CREATING DYNAMIC GRASS FOR TARTAN TROUBLES

Nicholas Mcelveen
Clemson University, nmcelve@gmail.com

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

Part of the Film and Media Studies Commons

Recommended Citation
Mcelveen, Nicholas, "CREATING DYNAMIC GRASS FOR TARTAN TROUBLES" (2013). All Theses. 1664.
https://tigerprints.clemson.edu/all_theses/1664

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.
CREATING DYNAMIC GRASS FOR TARTAN TROUBLES

A Thesis
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
Nicholas McElveen
May 2013

Accepted by:
Dr. Timothy Davis, Committee Chair
Dr. D. E. Stevenson
Professor Tony Penna
ABSTRACT

In computer animated movies, the setting and its environmental elements aid in storytelling. In Tartan Troubles, dynamic grass was created as a part of the highlands of Scotland set. This thesis discusses the importance of environmental elements in animation as well as the terminology, procedures, and techniques used to produce dynamic grass as seen in the animated short, Tartan Troubles. The scalability, control, and flexibility of the dynamic grass pipeline make this effect applicable for various solutions in computer animation.
DEDICATION

To my parents who invested in me, and my family that supports me in all that I do, this thesis is dedicated to you. In the moments where things seem unclear, you give me guidance and encouragement that make my worries less challenging. Thank you.
ACKNOWLEDGMENTS

I want to thank my advisor, Dr. Timothy Davis, for his guidance throughout the Digital Production Arts program. As an undergraduate at Clemson, I set a goal to complete this MFA program. It would not have been possible without Dr. Davis’ patience and continued support.

I would also like to thank Professor Tony Penna and Dr. D. E. Stevenson for sharing their insight and criticism. I appreciate their time and willingness to sit on my thesis committee.

The South Carolina Commission on Higher Education also deserves to be recognized for the Fellowship I was awarded in support of pursuing my masters at Clemson.

I would also like to acknowledge Clemson University’s tuition assistance program for employees. It has played a vital role in allowing me to complete the Digital Production Arts program.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2 BACKGROUND</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Environment</td>
<td>3</td>
</tr>
<tr>
<td>2.2 The Story of Tartan Troubles</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Tools, Terms and Techniques</td>
<td>9</td>
</tr>
<tr>
<td>2.3.1 Maya</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Fields</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Hair Systems</td>
<td>10</td>
</tr>
<tr>
<td>2.3.4 Fur</td>
<td>11</td>
</tr>
<tr>
<td>2.3.5 Adobe Photoshop</td>
<td>15</td>
</tr>
<tr>
<td>2.3.6 MEL Scripting</td>
<td>15</td>
</tr>
<tr>
<td>CHAPTER 3 IMPLEMENTATION</td>
<td>18</td>
</tr>
<tr>
<td>3.1 Dynamic Grass Effect in Tartan Troubles</td>
<td>18</td>
</tr>
<tr>
<td>3.1.1 Scene Set-Up</td>
<td>19</td>
</tr>
<tr>
<td>3.1.2 Air Fields</td>
<td>19</td>
</tr>
<tr>
<td>3.1.4 Cliff Geometry</td>
<td>20</td>
</tr>
<tr>
<td>3.1.5 Fur Attribute Image Map Templates</td>
<td>22</td>
</tr>
<tr>
<td>3.1.6 Hair System Set-Up</td>
<td>23</td>
</tr>
<tr>
<td>3.1.7 Hair System Settings</td>
<td>24</td>
</tr>
<tr>
<td>3.1.8 Maya Fur</td>
<td>27</td>
</tr>
</tbody>
</table>
3.1.9  Fur Attribute Maps........................................................................................................30
3.1.10 Wind Animation .............................................................................................................34
3.2  A Field of Grass ................................................................................................................35
3.3  Scene Set-Up with MEL ....................................................................................................39

CHAPTER 4 RESULTS .............................................................................................................44

4.1  Grass in Tartan Troubles ..................................................................................................44
4.2  A Field of Dynamic Grass ...............................................................................................46

CHAPTER 5 CONCLUSION AND FUTURE WORK ..................................................................49

REFERENCES ..........................................................................................................................50

