Construction of Software to Directly Manipulate Character Controls

Walter Fulbright
Clemson University, wfulbri@gmail.com

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Recommended Citation
https://tigerprints.clemson.edu/all_theses/2946

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.
Construction of Software to Directly Manipulate Character Controls

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Fine Arts
Digital Production Arts

by
Walter Fulbright
August 2018

Accepted by:
Dr. Brian Malloy, Committee Chair
Dr. Donald House
Insun Kwon
Abstract

Rigging is the process of creating a skeletal system for an animated 3D character or object, allowing animators to manipulate the character with the use of NURBS controls. These controls often make it difficult for animators to see the silhouette of the character since the controls clutter the body. Direct selection controls offer a solution to this issue by eliminating the need for visible controls. This thesis covers the creation, use, and the process of direct selection controls. Explaining the issues of traditional rigging and how it can be solved with direct selection controls.
Acknowledgments

I would like to take this opportunity to thank everybody within the DPA program. Especially my committee chair Dr. Malloy for really helping me complete this thesis. If it wasn’t for your python class I would not have been able to complete the coding aspect of this project without your teaching. I also thank Dr. House for coming on last minute to be part of my committee and for taking the time to read through my thesis and give feedback as well. I give special thanks to Prof. Insun Kwon, for introducing me to modeling, if it was you I would never have become interested in character modeling. Finally I would like to give a huge thank you to my family and friends for always encouraging me over the years and for the continuous support that you have shown me.
Table of Contents

Title Page ................................................................. i
Abstract ................................................................. ii
Acknowledgments ......................................................... iii
List of Figures ........................................................... v
1 Introduction ............................................................. 1
2 Background ............................................................... 4
  2.1 Keith Lango ....................................................... 4
  2.2 Wrap Deformer ..................................................... 5
  2.3 Premo .............................................................. 6
  2.4 Presto ............................................................. 8
3 Modeling ................................................................. 9
  3.1 The Process ...................................................... 10
  3.2 Topology ......................................................... 12
4 Rigging ................................................................. 16
  4.1 Setting Up the Rig ............................................... 17
5 Direct Selection ...................................................... 23
  5.1 Interface ......................................................... 24
  5.2 Implementation .................................................. 26
6 Results ................................................................. 28
7 Conclusion ............................................................. 33
Bibliography ............................................................. 35
List of Figures

1.1 (A) Traditional Controls (B) Direct Selection Controls ................................. 1
1.2 Eslam AboShady inktober drawing ................................................................. 3

2.1 Kieth Lango body trigger [10] ................................................................. 5
2.2 Example of wrap deform [1] ........................................................................... 6
2.3 Wrap Deformer Diagram .................................................................................. 6
2.4 Emo being used for How to Train Your Dragon [7] ........................................ 7
2.5 Screen shot of Premo [7] ................................................................................ 8
2.6 Screen shot of Presto [8] ................................................................................ 8

3.1 EA Sports Madden comparison [12] ............................................................... 10
3.2 Eslam AboShady inktober drawing ................................................................. 11
3.3 Render of bloodhound ..................................................................................... 12
3.4 Sergi Caballer edge flow demonstration ............................................................ 14
3.5 Topology of the model with poles highlighted .................................................. 15
3.6 Notice the difference in distortion of reflection by inserting a edge loop ........... 15

4.1 Joint hierarchy for the neck .............................................................................. 19
4.2 Bloodhound Rig: Joints and Traditional controls ............................................. 20

5.1 Direct Selection Diagram ................................................................................ 23
5.2 Interface for Direct Selection ........................................................................... 23
5.3 Transform Geometry Node .............................................................................. 27
5.4 Delete Faces ..................................................................................................... 27

6.1 Rig setup with direct selection controls .......................................................... 29
6.2 Rig setup with traditional controls .................................................................. 30
6.3 Three different poses of the dog ..................................................................... 30
6.4 (A) Traditional Rig (B) Direct Selection Controls ......................................... 31
6.5 (A) Traditional Rig (B) Direct Selection Controls ......................................... 31
6.6 All the selectable faces in the direct selection controls rig .............................. 32

7.1 Alternative method to direct selection controls .............................................. 34
Chapter 1

Introduction

When rigging comes to mind, we imagine hundreds of controllers encircling the body allowing the animators to control or manipulate the body. All of these controls impair the animator from freely viewing the character during animation without having to continuously hide controls or view the scene in a different perspective or camera, Figure 1.1. This thesis discusses the process of turning a simple concept into a production-ready rig, covering the importance of modeling and facial rigging, with an emphasis on explaining how direct selection controls can be built on top of a rig using only Maya software and python scripting.

![Figure 1.1: (A) Traditional Controls (B) Direct Selection Controls](image)

The inspirations for this project came from watching a short video about "The Technology Behind How to Train Your Dragon 2" [14]. In the video, Simon Otto, the head of character animation at DreamWorks gives a demonstration of DreamWorks proprietary animation software Premo. Otto
demonstrates Premo ability to select within any region of the character that would normally have controls and begin to animate the character without any lag. However, the most impressive aspect of the demonstration is the amount of window space freed up by not having controls or a GUI picker menu within the scene or window. This allows animators the ability to become immersed in the scene.

