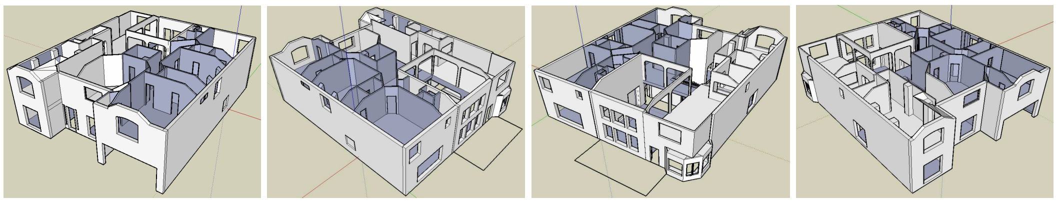
Digital Game and Animation – Fall 2008
Project Description (version 0.1)

Virtual “Reality Show” and Digital Detective Game



Introduction

This project is composed of two parts: automatic animation and a digital game. Given a ***house model***, the animation part consists of animating a family of ***virtual characters*** in this environment. These autonomous virtual characters will move around the house and perform a ***set of actions***. The house will be equipped with a number of fixed ***virtual cameras*** that will show every move of each autonomous character in each room. This way, the autonomous animation will virtually simulate a “reality show” such as big brother.

The second part of this project consists of a digital game where a detective investigates a murder mystery to identify the assassin. In this game, the detective and the assassin are family members. These two characters are the ones that are actively played. The assassin will have to follow a script and playfully murder another family member. The game will generate a number of crime clues. While the detective tries to find the clues, follow them, and discover the assassin; the assassin tries to erase these clues and remain undetected.

The project is divided into four parts: (I) object and character repository, (II) modeling, (III) animation, and (IV) gaming. In the first part, we will search for digital objects and virtual characters freely available in the internet to populate the house. The second part involves modeling additional pieces of the house, designing the lighting, and placing the virtual cameras. The third part concerns the implementation of motion-based animation techniques. The fourth part considers gaming issues to implement the functionalities required by the digital detective game. In the next sections, we will further describe each part of this project.

Part I. Object Repository and Virtual Characters

The virtual environment consists of a two-story house with a number of compartments especially created for this project (see Fig. 1). In the first story, we have: (1.1) master bedroom, (1.2) master bathroom with closet, (1.3) utility room with storage, (1.4) family room, (1.5) kitchen with pantry, (1.6) breakfast area, (1.7) half-bathroom, (1.8) formal dining, (1.9) formal living, (1.10) study, and (1.11) garage. In the second story, we have: (2.1) boys bedroom with closet, (2.2) boys bathroom, (2.3) girls bedroom with closet, (2.4) girls bathroom, (2.5) library/play room with closet, (2.6) guest room with closet, (2.7) game room, (2.8) bar, and (2.9) media room with closet.

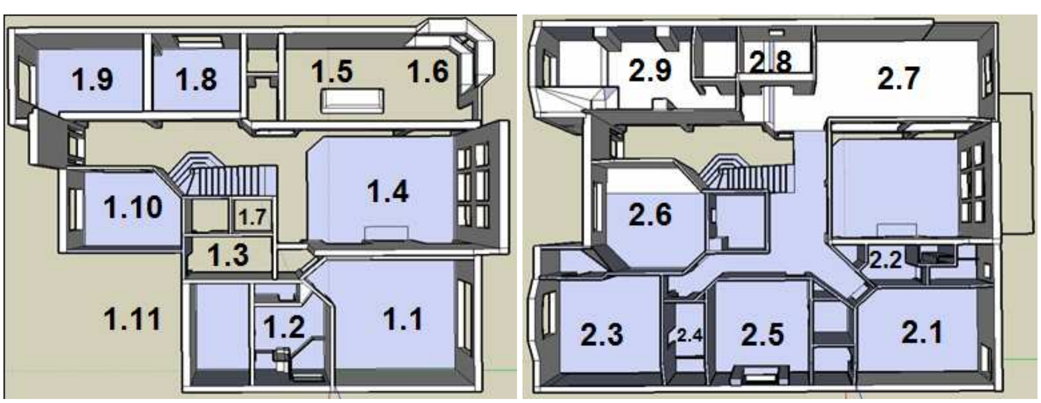


Fig. 1: Room compartments.

