Bima Journal of Science and Technology, Vol. 5(3) Jan, 2022 ISSN: 2536-6041

APPLICATION OF NON UNIFORM RATIONAL B-SPLINES AND SOFT BODY DYNAMICS IN THE THREE DIMENSIONAL MODELING AND ANIMATION OF GEOMETRIC SURFACES

MODI, B[1] and TIJJANIB.I[2]

[1]Department of Mathematics, Faculty of Science, Gombe State University, Gombe Nigeria.
email: bmodi@gsu.edu.ng

[2]Department of Mathematics, Faculty of Science, Bayero University, Kano. Nigeria.
email: idrith@yahoo.com

ABSTRACT

A central feature when sculpting, drafting, and painting objects based on the X, Y, and Z coordinate system is three dimensional (3D) modeling. The coordinate system distinctively specifies each point in a plane using a pair of numerical coordinates which forms the basic components of computer graphics and animation. This paper attempts to show the techniques used in modeling a water Seal named Flippy. The technique ranges from the use of primitive surface geometry object and *soft body* dynamics to the *texturing* of the animal while exploiting powerful *shading groups*. Flippy was later animated to play with a textured ball object for a few seconds. Additional features such as the *rendering* of entire *scene* and the modeling of the *floor* object using primitive *polygons* and *lighting* of the scene was applied. Lastly, *ray tracing* was added to give Flippy and the scene floor a reflective *tile* look.

keywords: 2D, 3D, NURBS, Soft body dynamics, Flippy.

Introduction

The application of soft body dynamics to 3D graphic objects is what this paper is trying to achieve. The image could exist in two dimensional (2D) form having just height and width but no depth or in 3D form having depth, height, and width (Modi & Tijjani, 2017) . 3D Modeling improves graphics by giving a model an appealing and yet complex appearance. Animation of 3D models requires the application of complex rendering techniques to object models. Rendering is used when transforming a 2D object form into a 3D object form. For instance, Set lighting is a good feature used for rendering effects. However, the lighting effect is subject to certain environmental settings such as background and atmosphere. The secondary effect of lighting can be simulated using graphic tools. Such graphic tools like Autodesk MAYA 8. 5 (AUTODESK, Autodesk Maya 8.5, 2007) , are used on polysets NURBS object surfaces.

Non Uniform Rational B-splines (NURBS) (Modi and Tijjani 2017) are used in describing 3D shapes within a computer. NURBS is a technical term for *Spline* curve, and the background of NURBS object forms has been comprehensively discussed by Modi and Tijjani (2017) and is a widely used object form for modeling and animation. Apart from NURBS, other object forms methods such as *Loft, Create, Extrude* and the *Revolve* are used as well. The Loft method usually gives higher control (Stahlberg 2004) to the object modeling process, but is less flexible compared to NURBS in terms of ease of use.

The advent of 3D modeling and animation has changed the artistic view of the world today. This is because while a simple, plain and flat 2D image generally may not impact the perspective of a viewer, the realism and dynamism 3D (Sanders A. , 2016) models has improved the way we view market brands and tangible structures such as bottles, cars houses motion pictures and video games etc.

Soft body dynamics refers to a polygonal surface, lattice, wire, or wrap deformer that bends and deforms when influenced by a field or struck by a collision object (AUTODESK, Autodesk Maya 8.5, 2007; AUTODESK, Autodesk Knowledge Network, 2016)

Animation is a great way to bring life to an idea or design that is still in the nascent stage or on the drawing board. Animation helps to bring to life in more detail graphic objects that are difficult to express through words or illustrations (Sanders A. , 2016) . A good example is the dynamic workings of an automobile engine block or the study of the cosmos in cosmology etc.

This research paper tries to identify the key technique used in modeling T-Rex. The rest of the paper is arranged as follows: Discussion begins with the mathematical background of the work, while the methodology used in the work follows. The results of the work are then discussed, which is closely followed by the conclusion.

Related Works

The knowledge required here is that of Non-uniform Rational Basis Spline (NURBS), and Soft Body Dynamics (AUTODESK, Autodesk Maya 8.5, 2007; Maya: Understanding Rigid and Soft body Dynamics, 2016) . A number of spheres, polyset NURBS (Hu, Li, Ju, & Zhu, 2001) surfaces and many more are modeled, animated and rendered (Weisstein E. , NURBS Surface, 2016a; Modi & Tijjani, 2017) . Such objects are made up of control points, polygon faces, patches etc. Using Euclidean norm, Hu, Li, Ju, & Zhu (2001, p. 904) observed that the shape modification of NURBS surfaces had various contraints.

The geometric refinement of NURBS (Qin & Terzopoulos, 1995) remains adhoc and ambigous. However, improving the use of weights is still a major reasearch consideration. Qin & Terzopoulos (1995) proposed a dynamic NURBS swung surface which was inspired by their previous physics-based generalization of standard geometric NURBS. The idea was to find a way a better way to deal with the problem of unstructured shape constraints, which are particularly problematic for cross-sectional designs. By preserving the exact geometric shapes as well as the data sizes of models, data embedding in NURBS curves and surfaces using reparameterization was made possible by Ohbuchi, Masuda, & Aono, (1999). The detailed formulation of the NURBS surface object and the concepts of B-Splines and Hulls were discussed more extensively by Modi & Tijjani, (2017). However, a B-spline is just a generalization of the Bézier curve. The Bézier curve constantly passes through the first and last control points and is placed within the curved hull of the control points of the curve.

