Benedikt Kaempgen **University of Exeter**
Erasmus **2006**

Modelling and Animation (COM 3404)

Documentation

The documentation describes the process of creating a video presentation for a continual assessment in the module *Modelling and Animation*.

Introductory Description

The proposed toy for Toys4U is a table game played with realistic figures, thus named **Real-T-Game**.

The Modelling and Animation was done with Houdini, the Postproduction with Adobe Premiere.

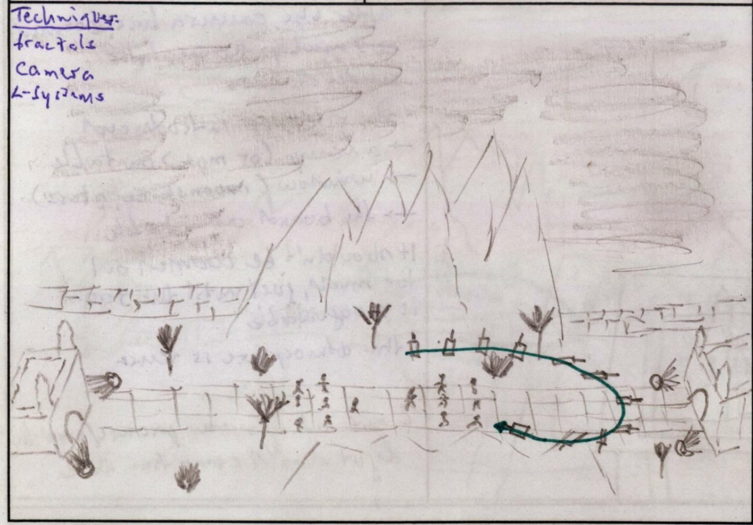
Storyboard

In the video presentation, the spectator first sees a beautiful landscape. The camera then shows a blue castle and blue figures. Still the walking blue figures focused the camera goes towards them and passes them. After that, the camera quickly turns to show a red castle at the other side and red figures coming from there. After stopping right behind the blue figures and waiting briefly, the blue figures turn, the red figures look up and both armies look into the camera. After a short while, the camera quickly zooms out to reveal the fact that the figures are playing on a table game located in a room. Two dices seem to be thrown by the spectator towards the game. The camera is zooming in to watch them fall. Also all figures look at them until they finally hit the ground twice and then turn out to be two 1s. The red team cheers about that and the blue team shows their anger by stamping on the ground and looking reproachfully towards the camera/spectator. Quickly after that, they start to shrink getting enclosed into red spheres which then vanish into thin air. Before that happens the camera already begins to zoom out again, showing the whole scene for the last time.

Original Storyboard

Modelling & Animation CA : Idea: Board Game with realistic figures

Cut Number: 1 **Timing:** not sure **Action:**



Techniques:
fractals
camera
L-Systems

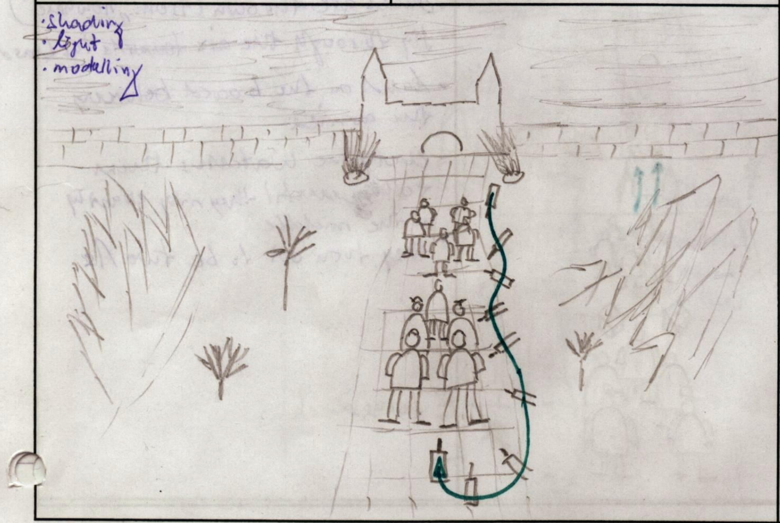
- The board is introduced
  - mountains
  - trees
  - a wall
  - street (field)

  } scene is gloomy
- The confronting castles, enlightened in red/blue are shown

At last we see the two armies
- close camera work
- camera gets faster
- (spectators understands aim)
- trees are moving

? time, tips

Cut Number: 2 **Timing:** not sure **Action:**



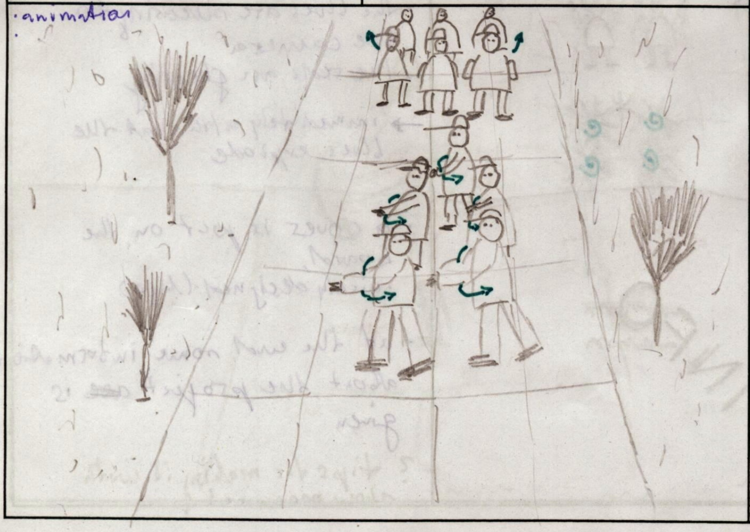
·shading
·light
·modelling

- the armies are introduced

→ red/blue → facial expressions
→ cool
→ as lively as possible
- the spectator understands the aim
- the armies walk/jump towards each other
- they stop in front of each other abruptely
- the camera goes along them and stops behind the blue team and stays there for a while

? robotic, human, jump/walk templates for guys

Cut Number: 3 **Timing:** **Action:**

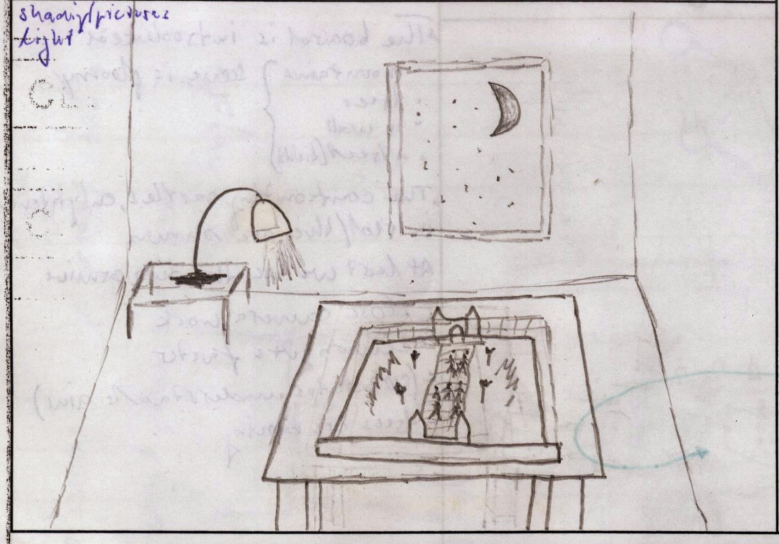


·animation

- the blues turn ~~suddenly~~ around, the reds took up and both armies look in the camera
- maybe some special animation (sunglasses put off), facial expressions
- fast motion (surprise!)
- after some time fast zooming out

? difficult animation

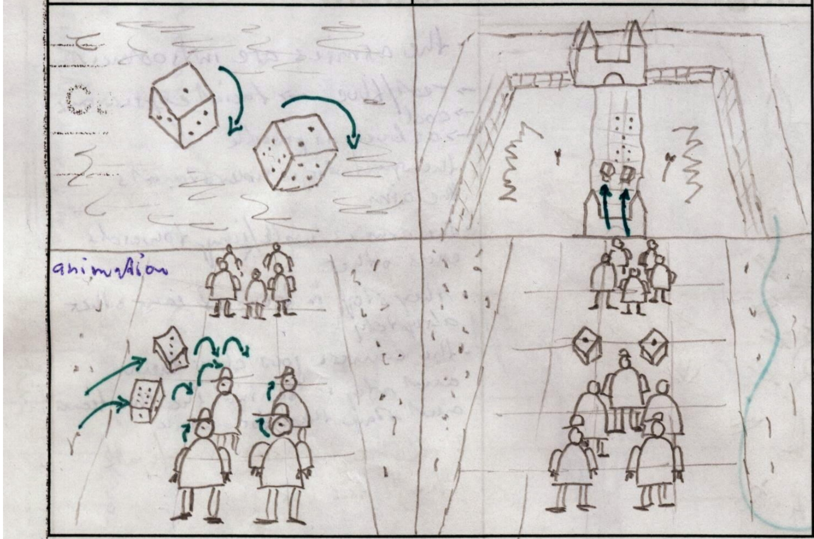
Cut Number: 4 | **Timing:** | **Action:**



- after the camera has zoomed out quickly, the spectator understands
- the room is introduced
→ a lamp (or more) on table
→ window (moonshine, stars)
→ the board on a table
It shouldn't be zoomed out too much, just until the room is recognisable
- the atmosphere is dense

? out of window picture/modelled light should come from there

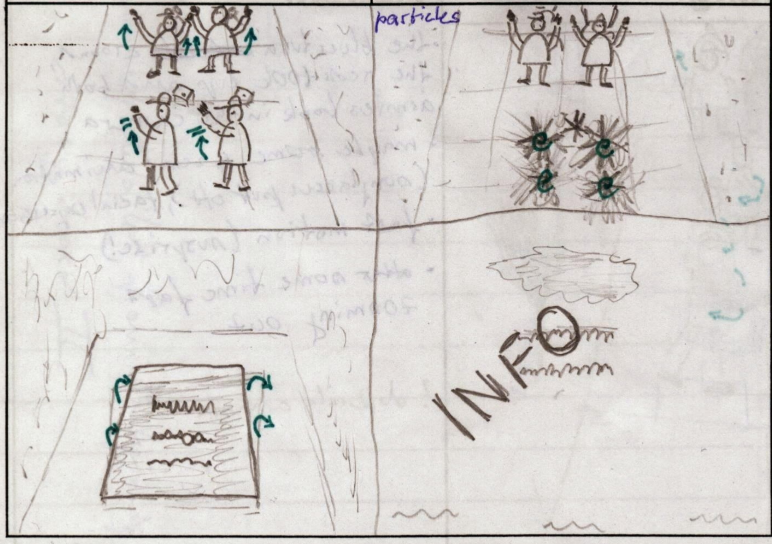
Cut Number: 5-8 | **Timing:** | **Action:**



- dices are thrown (from „nowhere")
- fly through the air towards the board
- land on the board between the armies
- everyone watches them roll until they stop exactly in the middle
- they turn out to be two 1s

? focus, dark

Cut Number: 9-12 | **Timing:** | **Action:**



- the blues are sneasing to the camera
- the reds are cheering
→ immediately after that the blues explode