During Part I, all students will search for digital 3D objects and virtual characters freely available in the internet to populate the environment. We are also seeking for objects that will serve as weapons, devices, or methods for the assassin to commit the murder. A list of suggested objects for the master bedroom is provided as an example in Appendix A. Students are welcome (not required) to create their own objects manually or they could use a 3D laser scanner to generate digital objects (*i.e.*, meshes with texture) from real objects. However, a list of internet links where 3D objects are available is provided in Appendix B. Each object is associated with a list of actions that determines how that object can be used. In the search for digital objects, each student will cover a subpart of the two-story house. The house subparts are:

- Adult bedrooms: 1.1 and 2.6;
- Bathrooms: 1.2, 1.7, 2.2, and 2.4;
- Utility areas: 1.3, 1.10, and 1.11;
- Family areas: 1.4 and 2.5;
- Kitchen areas: 1.5 and 1.6;
- Formal areas: 1.8 and 1.9;
- Children bedrooms: 2.1 and 2.3;
- Recreation areas: 2.7, 2.8, and 2.9.

For the virtual characters, the search will be a combined effort of all students. Ideally, we are looking for eight characters composing a family: (1) father, (2) mother, (3) teenage boy, (4) teenage girl, (5) young boy, (6) young girl, (7) grand-father, and (8) grand-mother. We also want a list of actions, activities, or behaviors each character could perform in such environment.

The deadline for this assignment is ***September 25th***. Students will submit the list of objects and virtual characters (downloaded – 1pt, modified – 4pts, digitally scanned – 12pts, and manually created – 20pts), the respective data files (with links to sources), a list of actions for each object, and a list of behaviors for characters. At this point in the project, we will have to agree in a standard programming language to implement looking into the gaming engine to be used.

Part II. Modeling

Part II concerns additional modeling that will (1) improve the house model and (2) include lighting and camera functionalities. The first task in Part II involves modeling additional detailed part of the house including doors, windows, handrails with poles, cabinets/shelves/drawers, and lighting (including associated 3D objects). This task also includes texturing the whole house and the placement of all objects found in the repository search. The two students () involved in this task may use Google SketchUp or Blender. The deadline for this task is ***October 16th***. Students will submit the final 3D model of their improved house.

The second task in Part II concerns the placement of virtual cameras. The camera placement will involve the implementation of sensor placement algorithms that will satisfy some constraints such as each point in the scene must be seen by at least two cameras. The algorithm will read the geometry of the house model and automatically find a set of projective cameras (internal and external parameters) that satisfy the given constraints (*i.e.*, cover the whole house space). The two students () involved in this task will have until ***October 30th*** to finish. Students will submit their source code and the output for the given house model with the camera locations.

Part III. Animation

Automatic animation consists in generating motion data to a virtual character in order to produce a video animation without manual intervention. Recently, motion-based approaches have being proposed to synthesize animation automatically. A motion-based approach is designed to reuse a human motion database to generate novel sequences of motion. Part III concerns the implementation of motion-based techniques for automatic animation. Since motion-based methods use a corpus of motion capture data to synthesize movement automatically, we will use motion capture data available in the internet (see a list of links in Appendix C), the Human Motion Database being built at UTA, or we may capture our own actions using MoCap equipment during a demo to be announced. We will use motion-based algorithms to turn the virtual characters into autonomous agents moving around the environment realistically. A list of technical papers covering each motion-based animation subject is suggested below. Each student will choose, with the instructor's agreement, one or two techniques to be implemented (or use code available in the internet).

There are four tasks in the motion-based animation part: retargeting, generalization, transitioning, and splicing.

Retargeting. In this task, we will study the retargeting of human motion from one source skeleton to a target skeleton (or a mesh). Basically, the human motion retargeting problem consists in converting the motion of one subject (*e.g.*, a tall person) into the motion of another subject (*e.g.*, a short person) satisfying a set of constraints that keep the motion realism intact. The students () involved in this task will learn and improve the Python script that allows Blender to load bvh files. They will reuse available retargeting code to transfer motion data from a skeleton to a virtual character's mesh. The goal is to make the process of attaching the motion to a Blender body structure seamless. The deadline for this task is ***October 16th***.