Background

The background elaborates a little more on soft body dynamics, especially as it relates to deformable objects that have skeletal structures that hold and determine the motion of the object under consideration.

Soft-body Dynamics

Soft *body dynamics* permits the use of *forces* on an *actual surface* to get realistic secondary motion. A *soft body* is actually a *surface* that is defined by *particles*. These particles can then be animated using *dynamic forces*, and the surface *deform*s as required.

A challenge in computer animation according to Ha, Ye, & Liu (2012) is that of physically simulating a virtual character that is doing a highly dynamic motion process. Landing and falling objects are achieved by the process of physical simulation of biped motion which has extensive use in robotics, particularly animal locomotion involving motion that simulates walking (Yin, Loken, & Van De Panne, 2007), jumping (Da Silva, Abe, & Popovic, 2008) etc.

However, when dealing with soft bodies and their controls, a number of forms have been proposed such as: external forces (Barbic & Popovic, 2008) , muscle fibers (Tan, Turk, & Liu, 2012) , and skeletons (Kim & Pollard, 2011) . Furthermore, the deformation of a given soft object is a "time dependent map from its undeformed material coordinates $\boldsymbol{X}$ to the world coordinates $\boldsymbol{x}$" (Liu, Yin, Wang, & Guo, 2013).

The soft body dynamics encoded in by Liu, Yin, Wang, & Guo (2013) generally followed the Finite Element Method (FEM) used for representing a soft body as a volumetric tetrahedral mesh with low-resolution, having a linear shaped function. The tetrahedral mesh elements used to encode the dynamics of the soft body is given in Equation (1):

$$M\ddot{x} = f_e + f_d + f_c + f_g \quad \ldots\ldots\ldots(1)$$

where $M = \text{diag}\{m_1, m_2, \ldots m_n\}$ is the mass matrix and m_i is the lumped mass of each given node i. f_g and f_c are the gravity and contact and gravity forces acting on the body respectively. The forces f_d and f_e are the damping and elastic forces acting on the body respectively. Liu, Yin, Wang, & Guo (2013) then computed the position of a given on the surface geometry by simply interpolating the nodal positions of the tetrahedral elements that enclose the vertex as seen in Equation (2):

$$\ddot{x} = \sum_{i=1}^{4} \o_i \boldsymbol{x_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

where $\sum_{i=1}^{4} \o_i = 1$ is the barycentric coordinates of the vertex. The vertices nodes are all assembled together as seen in

Equation (3):

$$\widetilde{x} = J\o\, x \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3),$$

where $\boldsymbol{J}$ is the constraint Jacobian. Notwithstanding, the deformable body dynamics adopted in this paper is that of Jain & Liu (2011), where two deformations were measured. The surface of the deformable body was defined as a 3D manifold triangular mesh, which is made by a set of point masses at the vertices (Jain & Liu, 2011) . The equivalent restoring forces at each vertex $\boldsymbol{v_i}$ and each edge $\boldsymbol{e_{i,j}}$ $\forall$ $\boldsymbol{v_i}$, and $\boldsymbol{v_j}$ is given such that; for a vertex $\boldsymbol{v_i}$, the deformation is measured from its rest position $\bar{x}$ as $x_i - \bar{x}_i$. As such, the equivalent restoring spring force that is applied with a spring stiffness k_v is given as seen in Equation 4:

$$f_{1,d} = -k_v(x_i - \bar{x}_i) \ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Likewise, the deformation of the edge $\boldsymbol{e_{i,j}}$ connecting $\forall$ $\boldsymbol{v_i}$, and $\boldsymbol{v_j}$ is given by $(x_i - x_j) - (\bar{x}_i - \bar{x}_j)$. Thus, the restoring force applied on $\boldsymbol{v_i}$ with a spring stiffness k_e corresponds to the equation as seen in Equation (5):

$$f_{1,d} = \sum_{J\in N(i)} - k_e\left((x_i - x_j) - (\bar{x}_i - \bar{x}_j)\right) \ldots(5)$$

where $N(i)$ represents subset of the vertices beloging to the mesh linked to the vertex $\boldsymbol{v_i}$ through a given edge (Jain & Liu, 2011).

The given force in Equation (4) tries to retain each vertex at its resting point, while the force described in Equation (5) attempts to keep the relative location of the vertex with respect to the vertices relatively surrounding it.

According to Jain & Liu (2011) there is a penalty for attempting to translate or rotate the entire mesh by the forces. Also, the forces try to keep the mesh in its undeformed state.The equations of motion finally put together by Jain & Liu (2011) for the linear system of equations is as seen in Equation (6):

$$M\ddot{x} = k_x (x_i - \bar{x}) - k_{\dot{x}}\dot{x} - g\ldots\ldots(6)$$

Where, $\boldsymbol{M}$ represents the diagonal mass matrix having diagonal entries that correspond to each vertex $\boldsymbol{v_i}$ with m_i and g as the gravitational forces. This is all based on the fact that, the restoring forces stated in Equation (4) and (5) ∀ the vertices is considered to be the product of the sparse matrix k_x and the deformation x - $\bar{x}$, where f = k_x (x - $\bar{x}$). As such, by adding a velocity damping force by means of a damping matrix $k_{\dot{x}}$. Further note that $\boldsymbol{x} \equiv (\boldsymbol{x_1}^T, ..., \boldsymbol{x_N}^T)^T$, where N is the number of vertices in the mesh, is the vector that contains all the collected postions of the vertices in mesh.