Direct selection controls is a considerable improvement since the current method of animating, or rigging can result in a cluster of controls surrounding the body. Even correctly placed, controls can become a hindrance to animators when trying to view the character. To get around this issue in the past: (1) animators could either place all the controls within a layer and toggle the visibility of the layer, (2) open up a new viewport where none of the controls are visible, or (3) have a GUI picker system off to the side of the window or scene. With direct selections, these issues can be resolved, without having to compromise the rig or screen space. To illustrate this, Eslam AboShady’s concept of a bloodhound sitting on a park bench shown in Figure 1.2, was used as the concept for the model. The long ears and droopy jaw made it perfect to show some of the possible weaknesses of direct selections of controls since some of the controls have to be located beneath the jaw. Throughout this thesis, AboShady’s concept of the bloodhound, Figure 1.1, will be referred to as bloodhound.

The problem with direct selection controls is that there is currently no implementation publicly available; all current implementations, such as Presto and Premo, represent proprietary software developed in-house at Pixar and Dreamworks respectively. Thus, in this thesis we describe techniques to create direct selection controls and we provide a proof-of-concept implementation using Maya and Python scripting. This thesis begins with an overview of related works and companies that are currently using direct selections controls within their pipeline. Chapter 3 explains the process of creating the model. This chapter also includes key theories to follow when modeling, whether its organic or hard surface modeling. The traditional method of rigging a character using NURBS curves is discussed in Chapter 4. Chapter 5 gives an overview of direct selection controls. This chapter also discusses the interface created to automate the creation of direct selection controls. Chapter 6 compares the benefits and disadvantages of direct selection controls in comparison to traditional rigs using NURBS curves. Finally, Chapter 7 will conclude with possible improvements and future uses of the direct selection control software.

To clarify, direct selection controls can be referenced as many different things, such as free-
form controls, deformable controls, direct manipulation controls. To keep things simple, it will be referred to as direct selection controls in this paper.
Chapter 2

Background

The idea of direct selection of controls is a relatively new concept that became popular in 2014 when DreamWorks and Pixar introduced Premo [7] and Presto [5]. So there is little history or information about direct selection controls. However, there have been several different attempts to create direct selection controls. The first attempt at direct selection controls was by Keith Lango [10] in 2008, the second method is to use wrap deformers of the controls ¹.

2.1 Keith Lango

The idea of direct selection was introduced but not developed in 2008 when Keith Lango released a video [10] creating invisible trigger controllers using a tool developed by Hamish McKenzie called, zooTriggered. Rather than using zooTriggered as a basic GUI Picker system, Lango used it to create trigger controls that are transparent. As Figure 2.1 shows, Lango created three cylinders to represent the upper arm, forearm, and hand placing them in the appropriate location to act as trigger controls. Lango then parent constraints the trigger controllers to the original NURBS controls and increases their transparency, rendering them invisible within the viewport. Parent constraints relate the parent object’s translations and rotations to the child object. He connects the trigger controls to the NURBS controllers, using zooTriggered. Trigger controls are controls that trigger the selection of another object when selected. By selecting the trigger control, the animator is selecting the NURBS curve controller located within the arm. However, this can’t be consider

¹The development of wrap deformers is not attributed to any single researcher, or group of researchers.
direct selection because a proxy controller is being used to control another controller and the joint.

Figure 2.1: Kieth Lango body trigger [10]

\[\text{Figure 2.1: Kieth Lango body trigger [10]}\]

### 2.2 Wrap Deformer

Wrap deformers is a powerful deformer that allows NURBS surface, curves or other meshes to deform another object [1]. As seen in Figure 2.2, the low-res sphere is the wrap influence object of the high-res sphere, meaning any changes made to the low-res sphere influences the high-res sphere. However, setting up deforming controls using the wrap deformer method is counter-intuitive.

To use wrap deformers to create direct selection controls the original mesh has to be duplicated twice, totaling three meshes, the original, blend shape and controller mesh, is shown in Figure 2.3. The original mesh is the wrap influence object for the blend shape mesh. The blendshape mesh connects to the controller mesh through a blendshape. This connection makes sure that any deformations produced by the blendshape mesh will influence the control mesh as well. The wrap
deformer method is a valid way to create direct selection controls however as more controls are added to the rig animators will notice an increase in the lag.

![Figure 2.2: Example of wrap deformer [1]](image)

![Figure 2.3: Wrap Deformer Diagram](image)

2.3 Premo

Premo was first used on the DreamWorks production of *How to Train Your Dragon 2* in 2014, revolutionizing their entire animation department [7]. Premo was built to replace DreamWorks
previous and out of date software, Emo. Emo was created in the 1980s, by Pacific Data Images and at the time it was considered groundbreaking. By no means was this process of animating considered natural and there were flaws. Animators had to punch in values into a spreadsheet database to animate the character, as shown in Figure 2.4. The second flaw was that there couldn’t be more than one character visible within the scene or else there would be a huge decrease in speed.