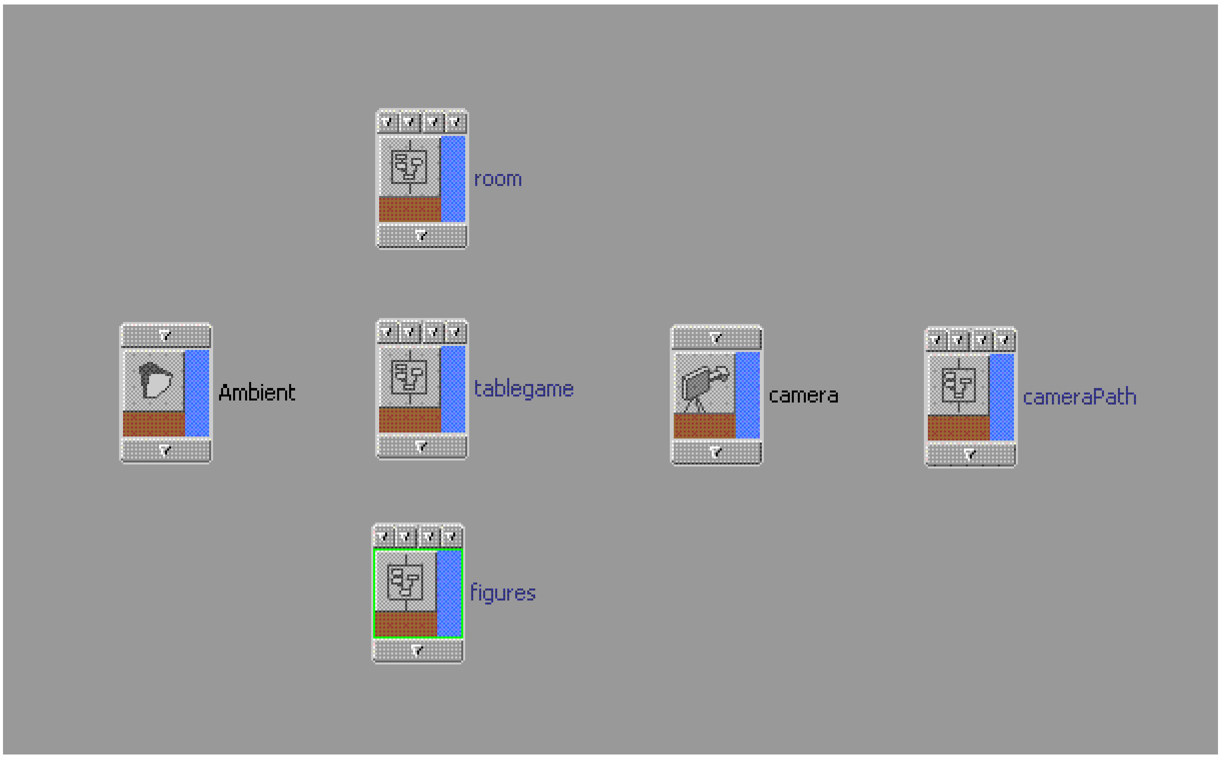
- a cover is put on the board.
: nicely designed (logo)
- at the end some information about the project ~~are~~ is given

? tips for making it with slow machine? (

Hierarchical Structure

The Hierarchical Structure separates the whole story into 3 parts:

The story is placed in a room (*room.hip*). A table game is then put on a table in the middle of the room (*tablegame.hip*). Animated figures and dices (*figures.hip*) are finally included in the table game. Most modelling, shading and basic animation was made in these files separately. After that, the files were merged and a camera, lights and more complex animation was included.



Technical Description

In this part sufficient information about used techniques is given. This includes *Modelling, Lighting, Shading and Animation*.

Frequently used techniques:

Following techniques help to recreate the described model.

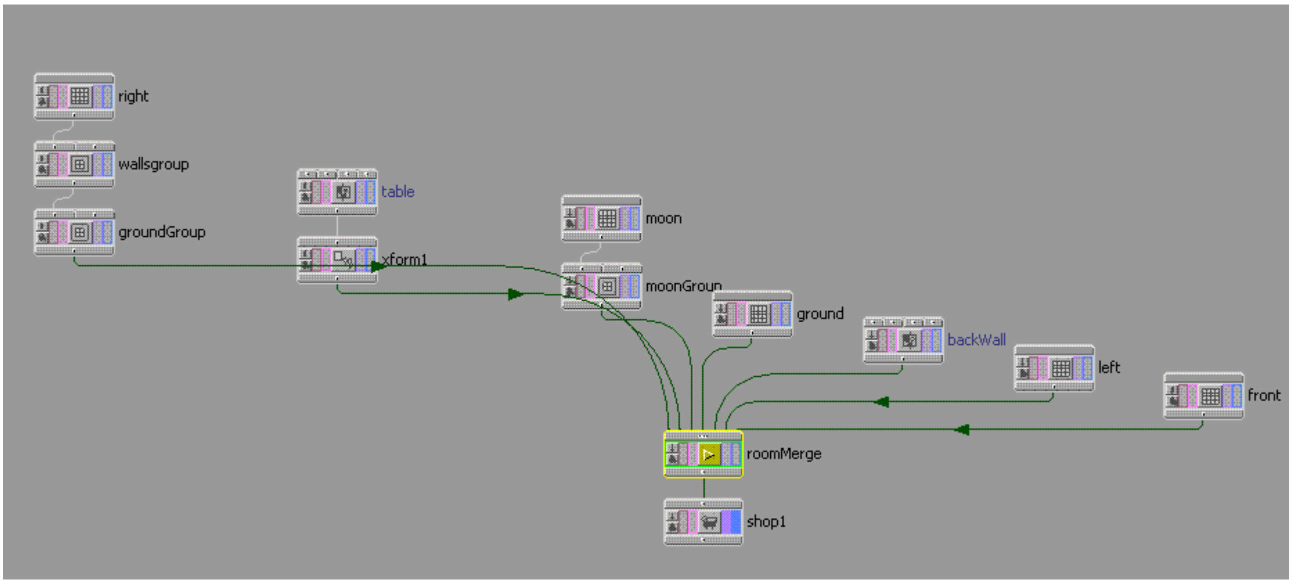
During the entire design emphasis is put on clarity and understandable structure. This is done by consequent partitioning. Therefore not only the main scene is divided into three main pieces (*room, tablegame, figures*) but *Subnetworks* are used throughout the design.

The design contains mostly symmetrical models. Modelling one half of the model and then using the *Mirror Operation* saves a lot of work and again makes the design more clear to others.

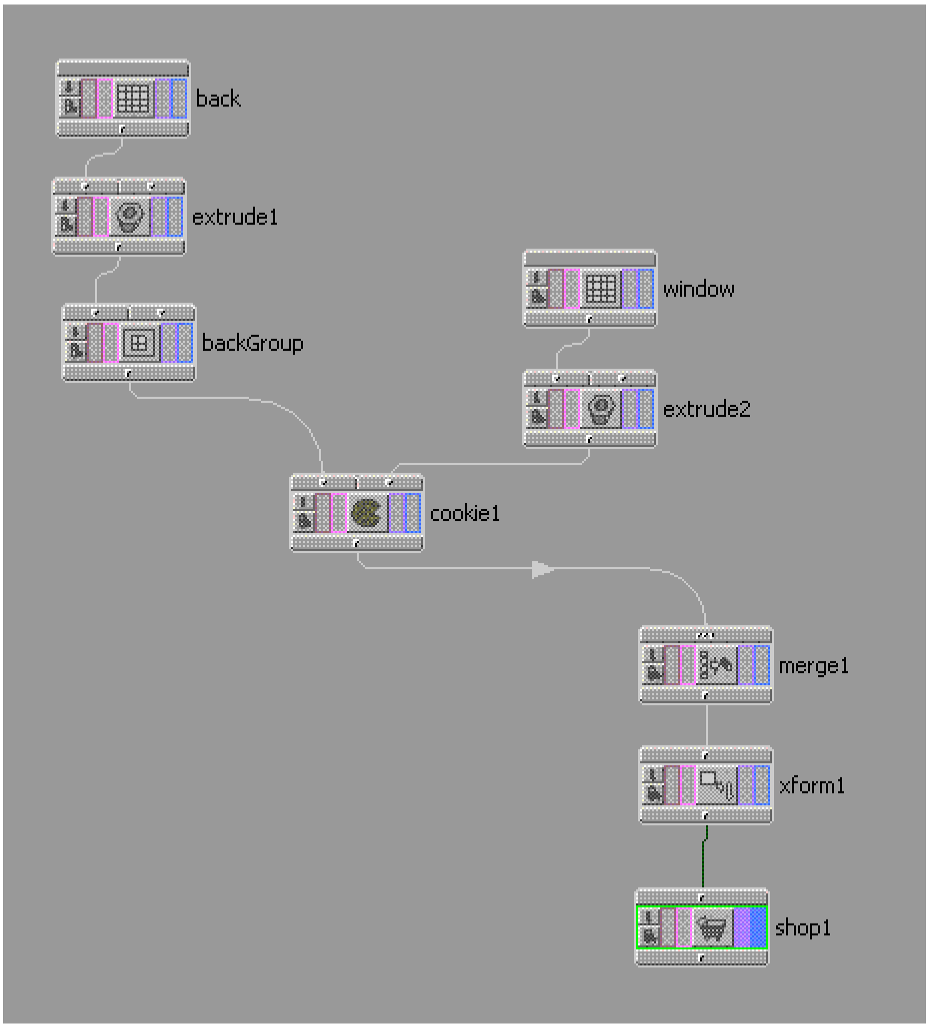
Combining the Construction Plane with Snapping is also an useful way of modelling. By that, paths, bones etc can easily be aligned.

Due to simplicity most models get the same dimensions at the first place. Finally, when they are put together, the appropriate *Scaling, Transformation and Rotation* is set.

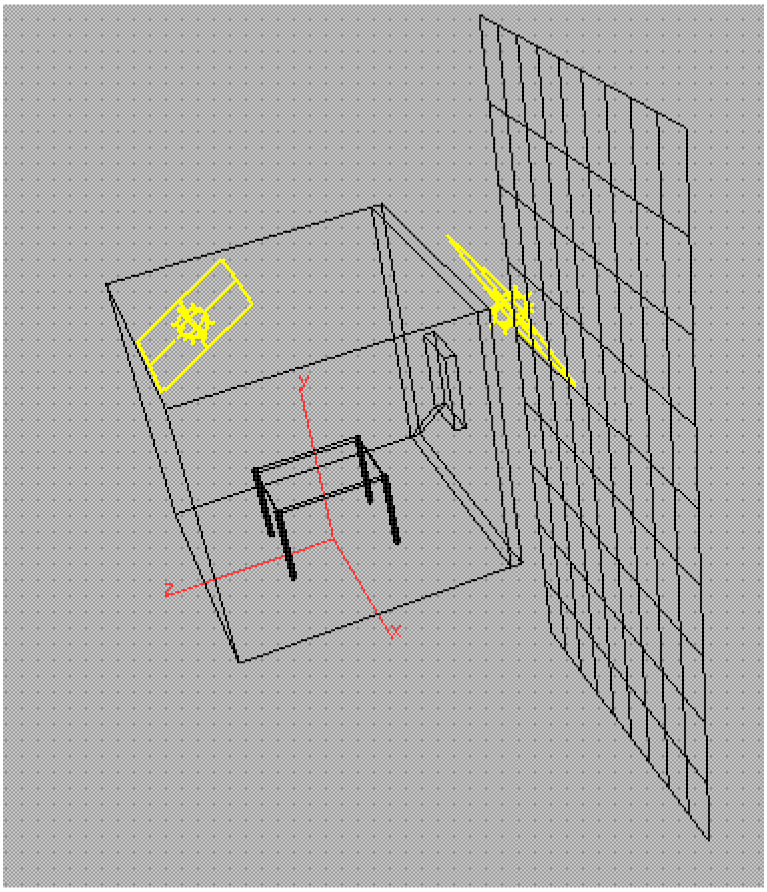
Room



Modelling: The window of the room is made by a *Cookie Operation* by which the window is cut out of the back wall (*A minus B*).