Methodology

A number of methods were deployed in order to build the animal character object named Flippy. The basic methods include:

a. NURBS

The body dynamics of Flippy was built using a primitive NURBS surface which is *sphere* object. By using the *grid snap* the sphere was then used to shape the body and head of Flippy. The process of shaping the head, body and torso of Flippy was achieved using selection *mask hulls* and *control vertices* (CV) as described in work by Modi & Tijjani (2017). The body and neck CV's of Flippy were well rotated, scaled and moved appropriately to give the appropriate shape. The same approach was used to create the fore and hind limbs. Finally the body was *deformed* in a realistic manner as seen in Figures 1a and 1c.

b. Skeleton Dynamics

Simulation and control of human-like soft characters is extremely challenging. This is due to the coupling process between the skeleton, the soft body and complexity of human skills with large numbers of Degrees of Freedom (DoFs) (Liu, Yin, Wang, & Guo, 2013) . However, the *skeleton* object mimicking the bone structures of Flippy was placed along its spine and the four limbs. This was achieved by using the *skeleton tool* and setting the *joint size* to an appropriate size measured in percentage. Afterwards, the skeleton was *bound* to the NURBSs surface as seen in Figure 1b and 1c.

c. Shading Group

To define the textured appearance of Flippy's skin, a new *shading group* was created. The skin was made dark brown and slick with a slight *shininess* as seen in Figures 1c. A *blinn* colour texture material with attributes: hue, saturation and value were set on the *material node.* Afterwards, the facial features i.e. the eyes, nose and whiskers, which are all NURBS surfaces were shaped into position to create the face of Flippy.

d. IK Spline handles and Solver

To help control the limbs of Flippy, *inverse kinematic* (IK) chains were built. The chains helped to control the movement of the limbs, while Flippy bounces the ball object off from its nose.

To animate a joint *hierarchy*, each joint had to be rotated individually to set the poses for Flippy. With the IK chain, the *start joint* and an *end joint* was created. The *start joint* is the *root* of the chain, while the *end* *joint* has a corresponding *end effector*. The end effector was defined by placing an *IK handle* that is then used to control all the bones within the chain.

The *IK Spline* solver is another type of IK solver that lets you use a *curve* to control the joint rotations. This permitted the skeleton chain to be animated by either moving the joints along the curve, or by animating the position or shape of the curve as seen in Figure 4.

The floor surface was built and textured using the *primitive polygon surfaces.* Afterwards, the floor was then beautified by adding reflectivity, shadow and lighting, which are basic effects required to render (Modi & Tijjani, 2017) the entire scene.

Discussion

The discussion dwells very much on the practical developments as they appear on every scene, each of the following are the various stages of development on the MAYA tool development environment.

a. Body, Neck and Head

This was achieved by sculpting the primitive sphere and editing its input node using a show manipulator tool. The initial size of the sphere object was set to s*ize = 14*. This was gradually increased and more *isoparm spans* set at the value of *14* offered more *flexibility* when *sculpting* the surface into the shape of Flippy. At this stage, the neck, body and head *CV's* were edited by *scaling and rotating* the manipulators accordingly to form its required shape.

b. Building Skeleton Joints

The initial joint *size* = 25% was set to the *joints* along Flippy's spine and head. This allowed for up to *8 joints* to be placed altogether. In order to properly adjust the positions of the 8 joints, all the *abdomen, fore and hind leg NURBS surfaces* were selected. This permitted work to be done on the joints only, without the *body and head surfaces* interfering. Furthermore, the *skin* of the *body* and *head of* Flippy were bound to the skeleton hierarchy as a single *object* as seen in Figure 1a.

After building the fore and hind *flipper* limbs, the required skeleton joints were attached with each *fore limb* having *four joints* and a *connector joint.* Each *hind limb* has *two joints* and a *connector joint.* By grouping the existing constructed hind and fore flippers as seen in Figure 1b, it was easy to then use a *mirror* method to symmetrically create the same for the other side of Flippy's body. The grouping process avoided having to reconstruct the other parts all over again. The new joints were all connected together at the *hip* joint and at the *shoulder* joint. This permitted easy movement at the shoulder joint.

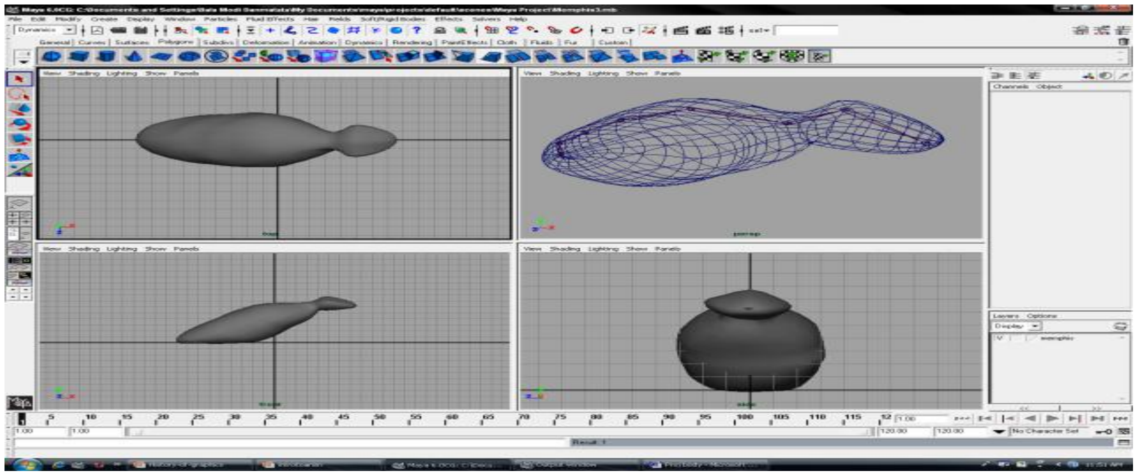
Figure 1a: Building the Body, Neck and Head and Skeleton Joints.