Premo changed everything, instead of animating through a spreadsheet, animators were able to control the character directly. Placing the pen back into the hands of animators and allowing animators to draw the characters’ pose rather than having to punch in values. Secondly, Premo introduced real-time rendering into their pipeline, allowing artists to see a full-resolution character and scene while animating the characters. Compared to Emo, real-time rendering was a considerable improvement. The difference is apparent when comparing Figure 2.4 to Figure 2.5. Thirdly, Premo reduced the amount of time it took to train an animator. Rather than taking animators twelve weeks of training to understand Emo, Premo allowed new animators to start working on productions after a couple of weeks of training.

Figure 2.4: Emo being used for How to Train Your Dragon [7]
2.4 Presto

Like Premo, Presto is proprietary software developed and used in-house at Pixar. Before the arrival of Presto in 2014, Pixar was using Marionette. Since the software was never released little is known about the program. Presto is relatively similar and rivals Premo in a lot of categories. Presto uses real-time rendering, which was helpful on Monster University [5]. Especially when it came to animating Sulley. Instead of having to hide the fur, the real-time render was able to render each follicle of hair without any lag. Like Premo, Presto uses direct selections of controls, eliminating the need for visible controls or GUI Pickers. What is most interesting about Presto is the program’s ability to subdivide the mesh in relation to the camera. As the camera approaches a character, the character’s subdivision levels increase automatically and as the camera zooms out the subdivision levels decreases [5]. All this is done on the fly with no lag.
Chapter 3

Modeling

Modeling or 3D modeling is the process of creating a physical representation of an object using a collection of points within a 3D space. These points can connect by a series of geometric shapes such as triangles, quads, or n-gons, the most popular being quads. These models can be produced by hand, procedurally or from scans.

Even though 3D scanning has been available since the 60’s, it’s the least popular modeling method of the three methods currently used in the animation industry. However, 3D scanning has become popular in the game industry. There are two primary ways to generate a 3D model from a scan, first is to use a laser. The laser records the changing distance and shape of the laser as it passes over the object and reconstructs the object in a 3D space using this data. The second method is to use photos. Every part of the object must appear in at least three photos to create a mesh from photos. One of the biggest game companies to use 3D scanning within the last decade has been EA Sports. Over the past years, the in-game graphics have become more and more realistic as seen in Figure 3.1. Pushing the envelope of real-time rendering. Nevertheless, 3D scans are not perfect and often require a sculptor to go back and touch up the details; however, 3D scanning is a useful way of generating an accurate base mesh.

Procedural modeling is a second way to create 3D models. Procedural modeling is best used to generate environment models, such as trees, forests, and cities. Procedural modeling can allow artists to create entire cities using parameters, rather than having to sculpt every building in the scene. Meaning little to no hand modeling is needed to generate a city. This is great for VFX artists since the destruction of entire cities has become popular within the film industry today. A
The two main styles for hand modeling an object are box modeling and digital sculpting. Box modeling can also be referred to as polygonal modeling. Box modeling is a technique where the final model is created from a box or another primitive shape. This method is best suited for modeling hard surface objects, like ships or robots. Unlike box modeling, digital sculpting is best for modeling organic surfaces. Digital sculpting allows artists to manipulate the object as if it’s clay. The artist can push, pull, smooth, build, or subtract volume from the object adding details such as pores, or wrinkles to the model, otherwise impossible to do with box modeling.

### 3.1 The Process

Even though the character is an anthropomorphic bloodhound and a cartoon, it’s important to understand the anatomy of a dog. Understanding the basic anatomy of a dog will yield a better
When a modeler is given a character, they usually receive a couple of drawings from the Look Development team. The first drawing provided is a character sheet. **Character sheets** is a front, side, and back profile of the character. Character sheets are handy when making sure the character has correct proportions. Since the bloodhound has such distinctive features, shown in Figure 3.2, it is essential to rely on the side and front profile of the character sheet to captivate these features. The second drawing is expression sheets. **Expression sheets** cover a wide range of expressions that a character might need to make during the film. Even though these are more beneficial for the rigger, they can still be helpful to modelers when modifying the topology of the model because the expression sheets show where deformations occur in the model and where extra edge loops will be needed. A modeler might or might not receive these illustrations since they are more valuable to the director or layout artists than anybody else.

After all the concepts are completed, the modeling process can begin. To start, a preloaded model and rig.