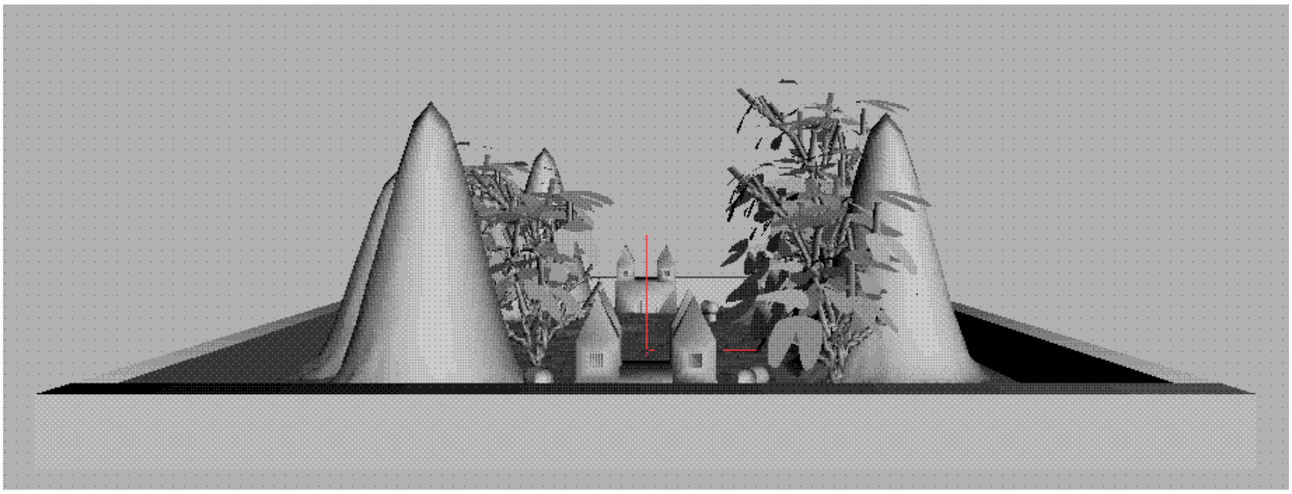


Light: Two lights are included. The first one to simulate the moonlight coming from the window. The second one coming from the other side of the room in order to have some additional light coming into the back of the blue figures.



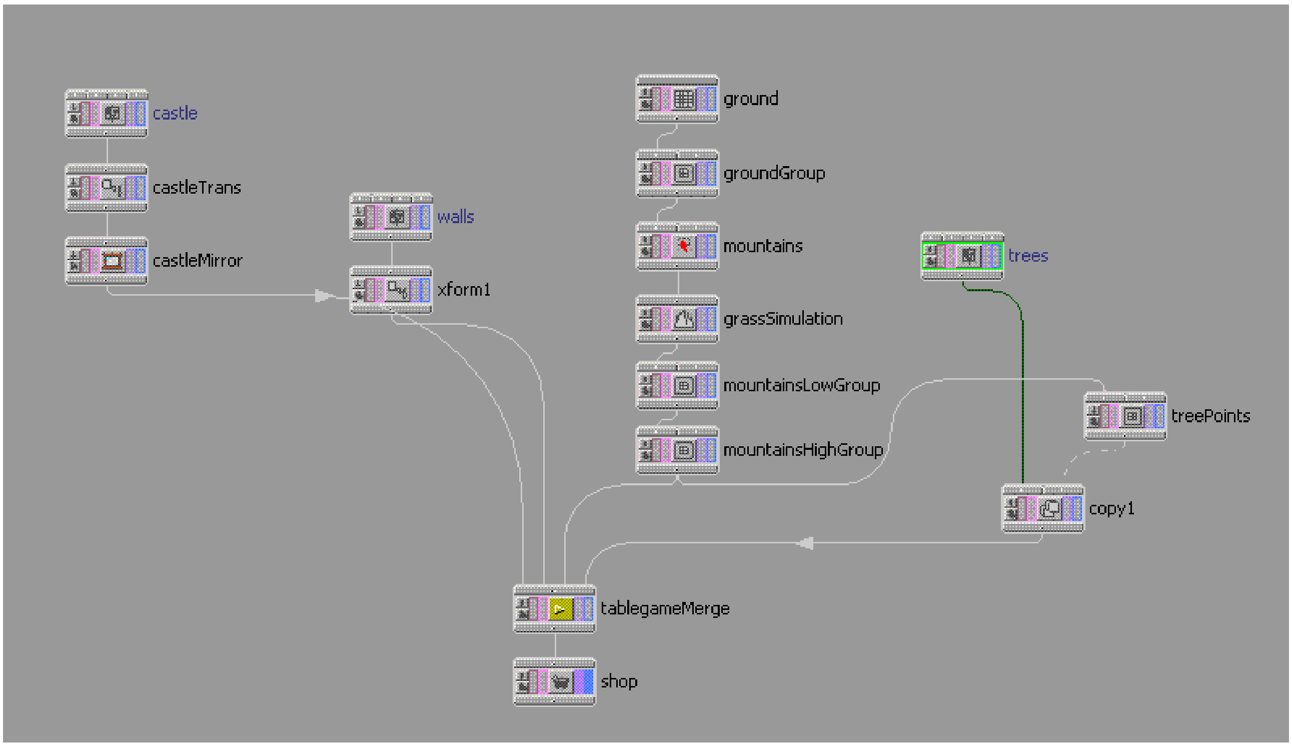
Shading: The grid placed outside the window gets a *VEX Galery Spots Shader* to simulate a dark sky full of stars.

Tablegame

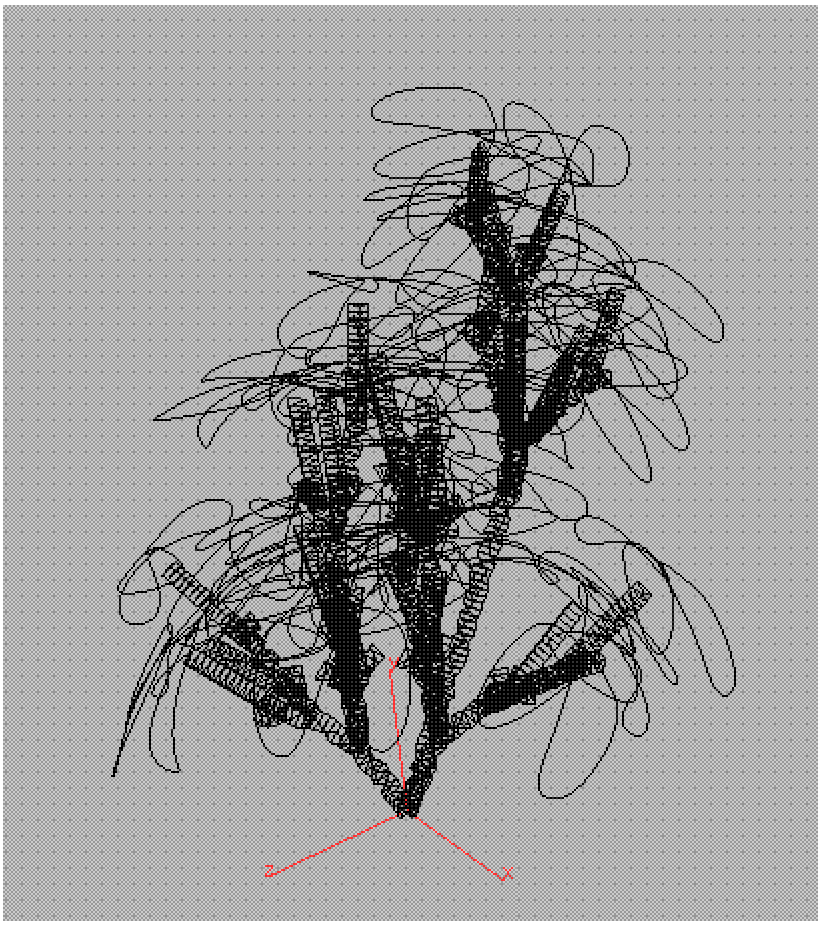


Modelling: An important part of the table game is the ground. A *Filter Operation* creates some high mountains and a low *Fractal Operation* makes it look like a grassy area. Trees are put on a group of points of the ground. Castles are built at both ends of the table game.

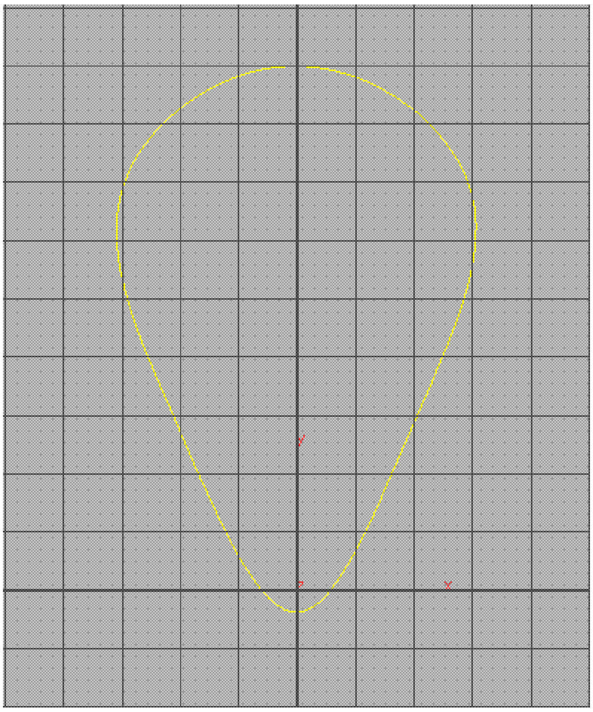
Shading: By using a *VEX Galery Splatter Shader* the ground gets a realistic brown/soily and green/grassy look. The mountains get an extra white/snowy shader on the top and a grey/stony shader in the middle.