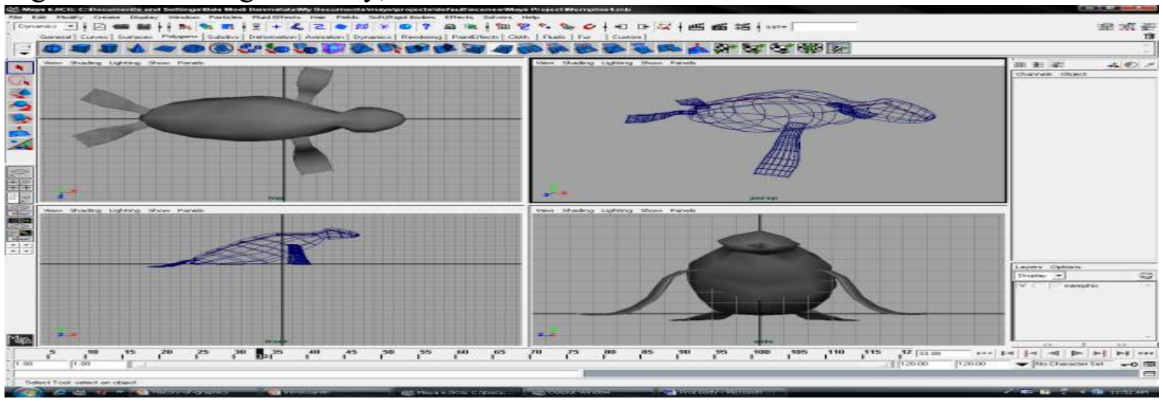
Figure 1b: Building the Body, Neck and Head and Skeleton Joints (*cont'd*).

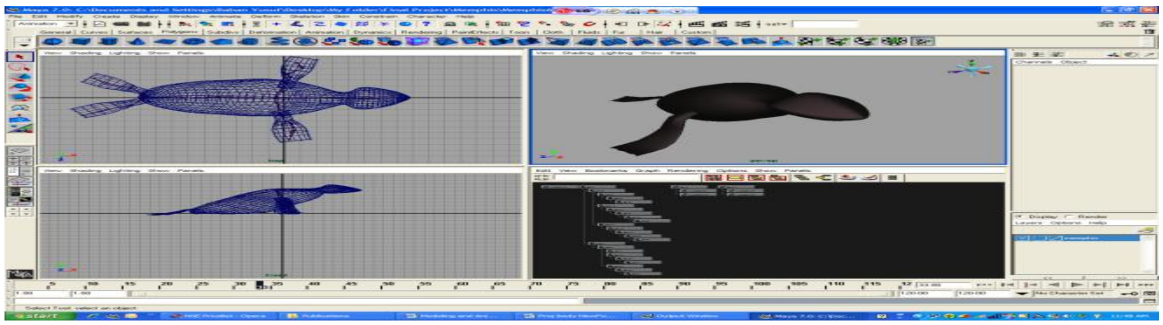
Figure 1c: Building the Body, Neck and Head and Skeleton Joints (*cont'd*).

c. The Texturing Process

A new *shading group* was created to define the appearance of Flippy's skin. The skin was made slightly dark with a slight shininess. To achieve all this, the *attributes* were set on the *material node*. To achieve that, the entire NURBS surfaces namely: body, head and limbs was assigned a *blinn material node* which is responsible for giving the skin a *shiny, metallic or wet-looking surface* appearance.

In creating the facial features, an eye assembly was built that allowed for the control of the direction of Flippy's vision. The *goal* was to explore how to model *procedural capabilities* to help make it easier to animate all aspects of the eyes, before some *whiskers* and a *nose* was

added. All of these elements are *texture-mapped* to prepare them for *rendering*. Next, the eye was controlled, while focusing as it was been animated. Firstly, an eye *locator object* was created and then *constrained* the eyeball to aim towards it. After creating the eye and its lid, it was time to *deform* the eye *group* appropriately. To achieve that, a *lattice deformer* was applied to the group.

By adding a *cluster* to the lattice deformer, it was possible to manipulate the shape of the eye group by editing the CV's of the lattice frame. Lastly, the *whiskers* and *nose* were created using another primitive NURBS surface. For all these pieces to move together when Flippy's head moves, they were connected by parenting into the skeleton as seen in Figures 4-8.

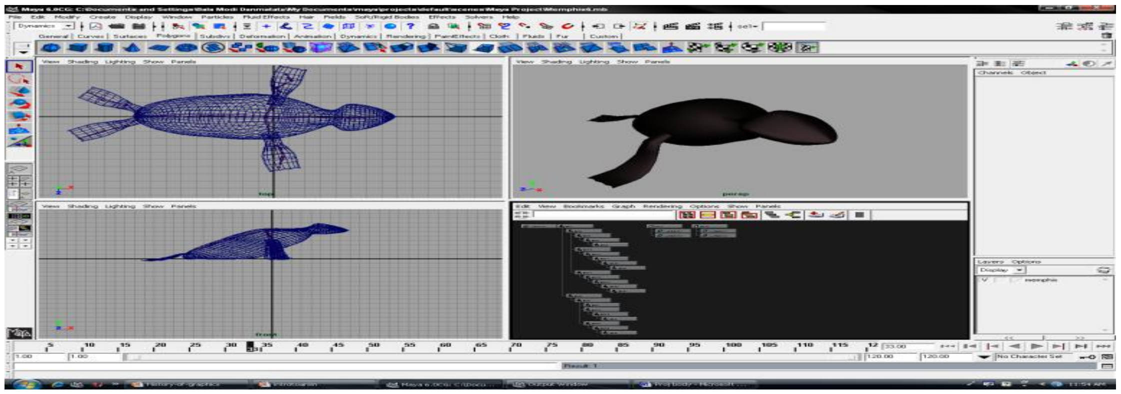
Figure 4: The Texturing Process and Eyes.

