation produced by 3D MAX. Therefore, we tried to use Java 2D graphic APIs as a substitution like many existing games do. We tried to use

Java AWT and Swing APIs as our interface for playing back our 3D animation to produce a semi-real-time interactive interface. In order to make seamless transfer from one animation to the other, if without interpolation, we have to make every motion clip starts and ends at the same position. Mike Johnson in the Media Lab demonstrated to me some interpolators written in Java, that could serve as a solution for our goal. However, many underlying codes written for the group in many years are not easy to transfer out in a short time. Thanks for Mike's valuable experience. We didn't use this valuable resource because of concerning about available time frame and tools at hand. Special thanks to my team member Charles Lee's enormous help on exploring Java AWT and Jave Media Frame APIs. Charles Lee helped me a lot even if he does not take this course for credit and have a lot course work from other subjects. Fig.11, Fig.12 illustrates the basic idea of animation clips.

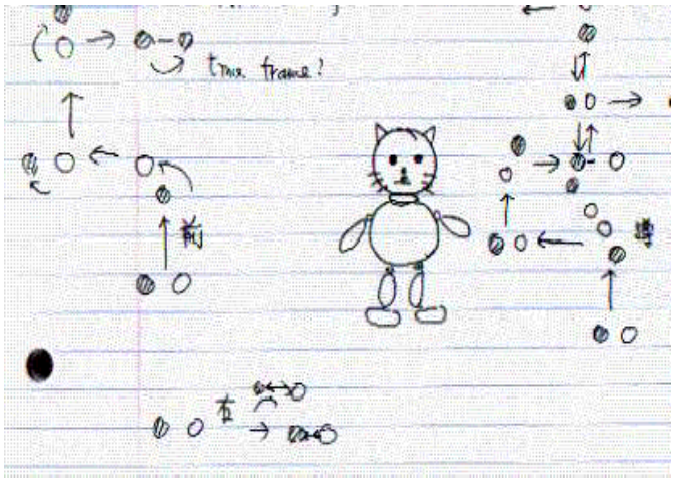

Fig.11 Continuous animation clip design idea sketch

Fig.12 Test dance clips

We spent a lot of time surveying the possibility using Java as our real time rendering software engine. We also reference materials about Java 3D game development environment. Unfortunately, they are all limited with simple polygon rendering. I believe there must be ways in Java beyond our survey which can achieve our goal. A pity that it’s not realistically possible for us to address them in the time we have. But the learning experience is by far the most valuable part. I hope the project is only

my first step toward future exploration in the graphic field. At current stage, Java3D is too new, available reference is very limited. We didn't achieve our goal using java. It was a painful decision to change back to Direct3D especially with such limited time.

- Direct3D

  Direct3D provides a complete suite of real-time graphics service which provide very fast software-based rendering of the complete 3D rendering pipline (light, transform, etc.)[1] and can access to the hardware acceleration capabilities available on the target PC platform. Together with the integration of DirectX, ActiveX technologies, Direct3D is able to bring capabilities such as video mapping, hardware 3D rendering in 2D overlay planes, and sprites to the home PC. [1] Libraries provided by Direct3D do have the solution of our project goal.

  However, before being able to implement some applications using Direct3D without prior experience, the learning curve is a little beyond realistic concern in the time we have.

  The first step is trying some available examples to see if we could successfully import predesigned animation. There is a shareware convertor which can convert 3ds files to .x files for Direct3D control. I tried to convert several 3ds file like the above test boy model as well as my cat model. I failed in keeping the texture mapping information and some details for the boy. As for the cat, only two whisker left. See Fig.13.