Tree

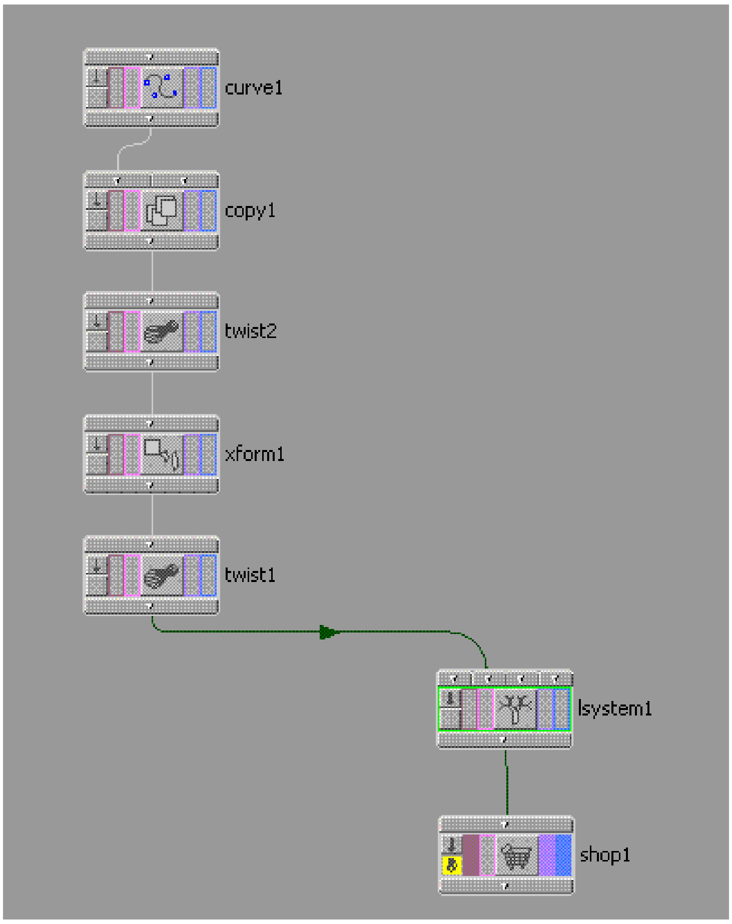


These trees are made by an *L-System*. It simulates a mixture of bush and tree with big leaves. These leaves should hide the walls of the table game. To avoid a too crowded tree just every third branch gets a leaf. The leaves are bended to look more realistic.



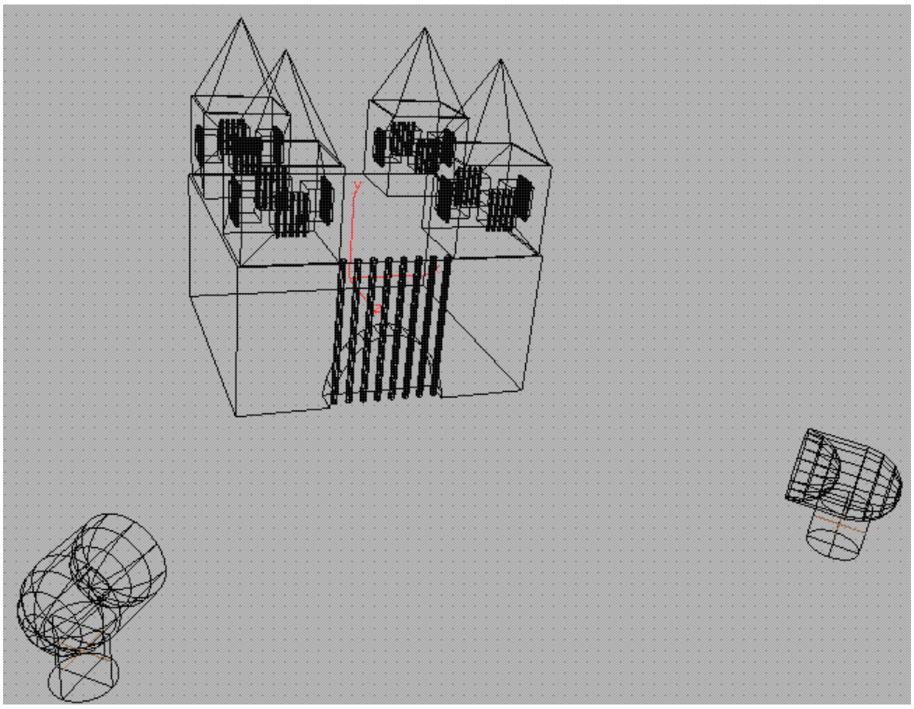
L-System Rules:

- Premise: A <= Startpoint of the L-System
- Rule 1: A=[&FFFFA[fK]]/////[&FA[f]]///////[&FA[f]] <= branching into three branches, split widely apart to cover a big area
- Rule 2: S= F <= to get a smooth ending

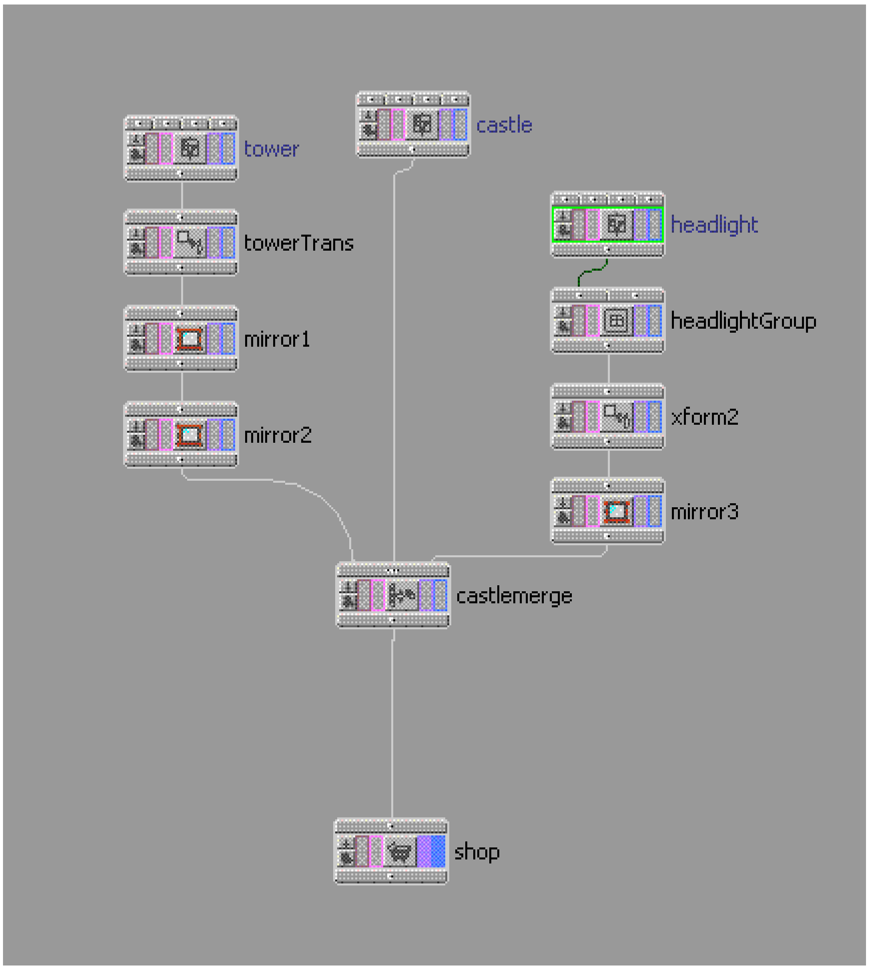


Shading: A *VEX Galerie Wood Shader* is given to the branches.

Castle



Modelling: A castle is made of the main castle, four towers on it and two headlights. Each part is designed separately in a subnetwork. The castle and towers get windows by a cookie operation.

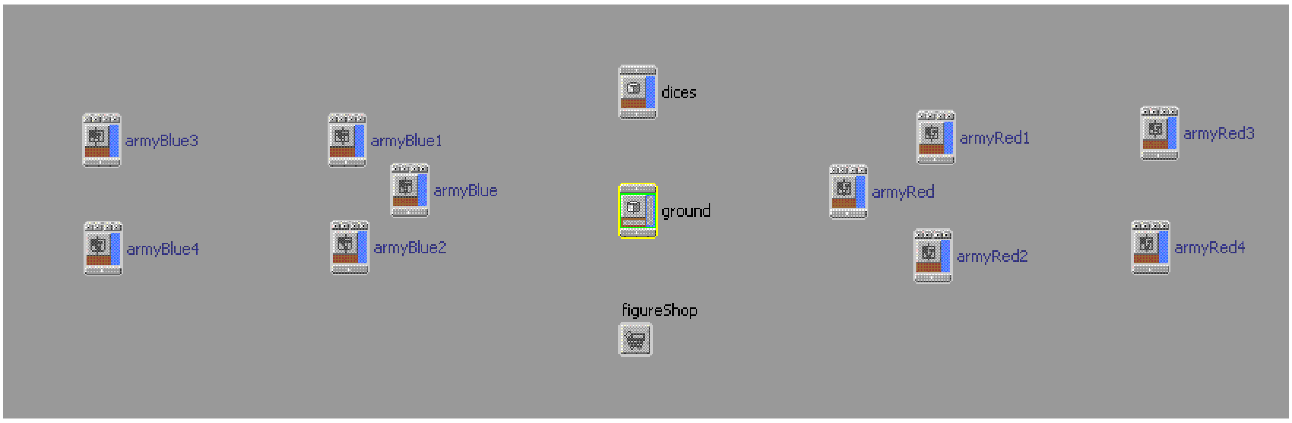


Shading: A *VEX Metal Shader* makes the lattices look authentic. Inside the windows a totally dark *VEX Plastic Shader* is attached to simulate darkened rooms.

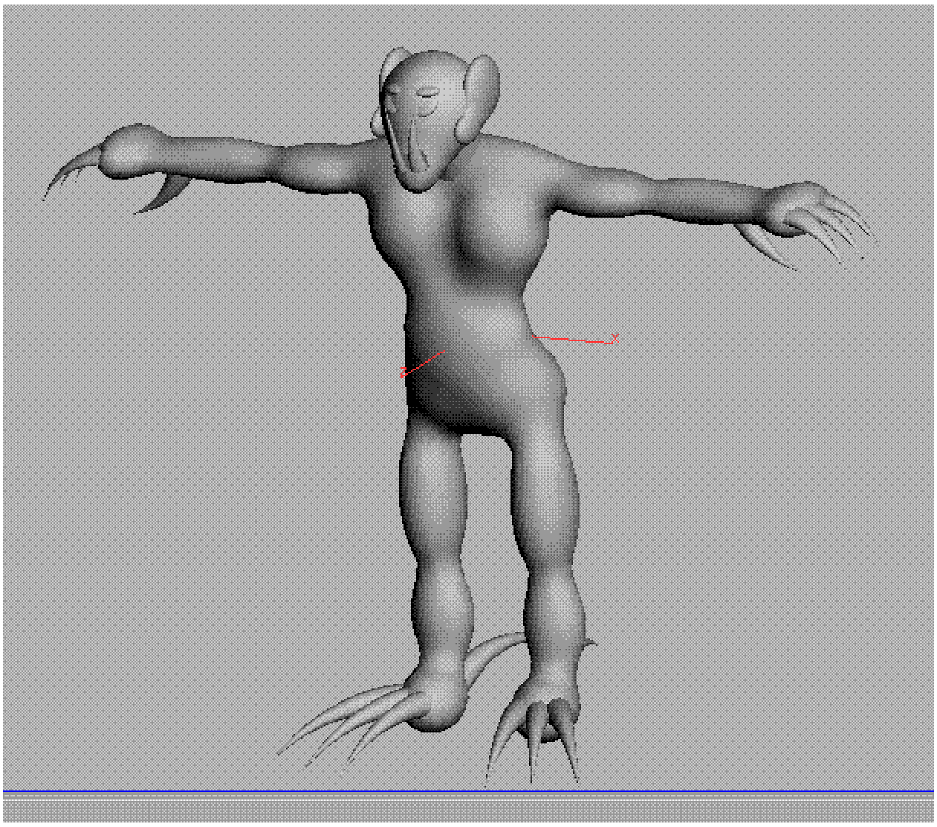
Light: Inside each headlight a light is insert with a red/blue colour depending on the position of the castle.