![Figure 3.2: Eslam AboShady inktober drawing](image)
sculpt of a dog is loaded into the scene. The dog is nothing special, a generic dog that’s included with Zbrush. It is a good starting point since the model is an anatomically correct dog. Making sure that the sculpt is anatomically correct when first blocking out the character is crucial. This makes adding smaller details like the bags underneath the eyes, or wrinkle along the neck, a lot easier. The goal is to create a physically appealing character. To do this, key features of a character are often blown out of proportion. Over accentuating the character key features of the bloodhound makes it seem more appealing. The eyes are large in relation to the rest of head because eyes large and far apart give cuteness to characters according to Preston Blair [3]. Both the bloodhound’s droopy ears and the nose are over exaggerated since these are two of a bloodhounds key features. The hope was to over exaggerate the sag along the jaw as well but this created complications when creating the rig, so the droopiness of the jaws had to be retracted. The teeth were shaped in reference to Max from *The Secret Life of Pets*. The teeth are a blend of human and bloodhound teeth, with the canines over-pronounced while the rest of the teeth are smooth, as shown in Figure 3.3.

![Figure 3.3: Render of bloodhound](image)

### 3.2 Topology

Topology is one of the key things that is frequently overlooked by modelers since they are often more concerned with having an aesthetically pleasing model. The start of a good character
rig begins with topology because poor topology can quickly ruin a good rig. There isn’t a specific
definition of how to retopoligize a face because each face is different, but there should be three key
concepts to follow when retopologizing any character. When retopologizing the bloodhound, edge
flow, quads, and subdivision were the three key concepts that were kept in mind.

3.2.1 Edge Flow

Edge flow is important when it comes to deformation, especially in the face. Edge loops
should follow the contours of the body and face helping to pronounce definition. For the bloodhound,
the two main areas of focus are the eyes and mouth. The edge loops should flow around orbicularis
oculi muscle in the eye and the orbicularis oris muscle in the mouth giving continuous edge flow.
This ensures that the deformations will be more natural when animating. An excellent example of
this is Segri [6] demonstrations of correct topology in Figure 3.4. The edge loops continuously flow
around the eyes and mouth in the human model. The same was replicated for the bloodhound. The
dge flow for the eyes looks similar to the Caballer’s model. For the mouth of the bloodhound, the
dge flow follows the same orbicularis oris that an average human would have. The only difference
is that bloodhounds mouth protrudes a lot farther from the face elongating the orbicularis oris. The
second thing to be extracted from the Caballer model [6] is the red dots on the models, these are
poles. Poles are vertices that share more than four edges, and they are used to change the direction
of an edge flow. Referring to Caballer’s model in Figure 3.4, take notice how the edge flow next
to these pole vertices makes a 45 to 90-degree change in direction. Poles are unavoidable in any
organic model, and as long they are placed in appropriate locations they can become quite useful for
deformation. However, two rules to follow when using poles is; first, two poles should never share
the same edge. Second, poles should never share more than five edges. If there is an excess of five
edges to a pole pinching or artifacts begin to appear in these areas when subdivided especially if the
mesh has any curvature. However, if the mesh is perfectly flat like a plane, no pinching or artifacts
will be visible even when subdivided. When retopologizing the bloodhound, the poles were placed
in correlation to the poles placed on Caballer model. The significant difference being the poles on
the dog mouth are lot closer to the mouth than it should be. Luckily this did not affect the model
when it came to deformations.

To have good edge flow, even spacing between edge loops is required. Edge flow should
always be evenly spaced out even when modeling hard surface objects. As seen in Figure 3.6,
pinching is noticeable along the edge of the ball. Resulting in distortion in the reflection of the building. The distortion was either caused by inserting an extra edge loop that wasn’t needed or by moving one of the vertices. At times it is necessary to place edge loops close to each other. This technique is called hardening and is used to define an edge and is commonly used in hard surfacing modeling but can be used in facial modeling. For the bloodhound, this technique was used to crease the edges around the eyelids so that the eyes retain their sharpness even when deforming to blink.

3.2.2 Quads vs. Triangles and N-gons

There are three main types of polygons for models: quads, triangles, and n-gons. Quads are the polygon standard for the industry today and for good reasons. First, quads easily allow for additional and removal of edge loops to occur as long as the edge loop does not run into a pole vertex. Inserting edge loops would not be possible with triangles or n-gons. Secondly, software like Zbrush and Mudbox require all meshes to be quads when imported. This is because Zbrush and Mudbox use subdivisions to add more significant details to a mesh, like scars or scales. Thirdly, artifacts can appear when rendering if the model has triangles or n-gons. These issues are most likely to occur in areas where n-gons are used. As for the bloodhound, it was mostly comprised of quads however it was unavoidable to use triangles. These triangles were hidden underneath the ears.
Figure 3.5: Topology of the model with poles highlighted

Figure 3.6: Notice the difference in distortion of reflection by inserting a edge loop

since they would be hardly visible. When modeling an organic character it is almost inevitable to have triangles, but if appropriately hidden the triangles will not pose an issue when subdivided.
Chapter 4

Rigging

Rigging is the process of creating a skeletal system for an animated 3D character or object, permitting animators to move the character. During this process, a hierarchical system of controls for the animators is built into the model [13]. If it wasn’t for rigs, the model would be stuck in its static pose that the modeler placed the character in. Like any part of the production, rigging is very technical and tedious, requiring hundreds of iterations and refinements. Even the simplest of rigs take a significant amount of time to complete. Complex rigs that allow animators full ranges of controls, to be able to get a wide range of emotions take even longer. An average production spends roughly five weeks on rigging the body alone and another five weeks on rigging the face of the main character. A few key terms to understand when rigging are joints, control curves, constraints, blend shapes, inverse kinematics (IK), forward kinematics (FK), deformers, skinning, and weight painting.