Generalization. Given a number of sample motions for a particular action and the respective arguments for a set of parameters defining this action. These parameters may be target location, speed, resistance force, and others. The generalization task concerns to adapt this action to be performed for any instantiation of a set of parameters. For example, given a set of reach actions for different target locations, we want to generalize the reach action to any reachable target location. The students () involved in this task will implement one or two techniques cited below using a standard programming language (*e.g.*, C++). The algorithm will be given a number of sample performances for an action and find a representation that allows the execution of such action at any generalized parameter configuration. The deadline for this task is ***October 16th***.

Transitioning. The sequential concatenation of different actions is the concern of this task. Given two actions to be performed in sequence (one after the other), we want to verify whether their concatenation is feasible and find the motion transition that takes a virtual character from one action to the next action. The students () involved in this task will implement one or two techniques cited below using the standard programming language. The deadline for this task is ***October 30th***.

Splicing. In this task, we will study the splicing of two different human actions into a single motion. Basically, the human motion splicing problem consists in merging two motions performed by different body parts into a single whole body motion such that the motion realism is intact. The students () involved in this task will implement one or two techniques cited below using the standard programming language. The algorithm will be given two actions and the respective motion representations. Their code will decide whether these two actions may be spliced (or not) and perform the merging (if possible). The deadline for this task is ***October 30th***.

Part IV. Gaming

Part IV involves adding functionality to our autonomous agents in order to transform our animation system into a digital game. The first task considered in the gaming functionality is motion planning. Motion planning consists in organizing basic actions into a hierarchy of behaviors and providing ways to find good plans to execute high-level actions using the basic activities. In a lower level, the motion plan will use path planning to select the activities required to achieve some goal: locomotion, non-locomotion, and manipulation. Students will work on the standard programming language, Blender game engine, or Microsoft XNA. The students () involved in this task will have until ***November 25th*** to complete this part.

The second task in the gaming part includes all the remaining functionality to transform the animation system into a digital game. The digital game must include the logic behind the user-activated characters (detective and assassin), reasoning, user control, automatic camera selection, character communication, and other features. Students will work on the standard programming language, Blender game engine, or Microsoft XNA. The students () will have until ***November 25th*** to finish the project.

Calendar

September 25th – Object Repository and Virtual Characters
October 16th – Modeling (VS, JE), Retargeting (AS, BT), Generalization (HB, KW)
October 30th – Camera Placement (EB, GR), Transitioning (KW, AS), Splicing (BT, HB)
November 25th – Planning (GR, VS), Reasoning (JE, EB)

Bibliography

Camera Placement:

Automatic sensor placement from vision task requirements [169 citations]

Retargeting:

Approximating character biomechanics with real-time weighted inverse kinematics. [0 citation]
Rapid Animation of Laser-scanned Humans [0 citation]
From motion capture to action capture: a review of imitation learning techniques and their application to VR-based character animation. [2 citations]
Validating retargeted and interpolated locomotions by dynamics-based analysis. [1 citation]
Facial motion retargeting [3 citations]
Semantic 3D motion retargeting for facial animation [4 citations]
Motion retargeting in the presence of topological variations. [0 citations]
Motion retargeting and transition in different articulated figures. [1 citation]
Motion normalization: the preprocess of motion data. [2 citations]
A Multiresolutional Approach for Facial Motion Retargetting Using Subdivision Wavelets. [0 citation]
Motion Retargeting for the Hand Gesture [1 citation]
Hierarchical Retargetting of Fine Facial Motions. [16 citations]
Example-based motion cloning. [8 citations]
Mapping optical motion capture data to skeletal motion using a physical model. [17 citations]
Motion capture assisted animation: texturing and synthesis [169 citations]
Using motion analysis techniques for motion retargeting. [6 citations]
Expression cloning. [174 citations]
Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting. [39 citations]
Hierarchical approach to interactive motion editing for human-like figures. [263 citations]
Retargetting motion to new characters. [412 citations]

Generalization:

Construction and optimal search of interpolated motion graphs. [5 citations]
Parametric motion graphs. [6 citations]
Style translation for human motion. [43 citations]
Analyzing the physical correctness of interpolated human motion. [27 citations]
Style-based inverse kinematics. [139 citations]
Automated extraction and parameterization of motions in large data sets. [100 citations]
Computing the duration of motion transitions: an empirical approach. [22 citations]
Automatic Gait Generation for Various Stairs. [0 citation]
Human motion and emotion parameterization. [6 citations]
Flexible automatic motion blending with registration curves. [85 citations]
Using motion analysis techniques for motion retargeting. [6 citations]
Verbs and Adverbs: Multidimensional Motion Interpolation. [342 citations]
Motion warping [411 citations]
Motion signal processing [498 citations]