Figures

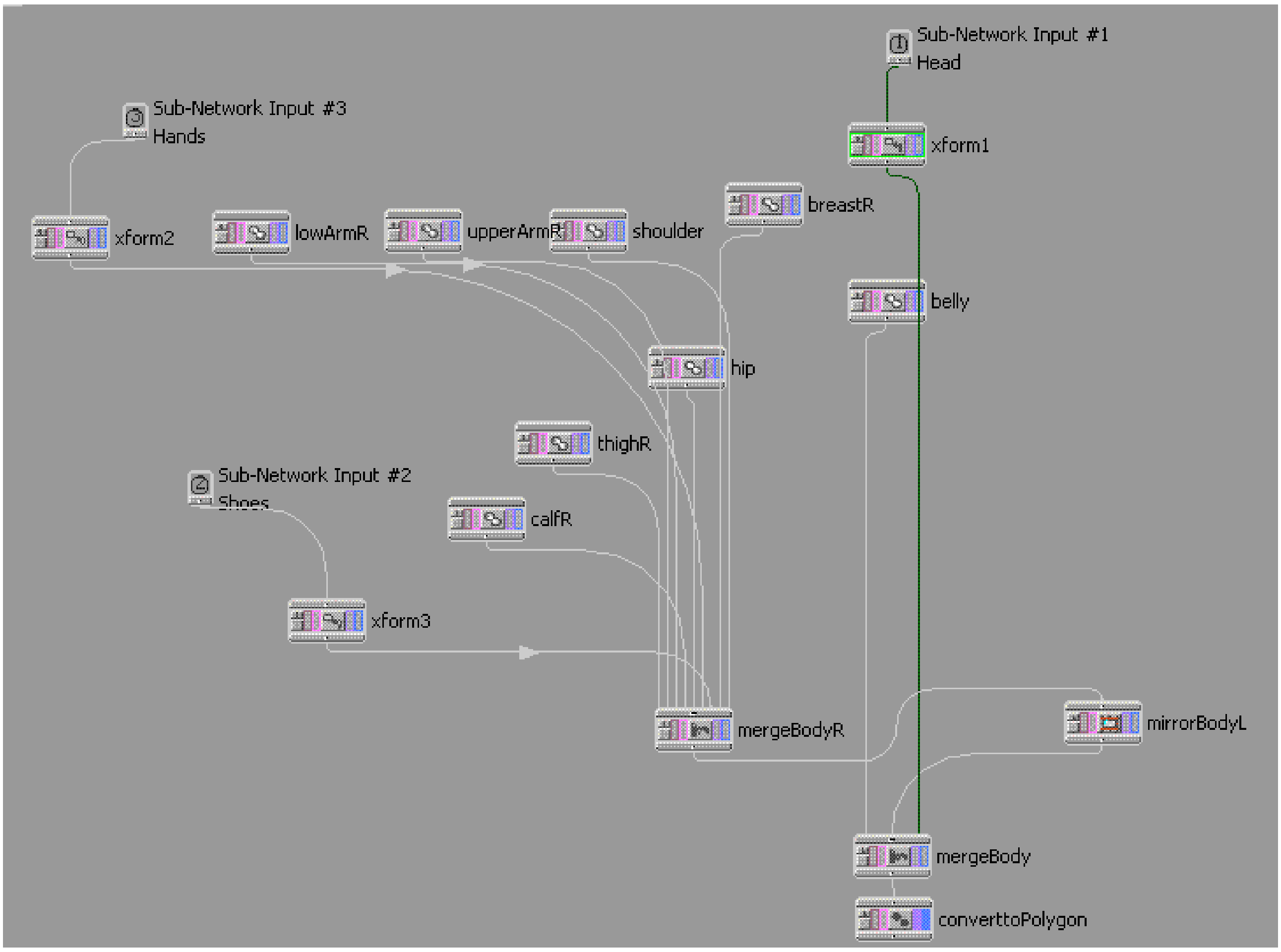
One figure is modelled and animated and then copied and slightly changed to fill the armies. The dices are modelled and animated.



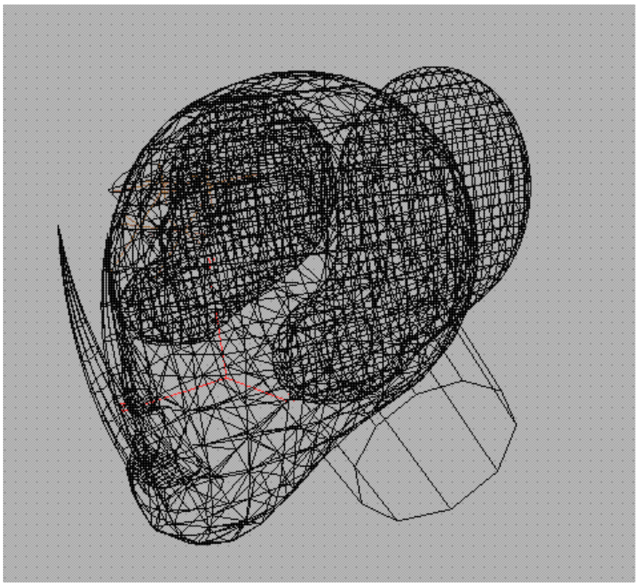
Figure



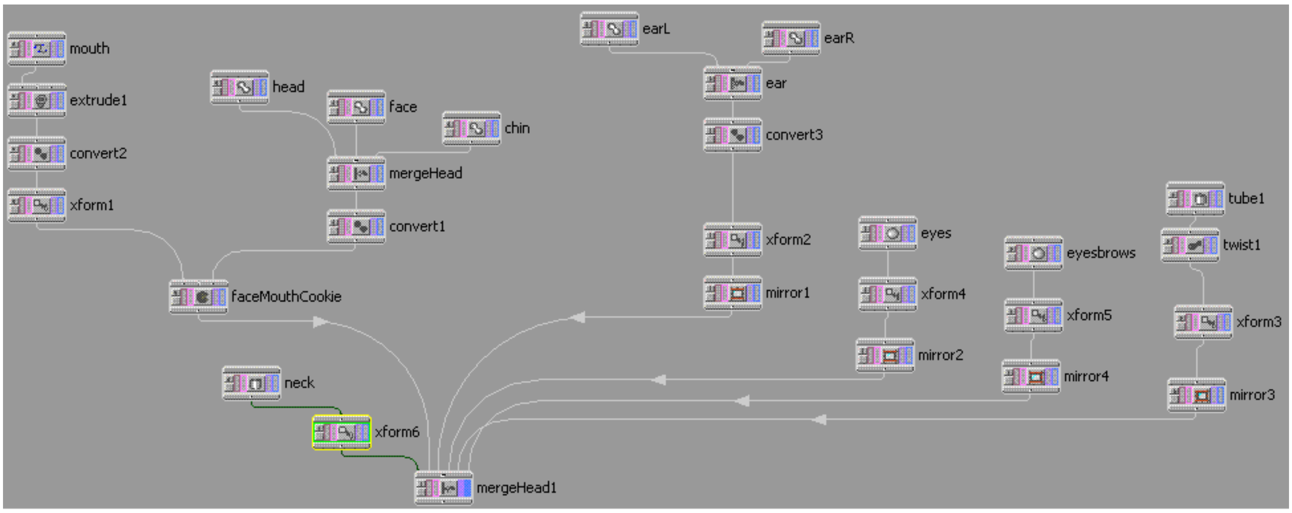
Modelling: A figure is made of a Body, a Head, Feet and Hands. One side of the Body is modelled, the other part is mirrored. The body consists of Belly, Calf, Thigh, Hip, Breast, Shoulder, Upper arm and Lower arm, all modelled as *Meta Balls* of different size. By using them it is possible to simulate realistic muscles. These meta balls are converted to polygons to improve the performance. The arms are designed hold up to make it easier to insert the bones.



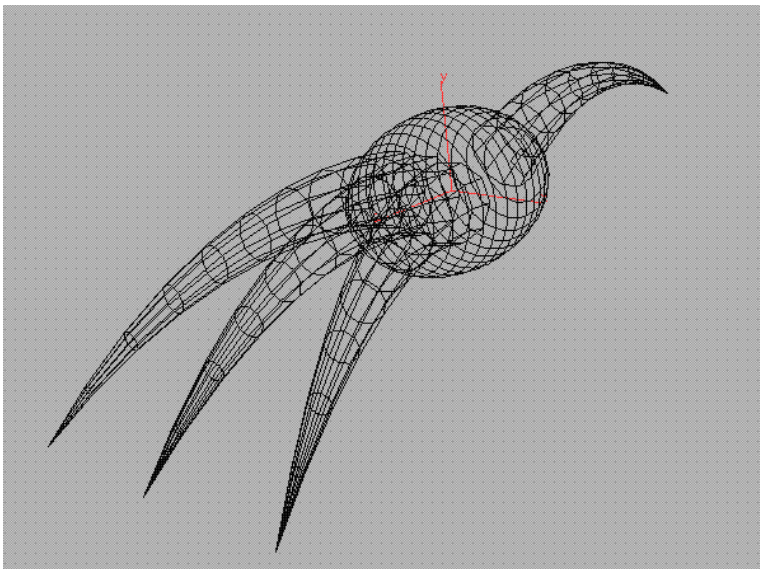
Head



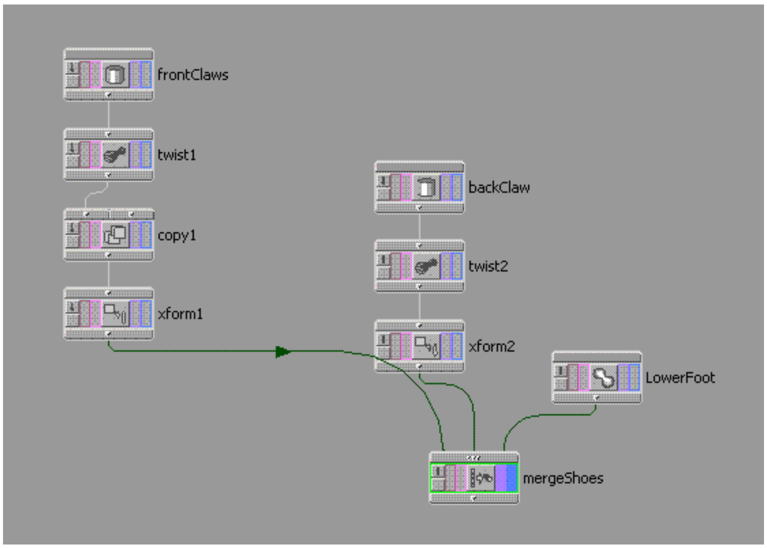
Modelling: Chin, Head and Ears are made of Meta Balls. The Mouth is cut into the face by a cookie operation. The claws are bended tubes.