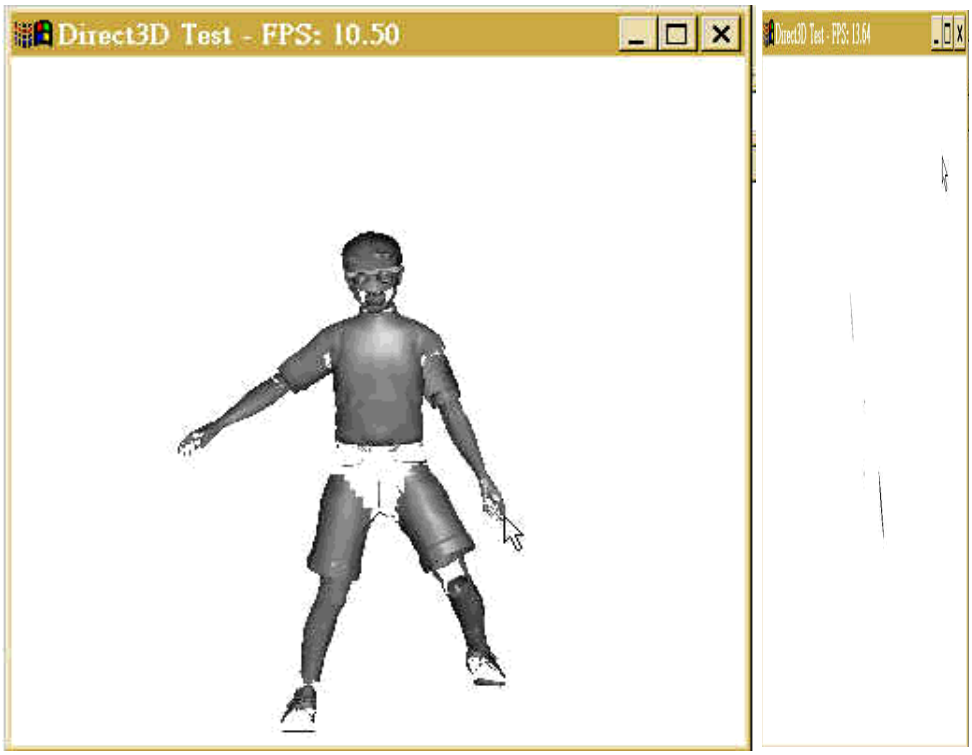

Fig.13 Failed tests of converting 3ds file into Dircet3D controllable format

I changed several parameters but still can't keep the right model. After a lot of try and error. I finally found out that the converter available has some limitation when dealing with parent - child setting as well as texture mapping , that's the main reason why my cat came out with only two whiskers. After realizing this limitation, I tried some simpler model and animation, finally, it came out OK! See Fig.14.

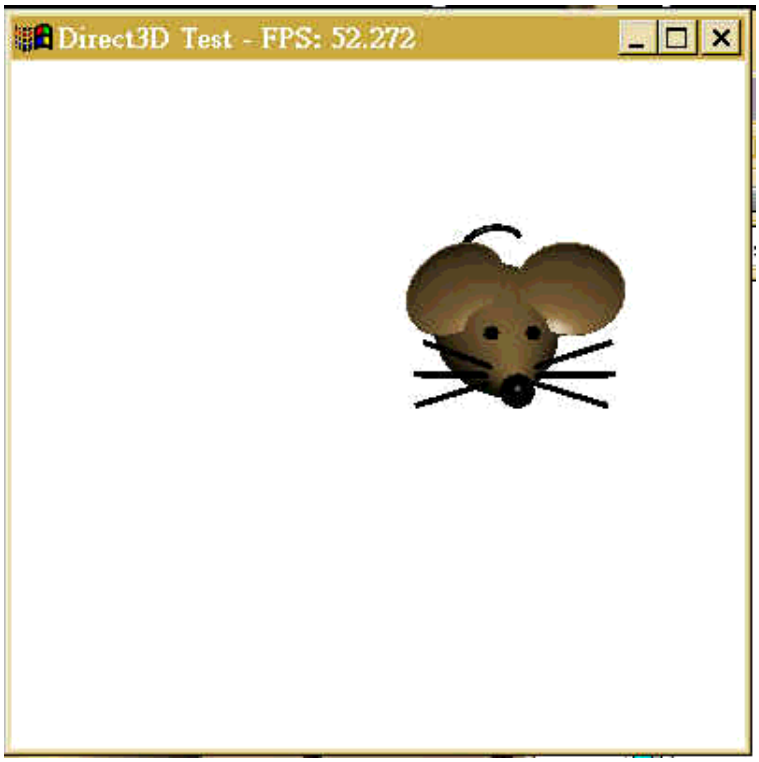

Fig.14 Successful test of converting 3ds file into Dircet3D controllable format

Ben from the Media Lab kindly provided me resources to convert 3D MAX files into AGT ASCII files. The result went OK, at least my cat model has body and legs, the animation goes smoothly, too. See Fig.15. Some colors are not that accurate, but that's minor problem. The major problem is, to there is still another convertor needed to convert AGT files into .x file format for Direct3D. I still haven't get access to the convertor, unfortunately. So, at least I know there do exist solution for Dircet3D. If I could get access to the necessary tools, implementing great real time 3D application on a PC with reasonable performance is really feasible.