APPENDIX A DYNAMIC FUR FX MEL SCRIPT ..................................................................51
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td><em>Braveheart</em> in a desert would change the story [IMDB95].</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td><em>Pirates of the Carribean: The Curse of the Black Pearl</em> would be odd without water [IMDB03].</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td><em>Happy Feet</em> would not be the same set in a zoo [IMDB06].</td>
<td>5</td>
</tr>
<tr>
<td>2.4</td>
<td>Patches hears the sound of bagpipes.</td>
<td>6</td>
</tr>
<tr>
<td>2.5</td>
<td>Patches finds McDouling.</td>
<td>6</td>
</tr>
<tr>
<td>2.6</td>
<td>Dreaming about tug of war.</td>
<td>7</td>
</tr>
<tr>
<td>2.7</td>
<td>McDouling in anguish.</td>
<td>8</td>
</tr>
<tr>
<td>2.8</td>
<td>McDouling guarded by underwear.</td>
<td>8</td>
</tr>
<tr>
<td>2.9</td>
<td>Sulley from <em>Monsters, Inc.</em> [IMDB01].</td>
<td>12</td>
</tr>
<tr>
<td>2.10</td>
<td>Sulley from <em>Monsters, Inc.</em> [IMDB01].</td>
<td>13</td>
</tr>
<tr>
<td>2.11</td>
<td>Puss in Boots and Donkey from <em>Shrek 2</em> [IMDB04].</td>
<td>13</td>
</tr>
<tr>
<td>2.12</td>
<td>Puss in Boots from <em>Shrek 2</em> [IMDB04].</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Wind revealing McDouling’s boxer shorts.</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>A screenshot of the modified cliff.</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>Sequence of steps for reducing geometry for the left grass plane.</td>
<td>22</td>
</tr>
<tr>
<td>3.4</td>
<td>Planes for left and right grass patches.</td>
<td>23</td>
</tr>
<tr>
<td>3.5</td>
<td>Painted hair system for left grass patch.</td>
<td>24</td>
</tr>
<tr>
<td>3.6</td>
<td>Attribute values for hair system.</td>
<td>26</td>
</tr>
<tr>
<td>3.7</td>
<td>Fur description attributes and values.</td>
<td>28</td>
</tr>
<tr>
<td>3.8</td>
<td>Illustration of Tip Curl attribute [AUT009].</td>
<td>29</td>
</tr>
</tbody>
</table>
Figure 3.9: Tip Color attribute map.................................................................30
Figure 3.10: Base Color attribute map.............................................................31
Figure 3.11: Length and Baldness attribute map.............................................31
Figure 3.12: Adobe Photoshop brush settings...............................................33
Figure 3.13: Grass where characters meet....................................................36
Figure 3.14: Length and Baldness Map for full cliff fur description..............37
Figure 3.15: View of full cliff hair system, Fur Feedback, and Collide Sphere. ....38
Figure 3.16: Color map for full cliff fur description.......................................39
Figure 3.17: Screenshot of GUI for MEL script..............................................40
Figure 4.1: Introduction of Patches the dog.....................................................45
Figure 4.2: McDouling struggles for freedom from Patches’ bite....................45
Figure 4.3: Shot of grass and kilt before strong wind....................................46
Figure 4.4: Shot of grass and kilt affected by strong wind............................46
Figure 4.5: Full field of grass technique applied to cliff...............................47
LIST OF TABLES

Table 3-1: Keyframes for right grass patch Tip Curl attribute. ................................................29

Table 3-2: Keyframe and values for Air Field Magnitude attribute........................................34
CHAPTER 1
INTRODUCTION

Movie theater audiences enter dark rooms, crowded with strangers, to be entertained by a story that unfolds in front of them. The images presented in sequence inform viewers of the setting, characters, and plot that move the story forward. In each image exists information about the environment, including lighting, weather, and props. The audience can sense the tension of a tornado harvesting houses from neighborhood streets or the peacefulness of a gentle breeze as trees sway and leaves skip in the wind. These images affect the mood and energy surrounding the region of interest in every shot, further supporting the action.

Environmental elements play an active role in the presentation of story. Just as novelists paint vivid pictures with words and affect mood with diction, cinematography employs multiple senses simultaneously to engage the audience. Before the final cut is released to theaters, however, a great deal of planning is required to prepare the production for filming. Costumes are designed, characters are developed, and the setting is sketched many times. No matter the budget of the production, preproduction planning is common among film, animation and plays.