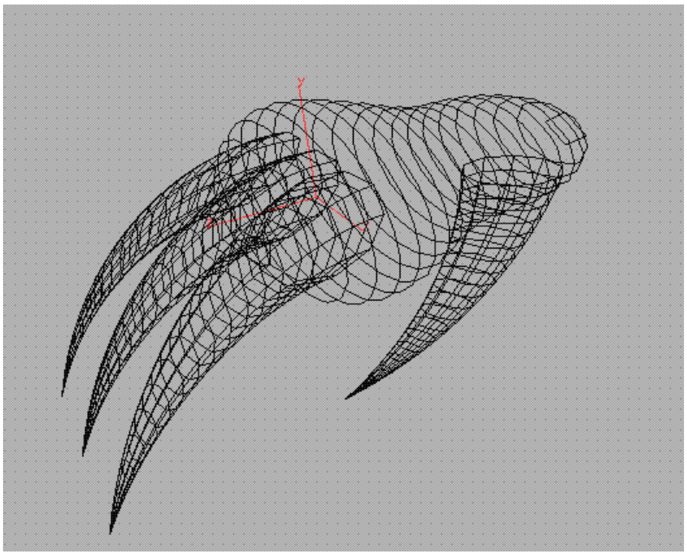
Feet



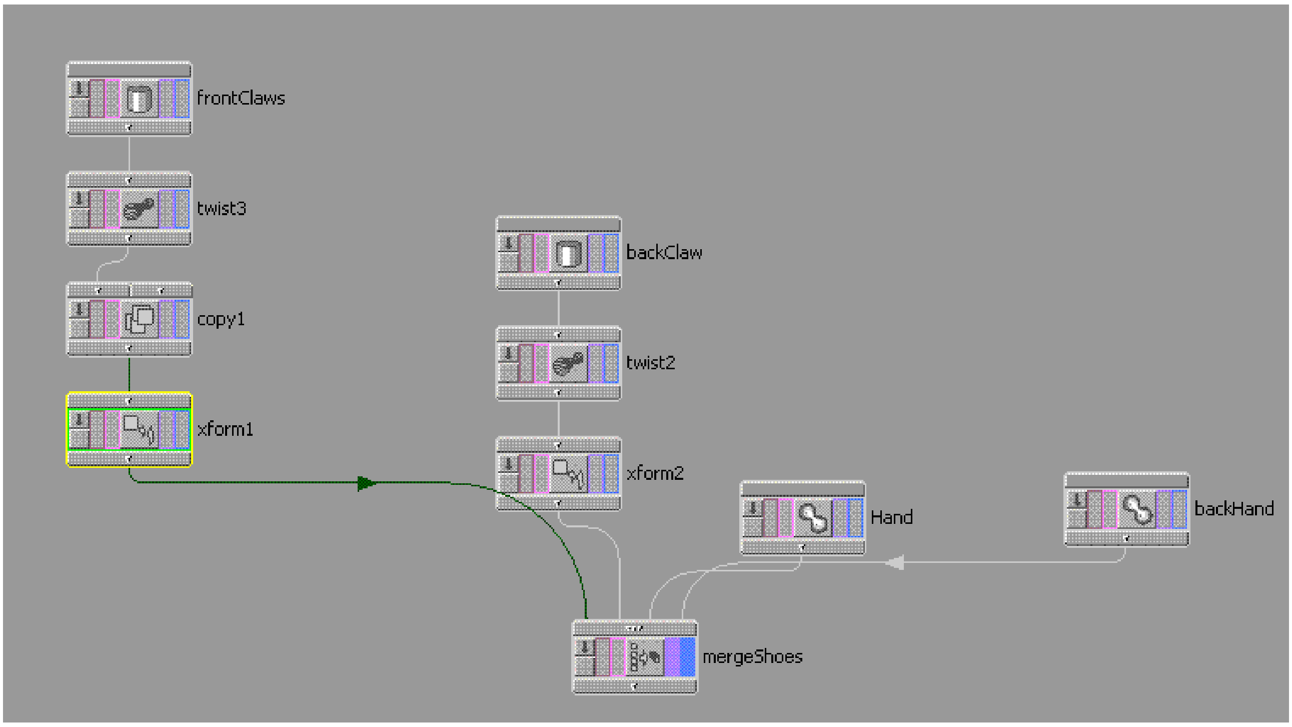
Modelling: The foot is made of a *Meta Ball* so that it sticks to the end of the calf. The claws are bended tubes.



Hands

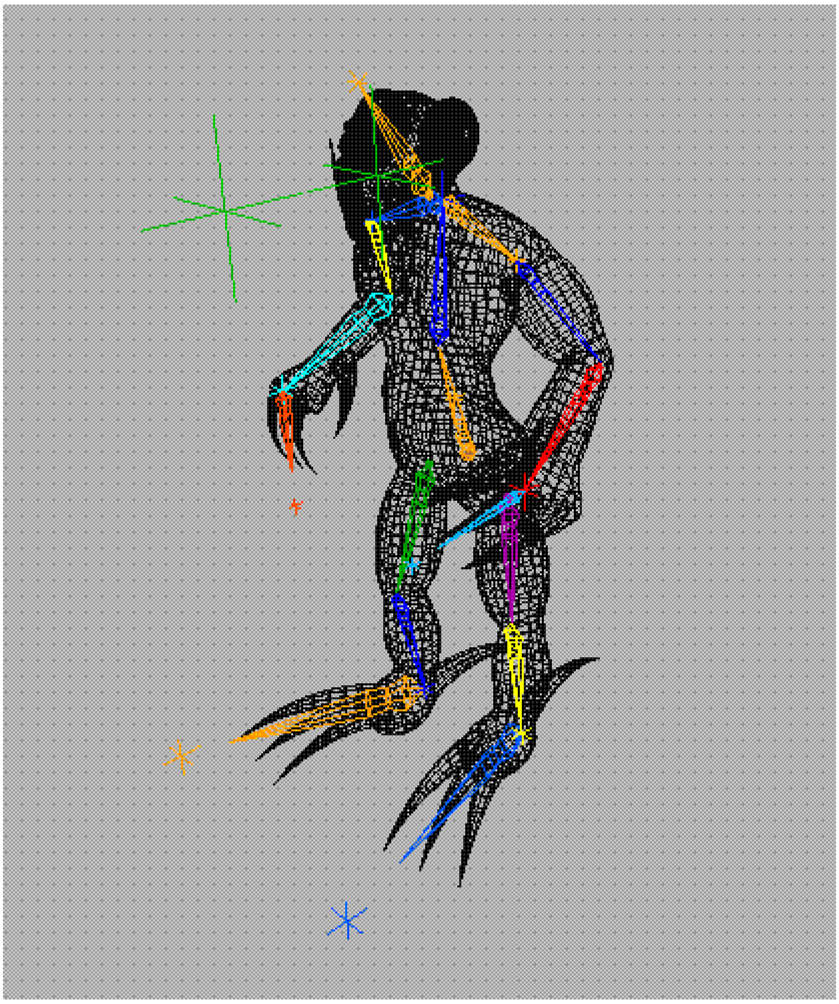


Modelling: Consisting of two *Meta Balls* so that it gets a proper form and sticks to the end of the lower arm. Bended tubes form fingers and a thumb.

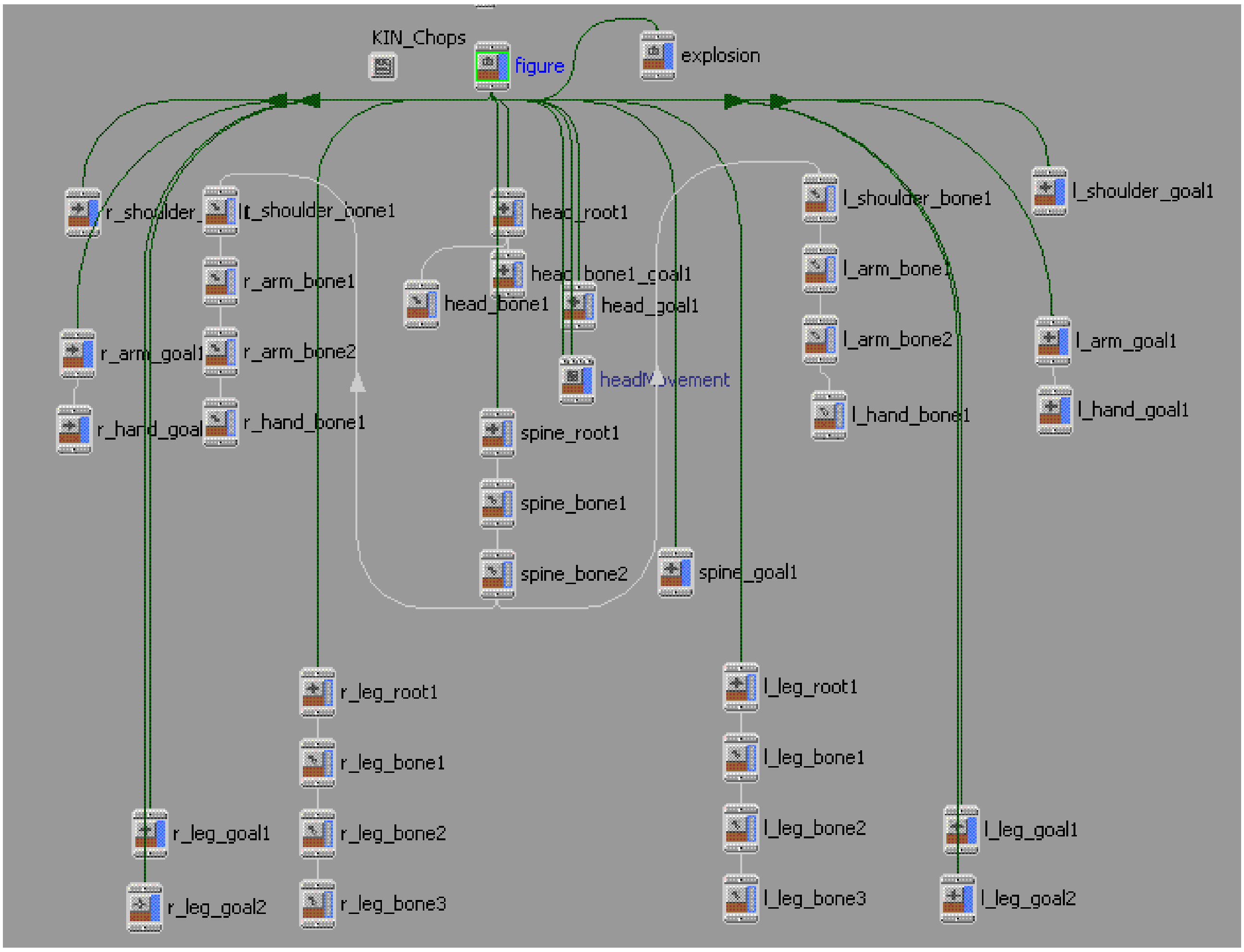


Shading: Depending on the position of the figure a red or blue shader is added. Later, the blue figures disappear in a growing ball. A VEX Super Material Shader with a red Diffuse Colour and low Alpha Perp and Roll values is attached.