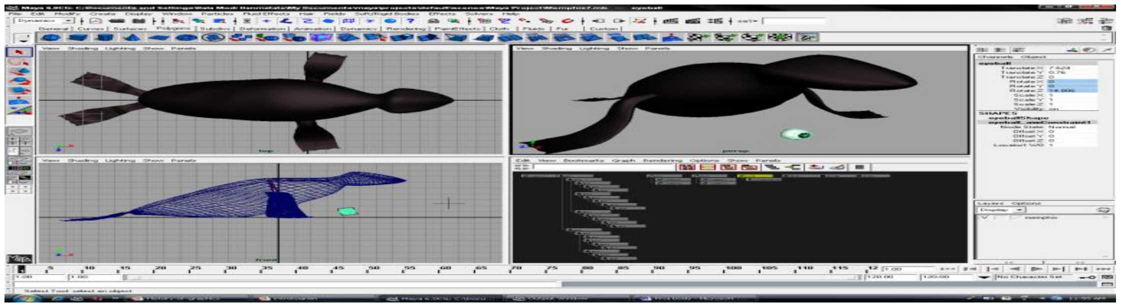
Fig 5: The Texturing Process and Eyes (*cont'd*).

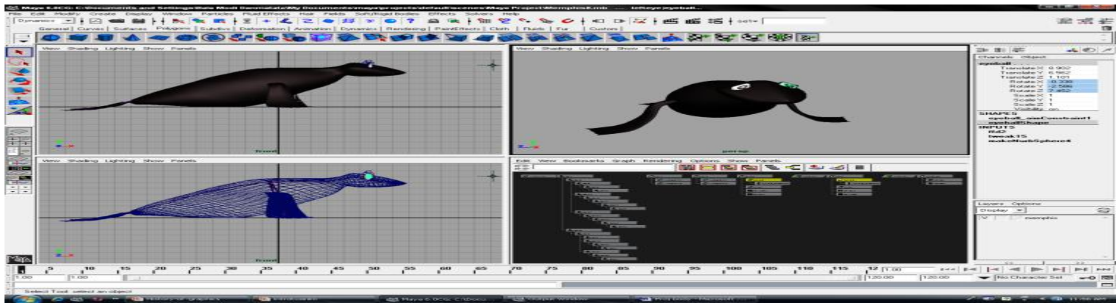

Figure 6: The Texturing Process and Eyes (*cont'd*).

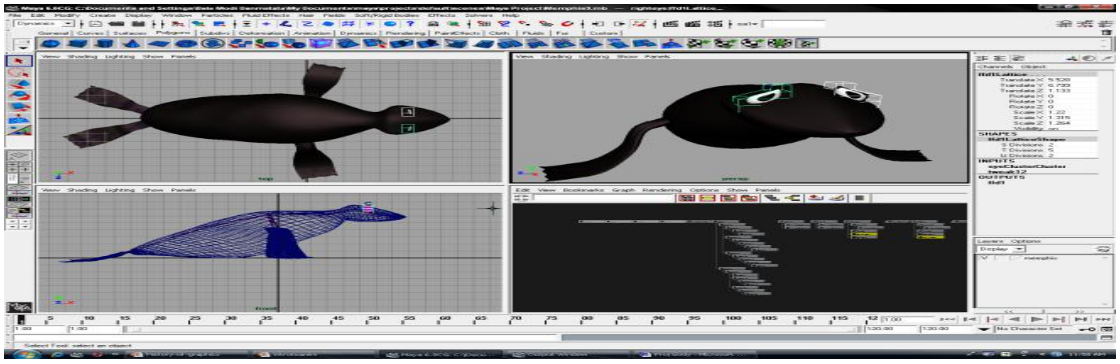

Figure 7: The Texturing Process and Eyes

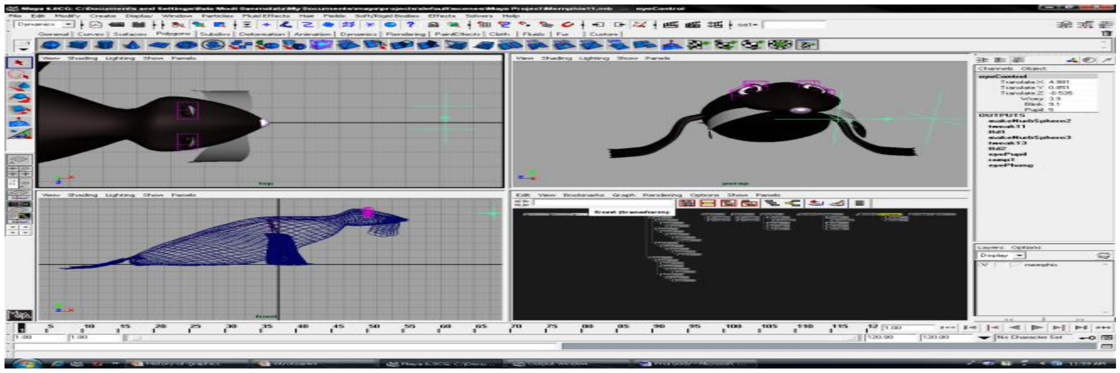

Figure 8: The Texturing Process and Eyes

d. The Animation Process

By turning the *body surface* and displaying only the skeleton, it was possible to control Flippy's limbs by using *inverse kinematic* (IK) chains. These allowed for the control of the limbs, while Flippy bounces the *ball* object from its nose. To successfully animate the *joint hierarchy*, each joint had to be set individually for different poses.