In the industry of computer-animated movies, every detail concerning place, time, and characters is developed with the aid of computer software, even the ground on which the characters stand. Each aspect of an animation is researched, designed and produced by a group of artists, animators, and directors. This thesis will review the production process for creating dynamic grass in the Clemson
University Digital Production Arts’ SIGGRAPH animation, Tartan Troubles. The processes and results presented represent work completed in a small production environment. Chapter 2 discusses the background of Tartan Troubles, the purpose of grass in this animation, tools used to produce the dynamic grass effect and the terminology related to each tool. Chapter 3 walks through implementing the dynamic grass effect and apply this effect to a large field while handling collisions between characters and grass. Chapter 3 also discusses automating scene set-up for the grass effect through MEL scripting. Chapter 4 reviews the final shots produced for Tartan Troubles where the grass effect is included, and a final render of the field of grass. Additional areas of research and conclusions are discussed in Chapter 5.
The environment, or setting, is an important component of every story. For *Tartan Troubles*, grass was an integral component of the environment. A number of software programs were used to create the dynamic grass effect to help tell the story of *Tartan Troubles*. Chapter 2 discusses all of these aspects, their purpose and the terminology synonymous with each.

### 2.1 Environment

Visual elements provide viewers with information about the natural world. Trees, leaves, and even light posts give visual cues that aid in defining the conditions observed. Similarly in animation, every detail of each shot, from environment to camera angles and composition, support the story.

When audiences see a man holding an open umbrella above his head, the anticipation of precipitation is paramount. If this scene takes place in Central Park on a sunny afternoon, the purpose of the umbrella is questioned. The audience will be confused about the meaning of the umbrella and the purpose of this shot in relation to the story. Without rain, the environment and character present two different messages. The character’s actions portray rain is coming. The environment conveys beautiful weather or no expectation of rain. The character appears out of place and the environment detracts from the plot, disrupting the visual harmony that would exist with a supporting or natural environment [SIJI05]. The movie,
Braveheart (Figure 2.1) set in the Saharan Desert, Pirates of the Caribbean (Figure 2.2) without the ocean, or Happy Feet (Figure 2.3) shot in a zoo would all be affected in a similar way. The environment is therefore important to suspending the audience's disbelief, placing the action in an appropriate context and drawing in viewers to be moved and entertained.

Figure 2.1: Braveheart in a desert would change the story [IMDB95].
The combination of characters and environment are first realized in storyboards. They serve as a guide or plan for each team involved in the production. As the storyboards are reviewed, the ideas of the director are realized along with different supporting elements. The next section will present the story of Tartan Troubles, and some of the storyboards for this project.
2.2 The Story of Tartan Troubles

Awakened by the sound of bagpipe music, Patches, a West Highland White Terrier, perks up to find the origin of the sound (Figure 2.4). He walks over the hills to find a Scottish man (McDouling) serenading the town below (Figure 2.5). Patches walks over to McDouling and continues enjoying the music.

Figure 2.4 Patches hears the sound of bagpipes.

Figure 2.5: Patches finds McDouling.
Mesmerized by the wonderful playing, Patches daydreams of years past when he played as a young pup (Figure 2.6). Tug of war was one of his favorite memories. He jumps to bite the rope in front of him and the music stops (Figure 2.7). McDouling shakes and screams trying to free himself from the dog. This game of tug of war seems more like an attack. McDouling snaps Patches out of the dream, freeing himself from the dog’s bite with all body parts intact. Later that day, McDouling resumes playing his pipes, but this time he is guarded by underwear (Figure 2.8).

Figure 2.6: Dreaming about tug of war.
Figure 2.7: McDouling in anguish.

Figure 2.8: McDouling guarded by underwear.

To produce this animation, models for a cliff and a town were designed and built, McDouling and Patches were created, storyboards were drawn to plan the sequence of shots, and a color palette was determined. Set in the highlands of Scotland, known for cliffs and hills, the animation contains a significant amount of
grass. Wind is another element of this environment affecting the movement of the grass and McDouling’s kilt.

To create computer-generated grass, either Maya Fur or Paint Effects could have been used. Maya Fur proved to be the better alternative, since it is able to cover large areas of geometry in a simple process and is linked to Maya Hair, which can be used to drive movement. Hair systems are affected by changes in translation and fields; therefore, simulating wind blowing over grass was a logical application for this workflow to achieve dynamic, or natural, grass animation.

2.3 Tools, Terms and Techniques

A number of software applications and concepts were used to complete the grass rendering and animation. The following sections will discuss terms related to Maya, which controlled all of the hair and fur features; Photoshop, which was used to help control color; and MEL Scripting, which could be used to simplify the workflow of producing the dynamic grass effect.

2.3.1 Maya

The dynamic grass effect, as well as the Tartan Troubles animation, were developed using Maya 7. The controls and steps outlined in this thesis are applicable to Maya 2011. The following two terms are used specifically in reference to Maya.