Transitioning:

Interactive and flexible motion transition. [1 citation]
Construction and optimal search of interpolated motion graphs. [5 citations]
Near-optimal character animation with continuous control. [1 citation]
A Dynamics-based Comparison Metric for Motion Graphs. [0 citation]
Automatic Construction of Compact Motion Graphs [0 citation]
Parametric motion graphs. [6 citations]
Evaluating Data Driven Character Animation [1 citation]
Fat graphs: constructing an interactive character with continuous controls. [10 citations]
Precomputing avatar behavior from human motion data. [33 citations]
Motion retargeting and transition in different articulated figures. [1 citation]
Fast and accurate goal-directed motion synthesis for crowds. [24 citations]
Motion modeling for on-line locomotion synthesis. [21 citations]
Group motion graphs [15 citations]
On-line adapted transition between locomotion and jump. [3 citations]
Evaluating motion graphs for character navigation. [24 citations]
Computing the duration of motion transitions: an empirical approach. [22 citations]
Modeling Physical Capabilities of Humanoid Agents Using Motion Capture Data. [5 citations]
On-line motion blending for real-time locomotion generation. [29 citations]
Motion synthesis from annotations. [130 citations]
An evaluation of a cost metric for selecting transitions between motion segments. [37 citations]
Planning biped locomotion using motion capture data and probabilistic roadmaps. [62 citations]
Snap-together motion: assembling run-time animations. [60 citations]
Motion graphs. [367 citations]
Interactive control of avatars animated with human motion data. [299 citations]
Interactive motion generation from examples. [233 citations]
Motion texture: a two-level statistical model for character motion synthesis. [204 citations]
Motion capture assisted animation: texturing and synthesis [169 citations]
Efficient generation of motion transitions using spacetime constraints [117 citations]

Efficient generation of motion transitions using spacetime constraints []

Splicing:

Imposing constraints on fragmented body motion for synthesis. [0 citation]
Splicing Upper-Body Actions with Locomotion. [3 citations]
Automatic splicing for hand and body animations. [1 citation]
Adding Hand Motion to the Motion Capture Based Character Animation. [0 citation]
Enriching a motion collection by transplanting limbs. [18 citations]
Combined Partial Motion Clips [1 citation]

Appendix A. Object Repository for Master Bedroom

_____ King bed
    _____ Headboard
    _____ Footboard
    _____ Rails
_____ Bed Frames
_____ Mattress
_____ Nightstands (2)
_____ Dresser
_____ Mirror
_____ Chest
_____ Armoire
_____ Lamps (2)
_____ Pillows (2)
_____ Curtains
_____ Curtain Rods
_____ Painting
_____ Flat Panel TV
_____ TV Antenna
_____ Alarm Clock

Appendix B. Digital Object Sources in the Internet

http://e2-productions.com/repository/modules/PDdownloads/
http://www.blendermodels.org/
http://www.katorlegaz.com/3d_models/
http://blender-archi.tuxfamily.org/Models
http://www.blender3darchitect.com/2008/05/31/download-of-furniture-models-for-blender-3d/
http://wiki.blender.org/index.php/Resources/Models
http://blenderartists.org/forum/showthread.php?t=26564
http://resources.blogscopia.com/index_en.html
http://telias.free.fr/Models_menu.html
http://www.blenderwho.co.uk/
http://dmi.chez-alice.fr/models1b.html
http://www.aroundthesims.com/

Appendix C. Motion Capture Data Sources in the Internet

http://mocap.cs.cmu.edu/
http://centralsource.com/blender/bvh/files.htm
http://www.animazoo.com/Free_downloads.aspx
http://accad.osu.edu/research/mocap/mocap_data.htm
http://gl.ict.usc.edu/animWeb/humanoid/
http://www.es3d.com/index2.html
http://truebones.com/freedemo/FreeBones.zip
http://www.truebones.com/JediBonesSeperated.zip
http://www.e-motek.com/entertainment/downloads/
http://www.audiomotion.com/downloads.html#mocapdata
http://www.charactermotion.com/products/motion_capture/index.html
http://www.mocapxtras.com/
http://www.mocapdata.com/
http://www.freemotionfiles.blogspot.com/