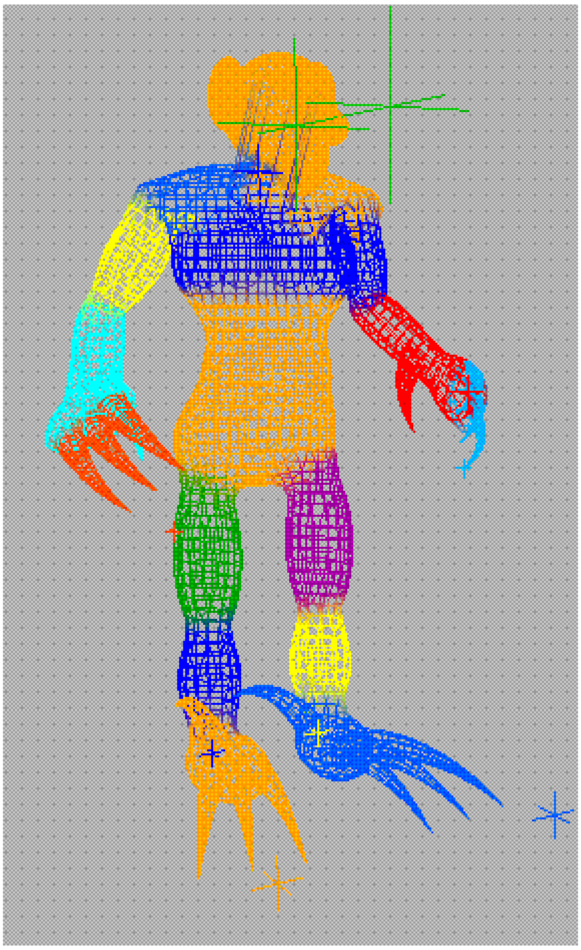
Bones



Chains of bones are inserted. The lower spine and upper spine bones provide the possibilitiy of a proper turning. The head bone and root make the head mobile. A shoulder, upper arm and lower arm bone fill each arm. At the end of the arm is an additional bone attached for moving the hand. Thigh, calf and foot are also filled with a bone, respectively, in order to make walking possible. Except from the head bone every bone has got *Inverse Kinematics*. By changing the position of so called *Goals* at the end of the arms, legs, hands and feet, the bones change their position.



To get the limbs move with the bones, the Geometry has to be captured to the bones.



Animation

In order to make the figures walk, the feet, legs, shoulders, arms, hands and the head have to move periodically. When on one side the leg and foot are moving forwards, the shoulder, arm and hand have to move backwards at the same time. This periodical movement is achieved by a Sinus.

Formular: ***sin(frequency*phase)*amplitude***

Frequency: It defines the pace of the movement. A pace of *$F times a number between 12 and 15* is used.

Phase: It modulates the frequence. As it was not useful with a walking animation a *1* was used.

Amplitude: It defines the amount of movement made by the formular. A *number between 2 and 4* is used.

The anatomy of the figure made it difficult to let the figure look at a certain object. Finally, orientating the head was animated by three different objects:

ToSee Cross: The figure shall look to that position. The TooSee Cross has an X,Y and Z Value (*ToSee-X, ToSee-Y, ToSee-Z*). By changing its position its X,Y and Z values change.

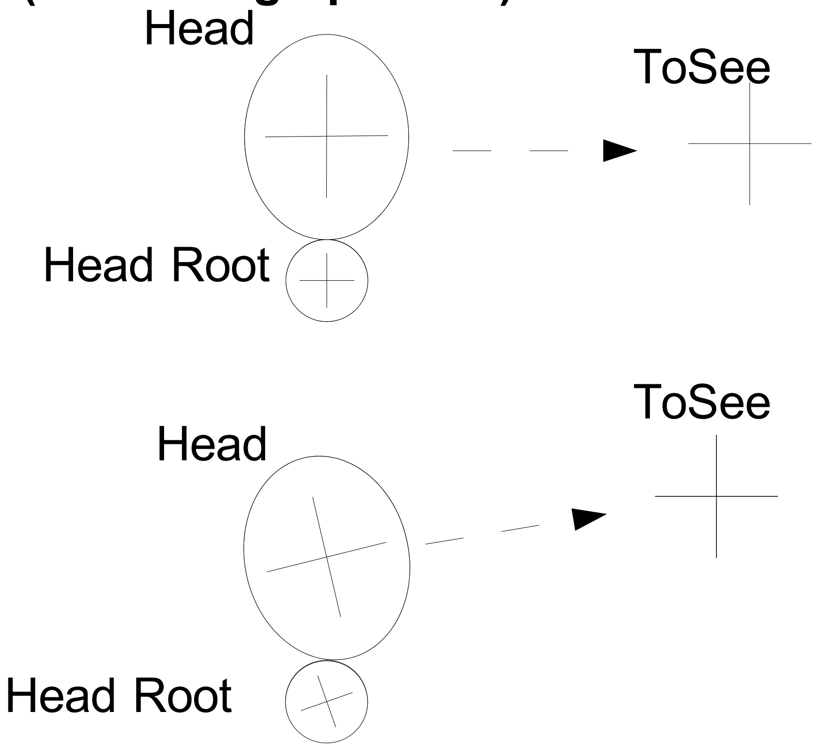
Head Cross: The *Head Cross* changes its *Rotation* depending on the position of the *ToSee Cross*. The Y-Rotation is set to ***atan(toSee-Z/toSee-X)***. By that the *Head Cross* is turned up or down depending on the *ToSee Cross*. The Z-Rotation is set to ***atan(toSee-Y/toSee-X)***. By that the *Head Cross* is turned left or right depending on the *ToSee Cross*.

Head Root: Finally, the *Head Root* gets the same Rotation Values as the *Head Cross*. By that, the head of the figure turns both up/down and left/right towards the TooSee Cross.

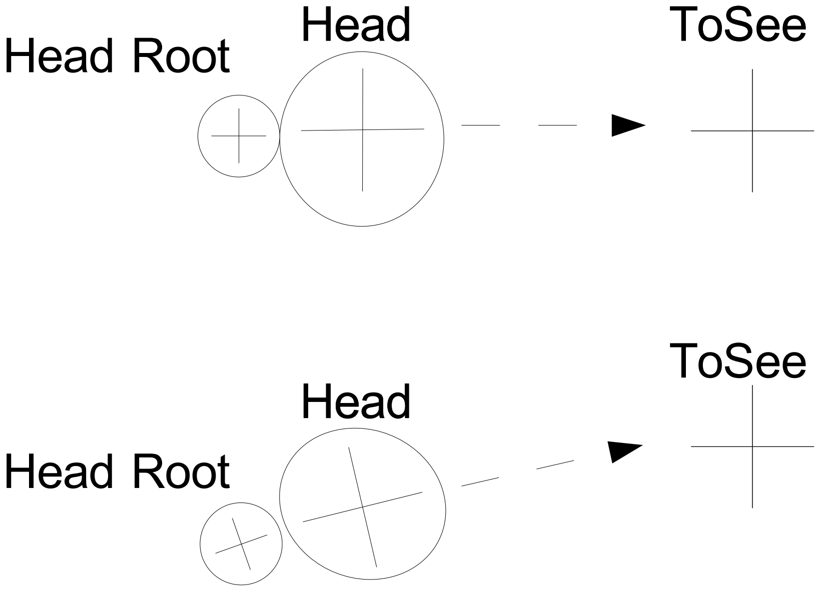
By positioning the *ToSee Cross* at the camera or the dices the figures seem to look at them.

This is explained by following drawing:

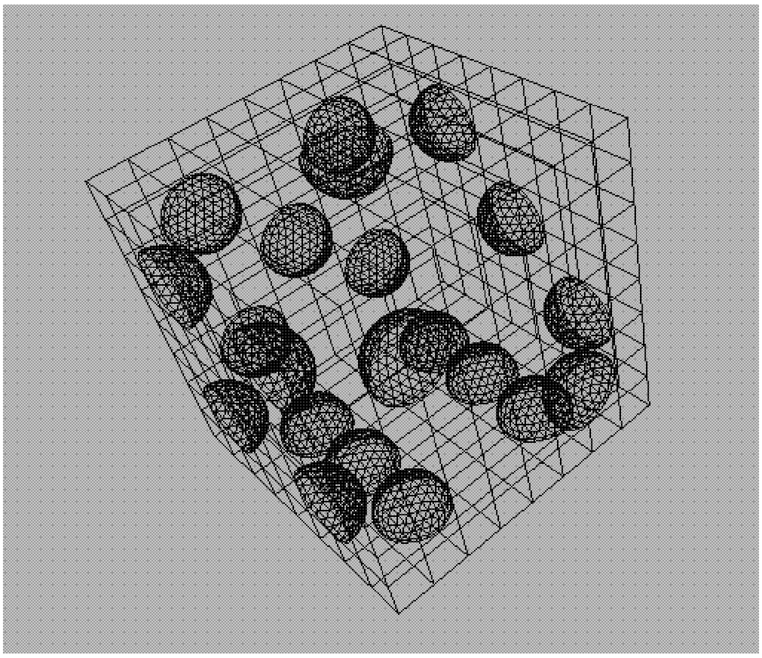
Head Movement from Right Side (for turning up/down)



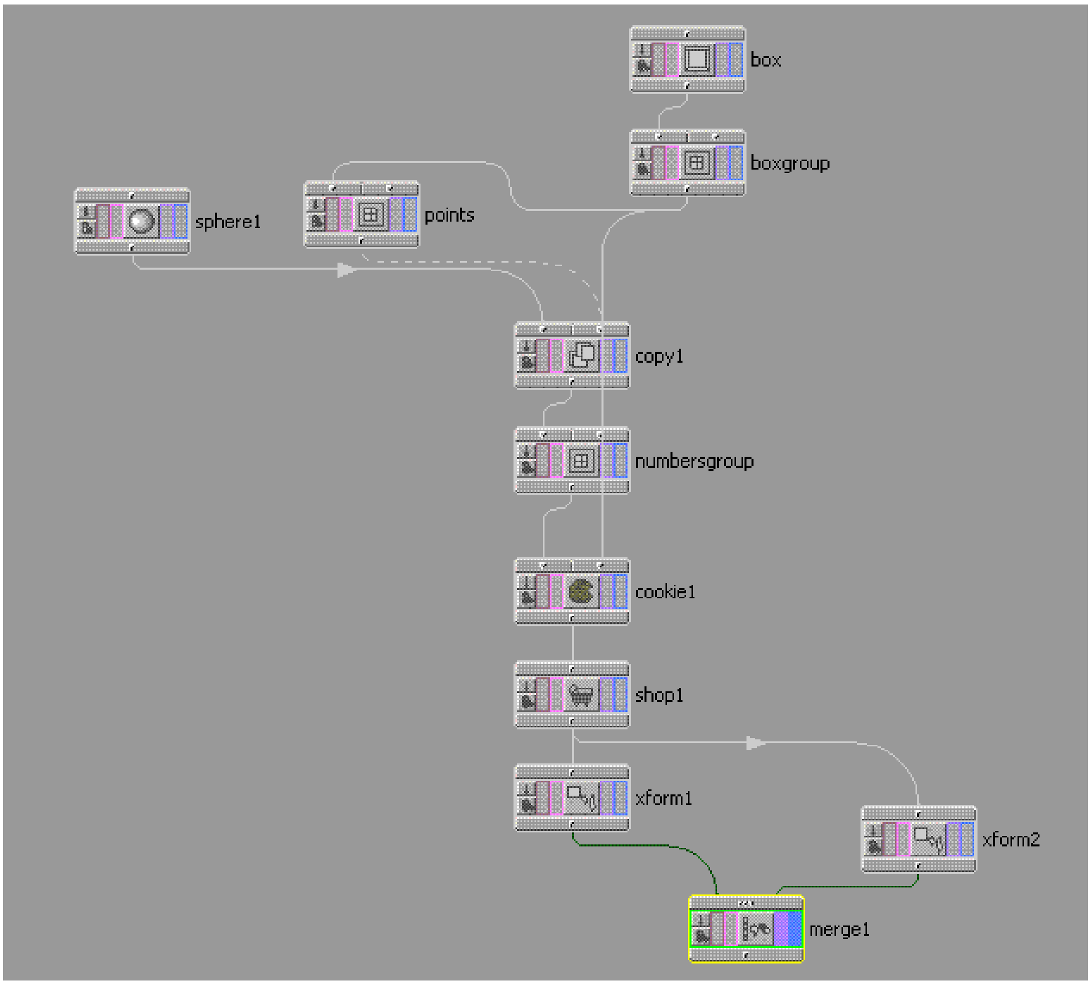
Head Movement from Top (for turning left/right)