Joints act in the same manner as bones in humans, joints allow for the character to translate, rotate, and scale. When placing joints, it is essential to reference anatomy since the joints are the points of articulation that control the model. It’s important to get the placement of joints correct the first time because they can truly make or break a rig when it comes to skinning the character. After the joints are placed, control curves are needed to manipulate the joints. Control curves are simple NURBS curves that allow the animators to control regions of the rig that the controller dictates. Control curves are very beneficial for both the rigger and animator since they can be "Zeroed Out" at any time. This means that translations, rotations, and scale values can be returned to its default value, returning the character to their default pose. To control joints, constraints are used. Constraints limit the target position, rotation, and scale to the attributes of the parent object,
in this case, the control.

Blend shapes and deformers are often confused with one another since their definitions are pretty similar to each other. Blend shapes allow one to change the form of an object into another object. Blend shapes are commonly used when posing facial animations. They can be used on top of joints, ignoring the hierarchical systems. Blend shapes can also be used to preserve volume deformations when a joint rotates, an example of this would be the knee bending. Instead of losing volume when the leg rotated at a 90-degree angle, a blend shape can be used to preserve the volume. 

Deformers allow for the manipulation of large selections of vertices using algorithms. An example of this is a cluster deform; a cluster enables animators to control a group of vertices with a single controller. This can be particularly useful when trying to control the cheeks within a facial rig.

Inverse kinematics (IK) is a system where the child node within the rig’s hierarchy controls the movement of its parents. Inverse kinematics allows for more intuitive motions by controlling three or more joint’s rotation with a single control. For example, inverse kinematics is used to replicate walk cycles since animators only have to worry about the location of the inverse kinematic controller rather than all of the joint rotations. On the other hand, forward kinematics (FK) follows the hierarchical chain of the rig, so each joint has to be rotated. This gives animators more control over the rig. However, this means that the animator has to position each joint independently of each other. Most rigs incorporate both IK and FK into the rig, allowing animators to switch between the two seamlessly.

The last two terms, skinning and weight paintings are the simplest of terms but the most tedious to implement. Skinning is the process of binding the mesh to the joints. This allows the mesh to move with the joints. The second most vital part of rigging after placement of joints is weight painting. When the mesh is bound to the joints, the weight influence is average between the closest joints. So weight painting allows riggers to dictate exactly how much control a joint has over the mesh.

4.1 Setting Up the Rig

Before the first joint is placed, it is essential to evaluate precisely what the rig is supposed to accomplish before implementing the rigging system. For this project, the expression sheets were examined first since the bloodhound is not a full body character. The expression sheets provide an
idea of how much deformation the bloodhound needs to be able to achieve. Using anatomy pictures of dogs helped lay out where the joints should go. The topology of the mesh was checked once more, to get an idea if the placements of joints would or wouldn’t work. Typically riggers would discuss with animators what exactly they would like out of the rig, where they prefer the locations of controls.

Since this rig is going to demonstrate both direct selections and a traditional rig, the rig uses a different hierarchy chain than a basic rig would use. A broken hierarchy system was used rather than a traditional hierarchy chain. A broken hierarchy rig or chain is one where the rig is broken into sections connected by constraints rather than using a skeletal/joint hierarchy. Common sections to be broken up are the left and right leg, arms, wrist, torso, and neck. For this project, all the joints that control join deformations in the face are broken away from the rig. This allows for more flexible rigs to be created. Broken hierarchy chains also allow for changes to be made to regions without affecting the entirety of the rig, which is a common problem with hierarchy rigging.

Since a broken hierarchy chain is being used, the first set of joints was placed in their correct positions without much care about the rotation axis. The second set of joints was created by snapping them to each joint in the first set of joints. After the second set of joints is created, the first set of joints is deleted. The reason that the first set of joints was created only to be deleted later is that their only purpose was to serve as locators to snap the second set of joints too. When creating a joint chain the orientation changes often between each joint. This might not seem like an issue however if any of the joint orientations is greater than 180 degrees or smaller than -180 degrees this means that orientations are flipped. So, for example, when using a value of 30 degrees into the Y rotation, the animator expects the joint to rotate upward instead the joint rotates in the opposite direction. This is the simplest solution to fixing this problem, plus it keeps the joints history clean.