However, due to the time consuming nature of setting keys for all of the joints, an *IK chain* was set instead. The *start joint* and an *end joint* were defined. The start joint, is the *root* of the chain while the end joint has a corresponding *end effector*. The end effector was defined by placing an *IK handle* which is used to control all the bones within the chain as seen in Figure 9.

The IK chain starts with the *root* joint of each limb and ends. The *IK Handle* was placed on the second to the last and the last joint, which helped staple the feet of Flippy to the rigid body floor during animation. The same was done to the left limbs. Afterwards, the ball, which the character Flippy played was created, and was made to throw the ball and also watch as the ball *bounces away*.

The actual animation began at frame 50 which shows Flippy tossing the ball up into the air, while gradually rotating Flippy slightly up with the ball touching its nose. At frame 70, Flippy bounces the ball trice, and the process simply copied the existing set keys to extend the animation as seen in Figures xxx.

The animation process ended with the application of a *dynamic simulation* process. This form of animation permitted the *ball* to drop off Flippy's nose by applying a *gravity field* and an *air field* to blow the ball to the left side of Flippy and drop downwards too. This sort of simulation allowed for the use of *rigid body dynamics* (Jain & Liu, 2011) where the objects position is animated. Afterwards, Flippy's whiskers were also animated using soft body dynamics where the surface was deformed using similar dynamic forces.

Flippy motion was then completed by moving its *cluster handle* at frame 240 a frame at a time to begin following the ball and *set* key along the z-axis until Flippy was looking at the ball as it rolls away somewhere between frame 270 to 380. It was at this stage that the *whiskers* were made to *flop* during the animation playback.

To achieve this, the surface was made a *soft body* defined by *particles*. These particles were then animated using dynamic forces, which made the surface to deform accordingly. The whiskers were made to move around so as to always maintain their original shape as seen in Figures 9 and 10.

e. The Play Stage

The stage where Flippy was to put up its performance was constructed using a *passive rigid body* to allow for possible *collision* (Jain & Liu, 2011) . This permits the ball to bounce *on* and not *through* the deck surface. In order to create the *back wall*, control vertices were extended up along the y-axis as high as possible.

Once the stage was made ready, *light* effects were added. From the *rendering menu* a *spot light* was created and positioned the *look at* and *eye points* of the light using the *show manipulator* tool, so that the light is pointing down at Flippy from the upper left corner of the stage. Furthermore, a *camera* was placed to *preview* the animation. This was achieved by adding a new camera and animating its position. When working in 3D space, an animated camera helps to *accentuate* the *depth* of a scene. In order to texture the deck surface, which was important prior to *rendering* of the deck surface, a *phong material* node was added and then *mapped* a grid texture to the colour as seen in Figure 10. Also, a *bump map* was then added to the phong material, to give the floor a *tile* floor effect. Lastly, the bump map was reassigned as a *specularity map* before reviewing the final result using *ray trace* rendering (Modi & Tijjani, 2017). This in effect gave the entire scene a reflective and realistic effect as seen in Figure 12.

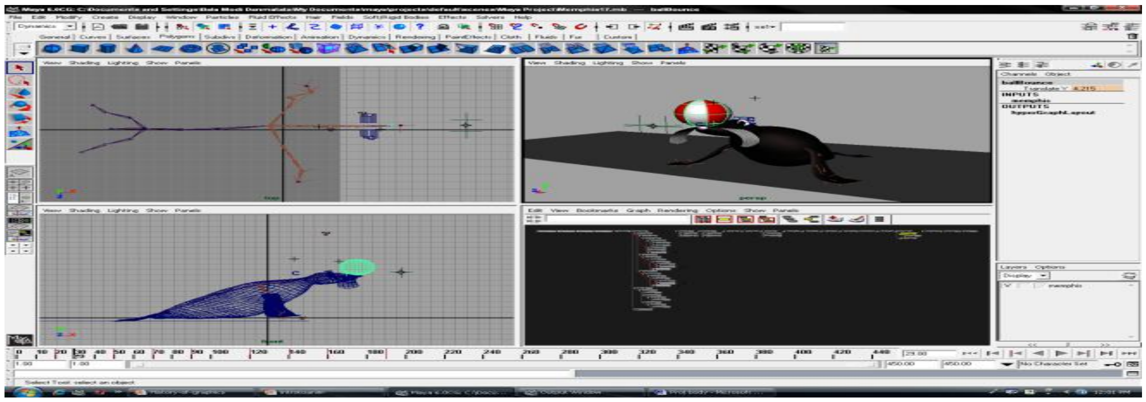
Figure 9: Building the Stage and Animation.