Dices



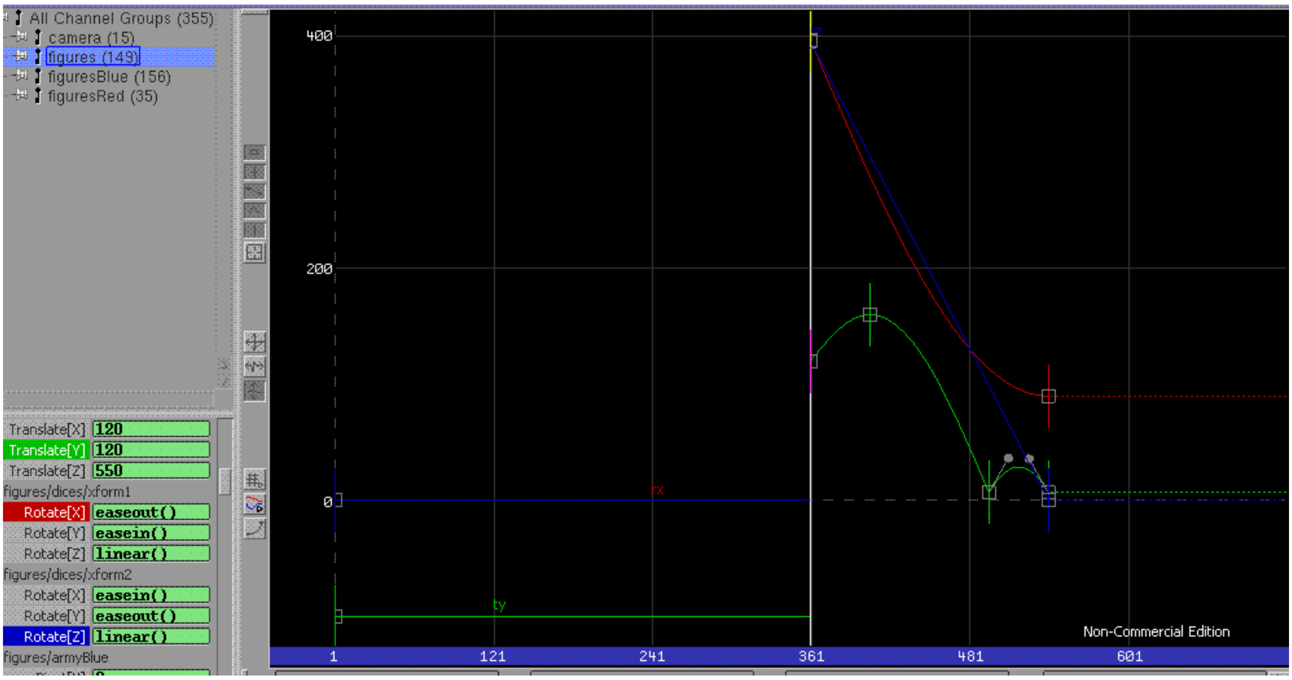
Modelling: One dice is modelled and then copied to form the other dice. By a *Group Operation* the number points on the dice are selected. Spheres are then copied to these points. Finally a *Cookie Operation* subtracts the numbers from the dice.



Animation

The dices move through the scene by keyframing. A rising and falling is made by three keyframes connected to each other by a *easeout()* and *easein()* function to make a smooth transition. The landing is animated by two keyframes connected by a *Bézier* curve to make the dices hit the ground twice.

The dices are constantly rotated in order to finally turn out to be two *1*: Each dice has got various channel functions attached to it (easein(), easout()) in order to give them a different rotation speed.

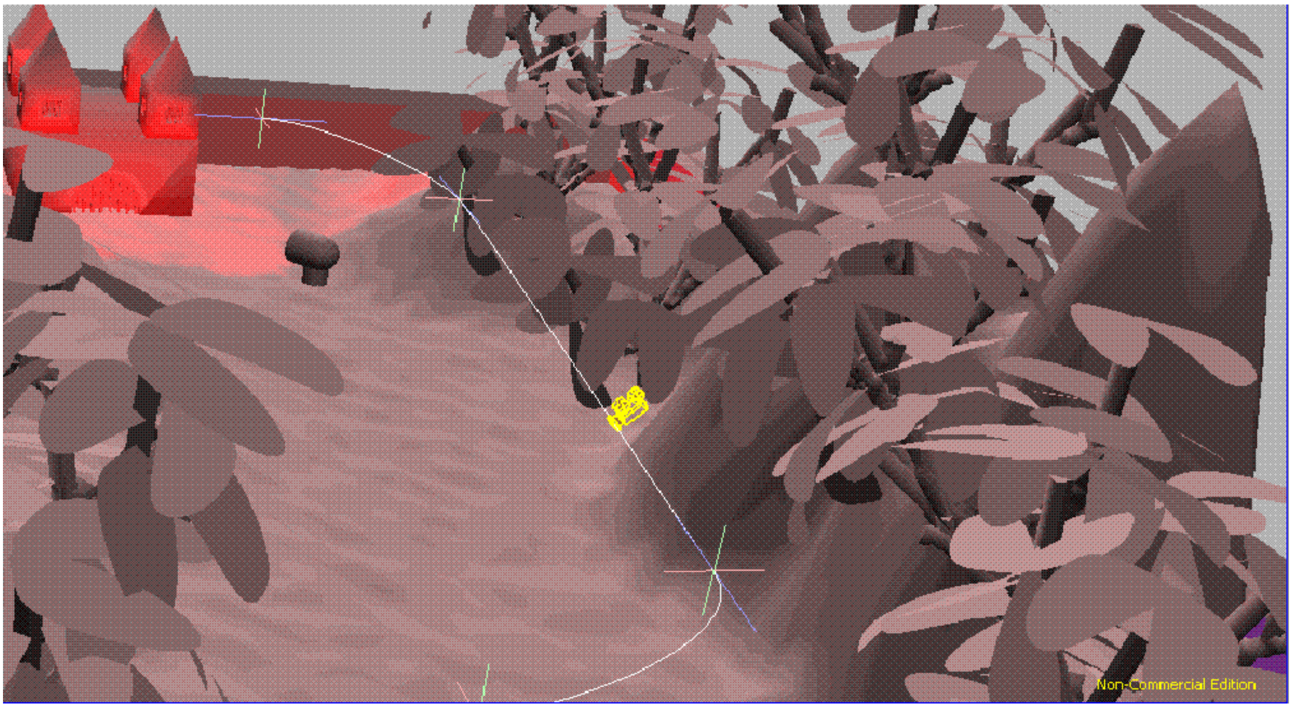


All Channel Groups (355)
camera (15)
figures (149)
figuresBlue (156)
figuresRed (35)

Translate[X]	120
Translate[Y]	120
Translate[Z]	550
figures/dices/xform1	
Rotate[X]	easeout()
Rotate[Y]	easein()
Rotate[Z]	linear()
figures/dices/xform2	
Rotate[X]	easein()
Rotate[Y]	easeout()
Rotate[Z]	linear()
figures/armyBlue	

Camera

The Camera Movements are done by a *Camera Path* and *Keyframing*. The suitable Camera Path is drawn using the *Construction Plane*. The position of the camera on the path is then keyframed to make the camera go along the path. Several keyframes along the path determine what the camera is looking at and also what the camera is not looking at as it is supposed to overlook the fact that the figures are on a table game.



Keyframing

Other animation is made by keyframing. In following timeline the most important keyframe sections are listed and described.

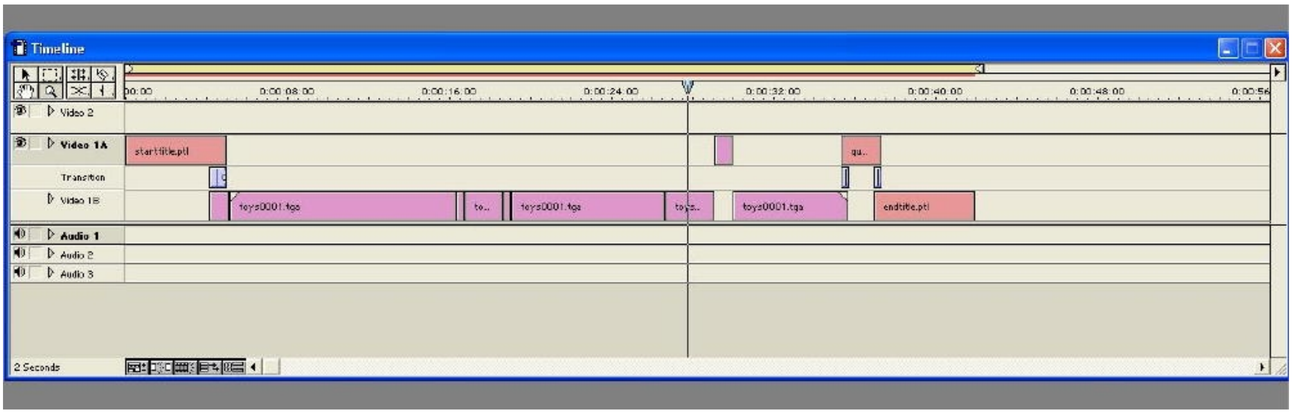
1-270	*271-315*	*316-360*	*361-540*	*541-585*	*586-720*
The armies walk towards each other.	Blue figures turn around and look into the camera (*2x keys*). Red figures look up into the camera.	The camera quickly zooms out until the room, the window and the whole table game can be seen.	The dices fly through the air. First they go up and then down towards the ground (3x keys). They are rotating in a different speed to each other. They hit the ground twice before turning out to be two 1s(2x keys).	The red army cheers (2x keys). The blue army stamps on the ground to show its temper(2x keys). The blue team also looks into the camera.	The blue figures shrink.
The camera is going along the path looking at the landscape, the castles and the armies (6x keys).			The camera is zooming in to follow the dices. It zooms until it gets to a position where it focus both armies and the landed dices (3x keys).		A ball inside the blue figures gets bigger and then starts to shrink (2x keys).
			The figures look at the dices and watch them falling and landing.		The camera zooms out.

Rendering

The Rendering, which was done with PAL 50% (384x288), took 17 hours on the computer and contained 253MB of raw video data. This either shows how complicated the rendering was or how slow my computer. I have to mention that in general it is more than recommendable to have a *High End Machine* for working with Houdini.

Postproduction

The Postproduction is done with Adobe Premiere. Titles at the beginning and the end are included as is a slogan for the final toy. Transitions enable it to change smoothly between them and the video animation. Both after the camera stops behind the blue figures and after they turn towards the camera small stops (00:00:00:10) are added. The reaction of the figures after the outcome of the dices gets more speed (200%). Finally as the rendered pictures turned out to be too dark the *Brightness* (+20) is increased. The Video is then saved as a compressed (DivX) AVI-File.



Illustrations

Attached to the paper hand-in.