After the second joint chain has been created the joint were renamed using the prefix "jointOffset" as seen in figure 4.1. It is crucial to have a proper naming convention as it makes life so much easier when searching for certain nodes or writing expressions. Duplicate the renamed joint chain a second time, renaming these joints using the prefix "bind." These joints will be used to bind the mesh to. These joints are parented underneath the jointOffset joint chain resulting in a combined joint chain similar to Figure 4.1. By parenting each of the child joints (bind joint) to their parent joint (jointOffset) zeroes out all the bind joints and allow for secondary/micro controls to be used.
After all the joints were placed with correct orientations came the creations of the controls. Simple NURBS circles were chosen to be the controls because circles are least distracting of all the primitive shapes and most commonly used in other rigs. The best method to set up controls is to create a locator at the location of each of the bind joints. This way, when the controls are parented underneath the locator, the controls possess the same orientations as the joint they will be controlling and everything is zeroed out. Secondly, the controls will be moving around in local space rather than world space. As the name suggests world space is the space relative to the world, whereas local space is the relative space of the object or control. The third benefits of parenting the controls under a locator allows for the controls, mesh, and joints to be separated into three groups.

4.1.1 Eyes

The eyes are the most important feature of any character. Almost all emotions can be conveyed through the eyes. A common misconception is the belief that eyebrows express the majority of emotions rather the eye itself, this is false. Eyelids can tell what the person is thinking, not the eyebrow. Eyebrows have two major brow movements: squeeze and up/down. A lot of expressions can be expressed by a simple movement of raising and lowering the eyebrow. But a more critical movement than the raising and lowering of the eyebrow is the squeezing of the eyebrow. It often overlooked by most animators but the eyebrows squeeze is more important, any degree of movement or expression in the eyebrow requires a squeeze. The squeeze indicates thought for any of the expressions given.

The two primary motions for both the upper and lower eyelid are to widen and close, but
there are many emotions that can be expressed in between these extremes. The upper eyelid shows the level of alertness. The level or position of the upper eyelid is always relative to the iris and pupil, no matter where the eye is looking. If the eye is looking at the ceiling the eyelid moves in relation to the pupil. The level of alertness dictates how much of the iris and pupil are shown. If the character is sleepy, the upper eyelid covers half of the pupil and iris, if the character is alert the upper eyelid widens exposing the entirety of the pupil and iris. The lower eyelid key motion is squint, the orbicularis oculi drive this motion. The orbicularis oculi affect a broad range of area, from the entirety of the lower eyelid to the corners of the upper eyelid. The lower eyelid is key for intensifying the expressions, it especially good for expressing anger, but can be used with other expressions as well.

In order to create the major motion for both eyelids and eyebrow as discussed above, a hybrid rigging approach was used. This means that both a blendshape-rig and joint-rig was created to get the needed deformations for the eyes. Each eye has six controllers for the eyelids, three each for the lower and upper eyelid. Both of the upper and lower eyelid controls are children to the macro controls that allow animators to easily open and close the eyes by translating the control up and down the local y-axis. This is represented by the green semi-circle seen in the right of the image in Figure 4.2.

![Bloodhound Rig: Joints and Traditional controls](image)

For the eyebrow, four controls are needed for each eyebrow since the eyebrow is so large that it is difficult to control with only three controls. Two additional controls were placed behind the eyebrow to control the brow, allowing wrinkle deformations to be created. Additional controllers were placed around the eyelid bag to give more control over the lower portions of the orbicularis oculi. These are beneficial for creating the squint motion of the lower eyelid. All of these controls are considered to be micro or secondary controllers and built on top of these micro controls are blend...
shapes, allowing for better deformations.

### 4.1.2 Mouth

The mouth is the second most important area when it comes to facial rigging. This may be hard to believe, but it is true. Take for example the movie "Rio," [4] all the main characters are birds, so this does not leave much room to create expressions with the beak. Studying Nigel expressions sheet [4], Figure 4.3, notice how little deformations are applied to the beak or mouth, if there is any then it is in the corners of the mouth. Instead, most of the expressions come from the eyes and the posture of the feathers.

![Figure 4.3: Rio: Nigel expression sheet [4]](image)

The mouth can be broken into four significant motions smile, frown, wide, and narrow. These are the four most essential shapes to nail when rigging the mouth. If the rig can nail these four poses, then it can undoubtedly replicate any other poses. Like the eyes, the mouth uses a hybrid of joint-rig and blendshape rig. The blendshape rig will control the four key expressions as well as lip up/down, lip roll in/out, and the cheeks. These are considered the macro controls for the joint rig’s controls. The joint rig will open and close the jaw as well as provide free-form controls for the mouth. The upper lips will have four joint controls for each side of the lip, and three controls for each side of the lower lip with a central control. Typically, there isn’t a need for so many micro...
controls but due to the bloodhound’s mouth being so large extra controls had to be added.
Chapter 5

Direct Selection

Direct selection controls is a relatively new but straightforward concept. The NURBS controls that animators would typically use to animate is replaced by a mesh controller that deforms to the mesh, eliminating the needs for control curves. However, this simple concept can be hard to implement. The process to create direct selection controls is illustrated in Figure 5.1.