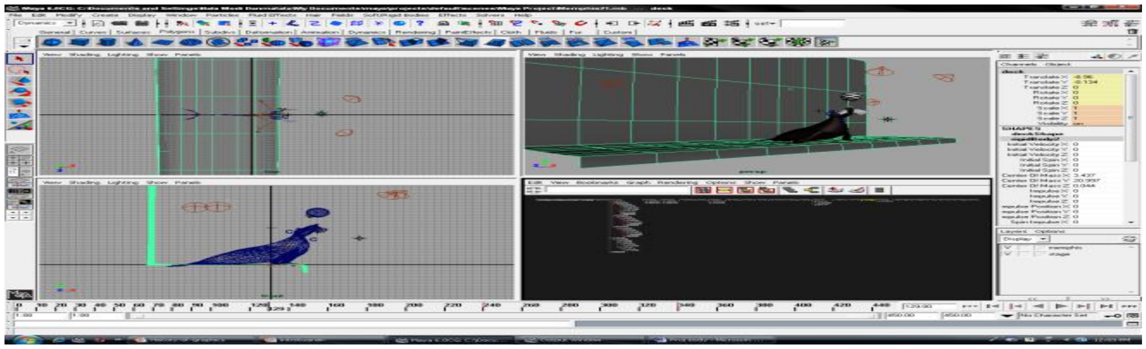
Figure 10: Building the Stage and Animation (*cont'd*).

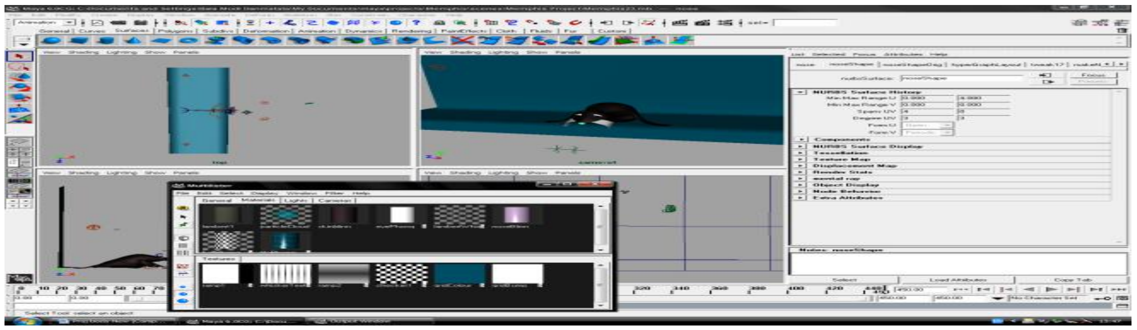
Figure 11: Building the Stage and Animation (*cont'd*).

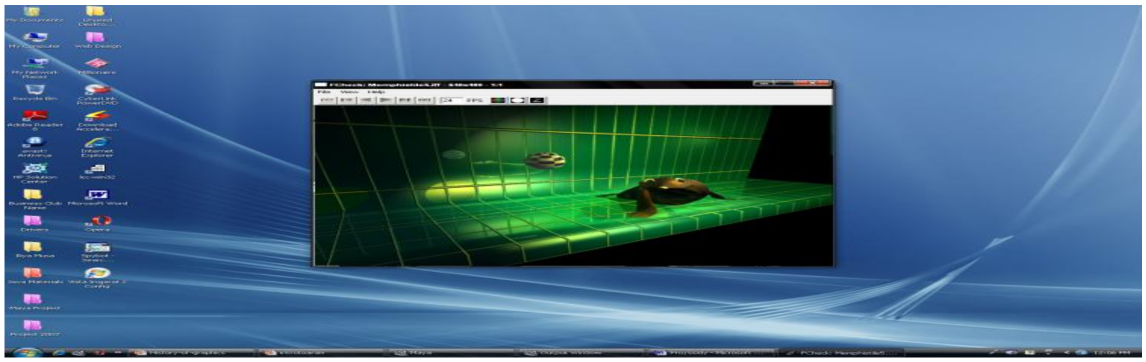

Figure 12: Finished Model and Animated Scene.

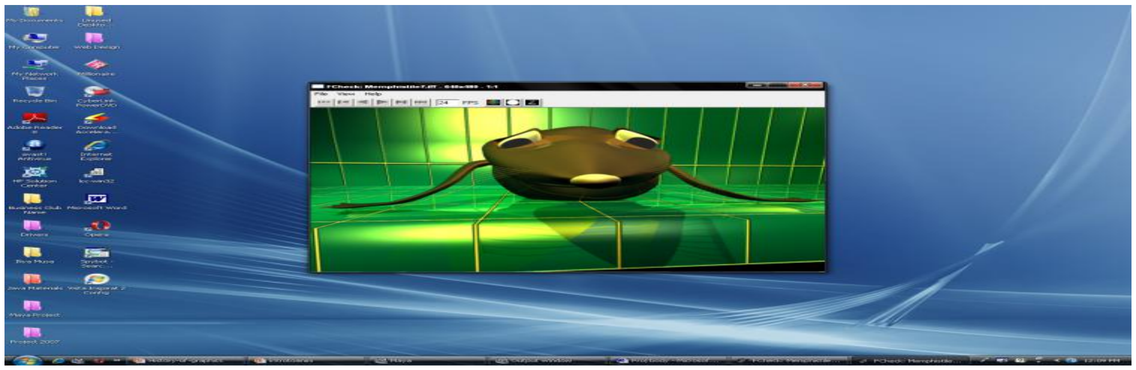

Figure 13: Close morgue up of Memphis.