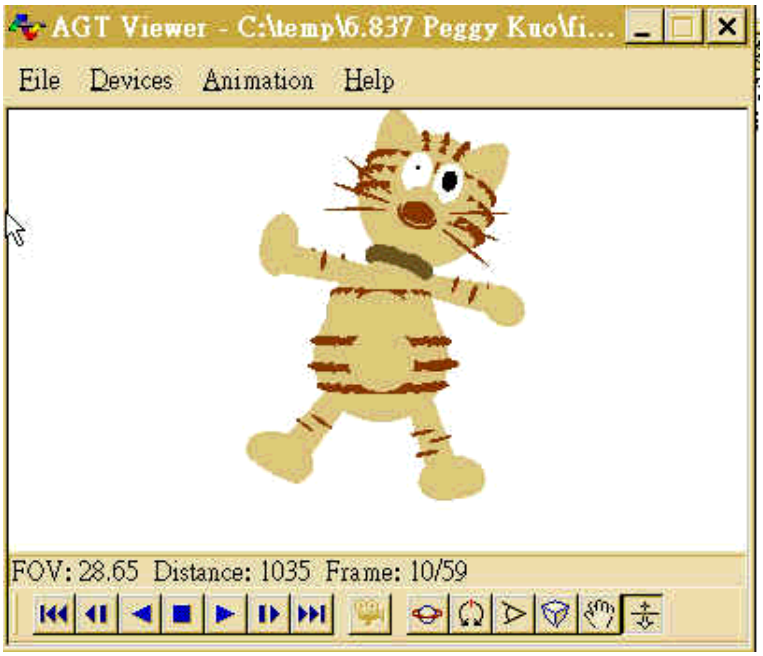

Fig.15 Test of converting 3d MAX file into AGT file format

Concerned with the available time, we move back to simpler model animation with no hierarchy, no IK setting. It went pretty well.See Fig 16. Currently, there are some working simple user interactions and sound effect.

Our exploration with Direct3D wasn't that successful. We used most of the time finding out what we can't do and not much time left for us to fully implement what we can do.

Lessons Learned

Tremendous experience gained from implementing this project !

Acknowledgments

- Professor Seth Teller, 6.837 professor- thanks for his teaching and advice during the whole semester.
- Charles B Lee, 6.837 team TA -special thanks for his patiently guidance and encouragement.
- Meng Hang Ho, Master Candidate of EE Dept, National Taiwan University - Special thanks to his numerous programming technical supports, idea brainstorming, encouragement, valuable references.
- Hao Yan, MIT Media Lab Gesture and Narrative Language Group - special thanks to

his continuous programming guidance, encouragement throughout every 6.837 assignments.
- David Da Gang Chiou, MIT Media Lab Sociable Media Group - thanks to his technical consulting and reference materials.
- Yi-San Lai, PhD candidate, MIT Ocean Engineering Dept - thanks to his kindly provided 3D modeling experiences, modeling tool evaluation and recommendation, reference materials.
- Ben Resner, MIT Media Lab Synthetic Characters Group - thanks to his valuable information and help about exporting 3D MAX objects and animations into controllable ASCII files (in AGT format).
- Michael Patrick Johnson, MIT Media Lab Synthetic Characters Group - thanks to his analysis and real time 3D methodology introduction and recommendation.
- Scott Eaton ,Marc Downie, Bill Tomlinson,Robert Burke ,Song-Yee Yoon, MIT Media Lab Synthetic Characters Group - thanks for their kindly provided references and experience.

Bibliography

- [1] Peter J. Kovach, "The Awesome Power of Direct3D/ DirectX," Manning, ISBN 1-884777-47-3, 1998. Ch1-16.
- [2] George Maestri, "Digital Character Animiation," New Riders, ISBN 1-56205-559-3, 1996. Ch1,2,5-12.
- Michael Stein, Eric Bowman, Gregory Pierce, " Direct3D Professional Reference", New Riders, ISBN 1-56205 -725 -1.
- Chismar, "3D Studio MAX 3," New Riders, ISBN 0-7357-0050-8, 1999.
- Anthony Potts, David H. Friedel, Jr, "3D Studio MAX Design Guide," ASIN 1-88357-783-7, 1996. Ch1,4,5,7,9,10,16,19,21.
- Steven Elliott, Phillip Miller, "Inside 3D Studio MAX," New Riders, ISBN 1-56205-427-9, 1996.
- Gotop Technology, "3D Studio MAX R2," Gotop Information Inc., Taiwan, 1998.
- Adobe Press, "Adobe Premiere," Adobe Press, ISBN 1-56830-119-7, 1994.
- Bill Fleming, "3D Photorealism Toolkit," Wiley, ISBN 0-471-25346-4, 1998.
- Bill Fleming, "3D Modeling & Surfacing," Morgan Kaufmann, ISBN 0-12-260490-3, 1999.
- Fan, Ries, Tenitchi, "Black Art of Java Game Programming", Waite Group Press, ISBN 1-57169-043-3.
- Adobe Press, "Adobe Premiere," Adobe Press, ISBN 1-56830-119-7, 1994.
- Web3D Consortium ,http://www.vrml.org/.

This page was last updated Dec 3,1999