![Figure 5.1: Direct Selection Diagram](image)

To construct direct selection controls, the character needs joints that are already skinned. It’s best if there aren’t any controls already created for the rig but since the bloodhound rig was built with controls, all the controls had to be deleted. Following the diagram, the mesh is duplicated
so that the necessary faces can be extracted from the duplicated mesh. The extracted faces will
become the mesh for direct selection controls. Next, the deformer (transform geometry), is going to
glue the extracted faces to the original mesh. To do this, the deformer needs two parameters, first
the outMesh from the original mesh and second the matrix of the extracted faces. Every object or
mesh in Maya is built from several nodes.

- a creation node that records the options that created the object
- a transform node that records how the object is moved, rotated, and scaled
- a shape node that stores the position of the object’s control points

From the shape node of the original mesh, the shape output (outMesh) needs to connect to
the deformer input geometry port, allowing for any changes to the original mesh to pass on to the
deformer, which then passes on to the extracted faces shape node, resulting in the extracted faces
to be influenced by any deformation made to the original mesh. It is important to note that the
original mesh history needs to be deleted. The initial shape of the original mesh before any tweaks
or size changes will be used as the direct selection face. If the history is not deleted, this will result
in the creation of the incorrect direct selection control. For the extracted faces positions to control
the deformer, the faces’ transform node (matrix) has to connect to the deformers transform node\(^1\).
After the connection is made to the deformer, the extracted faces control the shape node of the
direct selection control. All that is left to do is connect the control to the joint or blend shape.

5.1 Interface

Due to the repetitive process of setting up direct selection controls for each of the joints
within the face, a graphical interface was created. The purpose of this interface was to allow users
to quickly create direct selection controls and connect the controls to the joint in one go, as seen
in Figure 5.2. The interface is broken into four sections. The first sections, allow users to input
the mesh. There is also a checkbox option, allowing the user to tell if control is going to be the
child node of a locator or group. The checkbox is crucial because it tells the program whether to
use world matrix or inverse world matrix. The next sections, faces, allows users to input the faces
used for direct selections. After all the faces have been selected and imported, a list of the faces

---

\(^1\) Matrix is a vector array that contains the position, rotation and a scale of any object within Maya.
will appear in the text field. The third section, allows the user to input the joints that are going to be controlled. It’s important to note all the direct selections controller and deformer are going to be named after the joints followed by a prefix, so it is essential that the joint appropriately named. The fourth section allows users to determine how the controller will be connected. There are three options, point, rotate and scale constraint. The user can select as many as they want. If both point and rotate constraint are selected, rather than constraining them with separate constraints a parent constraint is used instead.

![Figure 5.2: Interface for Direct Selection](image)

After running the interface a newly created direct selection controller is at the bottom of the outliner. Rather than grouping or parenting the controller back into the hierarchy system for all the joints and controllers, it was placed outside the hierarchy system because each rigger has different preferences when it comes to rigging. Some riggers like to parent the joints underneath the
controller, like a linear hierarchy approach. Others like to use a broken hierarchy system, like the one used for this rig where the mesh, joint, and controllers are all separated into different groups so that it helps not to have double transformation. All that left is for the rigger to parent the controls into the rig hierarchy chain.

5.2 Implementation

The direct selection controls interface was created with python since Maya provides extensive documentation about Python, and Pymel allows access to Maya nodes. The second benefit to using python is that it allows for flexible, object-oriented design. The two primary functions needed for the direct selection interface was a transformGeometrySetup and deletFaces function.

5.2.1 Transform Geometry Setup

The following code sets up the deformer that connects to the shape node of the direct selection control. For this, a transform Geometry node is created. The node needs two attributes, the "inputGeometry" from the original mesh and the matrix of the extracted faces. The input geometry can be a NURBS surface, NURBS curve or mesh data. For this project, the mesh data is connected to the outMesh attribute of the transformGeometry node. As for the matrix, depending on whether the direct selection controls is going to be a parented or grouped under something, dictates whether inverse world matrix or world matrix is going to be used. If the control is not going to be grouped or parented to another object, then it’s acceptable to use world matrix. However, if the control is going to become the child of an object then inverse world matrix has to be used otherwise there will be double transformation. Double transformations are when skin points are subjected to the action of a joint more than once [2].

5.2.2 Delete Faces

The following code takes the unselected faces of the mesh and deletes them. The function uses a combination of Open Maya and standard Python. Open Maya is needed to get the faces of the original mesh so that they can be used for making comparisons. As for the rest of the code it is standard Python. The functions need two lists and an empty list. The first list stores all of the faces from the original mesh. The second list, retains all the selected faces chosen by the user and
the last list is an empty list that will store the deleted faces. A while statement is used to make a comparison is made between every source face to the selected faces. If the two faces are a match then it the iteration skips over them otherwise the current source face is added to the empty list. After the entire source list checked the once empty list is deleted, removing all the unselected faces.
Chapter 6

Results

The final results of direct selection controls are illustrated in Figure 6.1. Created with 181 joints and 77 controls in the face, only seven controls are visible. By eliminating all the controls, we can see the bloodhound. Controls do not clutter the bloodhound, there is no longer any need to select minute controls, and no need for an extra window to see the scene without any controls, unlike Figure 6.2. It simple and clean.