Conclusion

For the purpose of this project, a combination of *NURBS* surfaces and soft body dynamics are used to build Flippy and make its movements possible. For this purpose *skeleton* objects were used for the bone-like structure that enabled possible motion. The skeletons were placed at specific parts within its body, thereby realistic animal like motion. *materials, shaders and textures* were utilised before the entire scene was rendered to give it a more polished and realistic glow effect. *Software rendering* was deployed due to the machine limitation.

As a future work consideration, this paper hopes to delve into overcoming the problem of the expensive cost of computing a large number of vertices contained on mesh, which makes the dynamic system difficult to build.

This could be achieved possibly by pre-computing and taking the averages of the point masses while still in their rest positions in a given mass matrix.

This however will require reviewing the basic concept behind deformable body simulations.

REFERENCES

Maya: Understanding Rigid and Soft body Dynamics. (2016, August 26). Retrieved from steves-digicams.com: http://www.steves-digicams.com/knowledge-center/how-tos/video-software/maya-understanding-rigid-soft-body-dynamics.html#b

AUTODESK. (2007). In Autodesk, *Autodesk Maya 8.5* (pp. 1-474). San Rafael, CA USA: Autodesk.

AUTODESK. (2016). *Autodesk Knowledge Network*. (Autodesk Maya Incorporation) Retrieved July 18, 2016, from Autodesk 3DS Max: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-4422403E-3EA1-4103-B718-41A0954CF6A6-htm.html

AUTODESK. (2016). *Autodesk Knowledge Network*. (Autodesk Maya Incorporation) Retrieved July 18, 2016, from Autodesk 3DS Max: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-4422403E-3EA1-4103-B718-41A0954CF6A6-htm.html

Barbic, J., & Popovic, J. (2008). Real-time control of Physically based simulations using Gentle Forces. *In ACM Transaction on Graphics, 27*(5), 163.

Da Silva, M., Abe, Y., & Popovic, J. (2008). Interactive Simulation of Stylized Human Locomotion. *In ACM SIGGRAPH 2008 Papers*, 82:1-82:10.

Engleman, J. (2007, March 6). *The Gnomon Workshop.* Retrieved March 14, 2007, from Character Modeling in Maya: http://www.thegnomonworkshop.com

Ha, S., Ye, Y., & Liu, C. K. (2012). Falling and Landing Motion Control for Characters. *In ACM Transactions on Graphics (TOG), 31*(6), 155.

Hu, S. M., Li, Y. F., Ju, T., & Zhu, X. (2001). Modifying the Shape of NURBS Surfaces with Geometric Constraints. *Computer-Aided Design, 33*(12), 903-912.

Jain, S., & Liu, C. K. (2011). Controlling Physics-based Characters using Soft Contacts. *In ACM Transactions on Graphics (ToG0, 30*(6), 163.

Kim, J., & Pollard, N. S. (2011). Fast Simulation of Seleton-driven Deformable Body

Character. *In ACM Transaction on Graphics, 30*(5), 121.

Kumar, S., & Manocha, D. (1995). Efficient Rendering of Trimmed NURBS Surfaces. *Computer-Aided Design, 27*(7), 509-521.

Liu, K., Yin, K., Wang, B., & Guo, B. (2013). Simulation and Control of Skeleton-driven Soft Body Characters. *In ACM Transactions on Graphics, 32*(6), 215.

Modi, B., & Tijjani, B. (2017). Application of Control Vertices, Curves and Hulls in Three Dimensional Modeling and Animation of Geometric Surfaces. *BIMA Journal of Science and Technology, Faculty of Science, Gombe State University, Gombe, 1*(1), 113-130.

Naas, P. (2013). *Autodesk Maya 2014 Essentials* (1st ed., Vol. 1). Indianapolis, Indiana, USA: Sybex, Wiley.

Ohbuchi, R., Masuda, H., & Aono, M. (1999). A shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces. *In Proceedings of Computer Graphics International* (pp. 180-187). Washington DC, USA: IEEE.

Qin, H., & Terzopoulos, D. (1995). Dynamic NURBS Swung Surfaces for Physics-based Shape Design. *Computer-Aided Design, 27*(2), 111-127.

Sanders, A. (2016, July 31). *An Introduction to Computer Animation*. Retrieved September 2, 2016, from About.com: http://Animation.about.com

Sanders, A. (2016, August 31). *An Introduction to Computer Animation*. Retrieved from About.com: Animation.about.com

Stahlberg, S. (2004, August 12). *Techi Warehouse*. Retrieved March 5, 2007, from http://www.techiwarehouse.com

Tan, J., Turk, G., & Liu, C. K. (2012). Soft Body Locomotion. *In ACM Transaction on Graphics, 31*(4), 26.

Weisstein, E. (2016a, June 24). *NURBS Surface*. Retrieved July 7, 2016, from MathWorld: http://mathworld.wolfram.com/NURBSSurface.html

Weisstein, E. (2016b, June 24). *B-Spline*. Retrieved July 8, 2016, from MathWorld: http://mathworld.wolfram.com/B-Spline.html

Weisstein, E. (2016c, June 24). *Bezier Curve.* Retrieved July 9, 2016, from MathWorld:

http://mathworld.wolfram.com/BezierCurve.html

Weisstein, E. (2016d, June 24). *Convex Hull.* Retrieved July 9, 2016, from MathWorld: http://mathworld.wolfram.com/ConvexHull.html

Yin, K., Loken, K., & Van De Panne, M. (2007). Simbicon: Simple Biped Locomotion Control. *In ACM SIGGRAPH*, 105.