Only three macro controls and four micro controls remain visible: the world and two eye controls, Figure 6.1. These controls were intentionally left visible because its easier for animators to see where the eyes are pointed if there is a marker for them to use. The marker also makes it easier for the animators to lock the control on another character face. The goal of our work is to create an animator friendly rig, one that’s both reliable and allows the animator to get a full range of emotions from the character. We have clearly achieved this goal with direct selection controls. However, if the animator should ever wish to see the controls, they can do so by turning them on within the attributes of the world controller.

To compare direct selections controls to a NURBS control curve rig both rigs were used to pose the character in three different poses, shown in Figure 6.3. The test was used to see if there was any difference in the speed of animating the character with direct selections controls over traditional controls. Each pose takes place at intervals of 24 frames on the timeline; a time lapse was also taken to record how long everything took, Figure 6.4 and 6.5. The trial showed that animating the character with direct selection controls was faster than using traditional controls by two minutes. It took 12.24 minutes to animate the character with direct selection controls whereas
it took a little over 14.47 minutes to animate the character with traditional controls. As for selecting the controls, direct selection controls was much easier than using the traditional controls but more camera rotations and movement were required when trying to select the lower jaw controls for the direct selection controls, than NURBS control curves. Most importantly, the quality of animation improved compared to animating with traditional controls. By taking advantage of being able to clearly see the character, the poses achieved were much more dramatic and expressive compared to the poses achieved with traditional controls.

There were two distinct downsides to using direct selection controls. First, a novice animator or someone who has never animated before might not understand how direct selection controls work, or where controls are generally placed. They would not understand how much of a benefit it is to have a control free mesh and might possibly become frustrated with this method of animating. And second, there is a small amount of lag when working with direct selection controls. The lag is not disruptive enough to animators that they cannot animate with direct selection controls. But it can be noticeable when first selecting a control and moving it around.

Figure 6.1: Rig setup with direct selection controls
Figure 6.2: Rig setup with traditional controls

Figure 6.3: Three different poses of the dog
Figure 6.4: (A) Traditional Rig (B) Direct Selection Controls

Figure 6.5: (A) Traditional Rig (B) Direct Selection Controls
Figure 6.6: All the selectable faces in the direct selection controls rig
Chapter 7

Conclusion

The interface proposes a framework to automate creating direct selection controls. The design allows for the creations of direct selection controls with little to no changes to the rigging process, allowing riggers to create controls faster than creating NURBS control curves.

Direct selection controls are more than hidden controls. Direct selection controls allow animators to feel like they are drawing the pose like 2D animation instead of having to plugin values, or clicking and dragging the controls with a mouse. Direct selection controls allow animators to be more expressive than they have ever been before and push the pose of a character. This is all due to direct selection controls allowing for the animator to see poses that they would not otherwise see if the controls were blocking the view.

Like any tools or software, there is still room for improvements both to the method of creating direct selection controls and the interface to generate the controls. The two main improvements that could be made to this project are first the lag and second macro and micro controls. There could be two reasons for the delay. First not enough power in the computer. The standard machine at Pixar is powered by a 2.3GHz, 16-core Intel processor with 64GB of RAM and a 12GB Nvidia Quadro M6000 [5]. That’s almost twice the power that the machine currently being used for this project. Second, and probably a more realistic reason for the lag, is due to the current setup for direct selections controls. Because the original mesh is influencing all of the direct selections controls, this is probably the source of the lag. An idea to fix this issue is to break the influence of the original mesh on the control. Instead, the control could be directly snapped or glued to the mesh. Now the controller would be affecting the joint which in turn is affecting the original mesh,
almost like a cycle. An illustration of this new method is proposed in Figure 7.1; this method could eliminate the lag or make it worse.

![Alternative Direct Selection Controls Diagram](image)

Figure 7.1: Alternative method to direct selection controls

As mentioned direct selection controls work great with macro controls but not so much with micro controls. Micro controls represent a second layer of control that are child controls to the macro controls and often hidden when animating. Because micro controls control a smaller region than macro controls, it is challenging to create direct selection controls for them. The micro controls are overlapping with the macro controls resulting in possible double transformation. Two options to solve this problem are to either use a GUI Picker or use a NURBS control curves. The GUI picker would allow for the controls to remain invisible, but this requires the use of an extra interface or window. The second method to address this issues is to use NURBS control curves. Animators could toggle the visibility of these controls, which would not be too inconvenient. However, this defeats the purpose of direct selection controls.

In conclusion using direct selection controls has a great deal of potential. It allows animators to work faster, clearly see the character and find poses that they would not generally find with traditional controls blocking the view. It has already been proven to work in two major animations studio: DreamWorks, and Pixar. So there is a significant possibility that this could become the norm within other animation studios and even within the Maya package itself.
Bibliography