Node: Maya is built on a node-based architecture. All elements in a Maya scene are represented in MEL script by a collection of nodes, inter-connected to produce the desired result. Fields, hair systems and fur all have corresponding nodes [AUTO09].
**NURBS**: NURBS, Non-Uniform Rational B-Splines, is a type of geometry used to create 3D curves and surfaces in Maya. NURBS geometry uses a mathematical representation of Bezier Curves to construct many types of organic 3D forms [AUTO09]. The cliff where McDouling and Patches meet is a NURBS plane. The dynamic grass is also attached to NURBS planes derived from sections of the cliff.

### 2.3.2 Fields

Wind, vortices, gravity, and turbulence are represented as force fields in Maya. These computer simulations of real-life phenomena are used to re-create real-world occurrences. For example, dust or leaves can be swirled around through simulation. From hurricane winds to the breath that blows a dandelion, fields play a role in multiple effects. Wind is a natural environmental element in the Scottish highlands that needed to be incorporated in the animation. From the opening credits where flags are whipping in the wind, to the introduction of Patches, to the final shot, wind is present throughout the story.

### 2.3.3 Hair Systems

In order for computer-generated characters to mimic humans, hair is a necessary element. The complex computations of hair movement, color, shading and texture are all included in Maya Hair systems. Traditionally, hair systems are used for simulating the effects of human hair, but the components of Maya are often manipulated in creative ways to achieve special effects. In this thesis, a hair system is used to drive the dynamic grass effect. The following terms are use in reference to hair systems.
**Follicles:** A follicle is a tubular sheath that encases the lower part of the hair shaft. Each follicle contains one NURBS curve that represents the position of the hair in that follicle. The follicle appears similar to a red rivet at the base of the curve. A collection of hair follicles, similar to the follicles of the human scalp, comprises a hair system. Three different types of curves can represent a follicle: static curves, dynamic curves, and passive fill curves [AUTO09]. Static curves have a fixed shape, but were rarely used in this project.

**Dynamic Curves:** Dynamic curves are NURBS curves that are affected by other objects in the scene. Changes in position and orientation are calculated based on events that take place over time including: collisions, forces, and changes in translation due to a parent object. Using forces specifically in this grass implementation, dynamic curves drive the fur animation.

**Passive Fill Curves:** These NURBS curves are similar to dynamic curves, but are animated based on surrounding dynamic curves. The use of passive fill curves reduces the number of dynamic calculations required for rendering hair sequences, which also reduces the amount of time required to render each frame. In the implementation of a field of grass affected by Maya forces, passive fill curves are used to optimize performance while achieving the same results.

### 2.3.4 Fur

Fur is a component of Maya dedicated to simulating realistic animal fur. Lighting, shading, texture, and color are a few of the controlling attributes for achieving realistic or stylized fur. In animation, Sulley from *Monster’s Inc.*, seen in
Figures 2.9 and 2.10, as well as Puss in Boots from *Shrek 2*, seen in Figures 2.11 and 2.12, are examples of CG characters where fur is applied.

Figure 2.9: Sulley from *Monsters, Inc*. [IMDB01].
Figure 2.10: Sulley from *Monsters, Inc.* [IMDB01].

Figure 2.11: Puss in Boots and Donkey from *Shrek 2* [IMDB04].
Fur plays a different role in Tartan Troubles. Fur is designed to resemble tall blades of grass swaying back and forth in the highland winds. In the closing shot, the dynamic grass effect indicates a strong wind that blows McDouling’s kilt, revealing his boxer shorts.

**Fur Description:** The fur description is a node that contains all of the attributes and values for a particular type of fur. It can be attached to any NURBS or polygon geometry [AUTO09]. Each fur description represents a patch of grass in the final animation.

**Fur Feedback:** Fur feedback is the visual representation of a fur description applied to geometry in a Maya scene. The density of feedback in Maya is usually greatly reduced to conserve computing resources that allow interactive execution in Maya.

**Maps:** Also known as attribute maps or textures, maps are images used to control the settings and appearance of node attributes. In this thesis, maps control fur description attributes including color and the location of fur within the area covered.
by a fur description. All of the attribute maps used in this project were created in Adobe Photoshop.

2.3.5 Adobe Photoshop

Maya offers built-in tools for painting textures, but those tools are not as robust as external image editors such as Adobe Photoshop. Photoshop allows photographers to correct over-exposed pictures and adjust colors. Photoshop is also used in the graphic arts community, providing a digital canvas for matte painting, compositing, and design. In this thesis, Photoshop CS4 was used for painting image maps with a workflow that promotes reuse and adjustments. These image maps were then saved into a Maya project directory for fur attributes and used to gain fine control over Maya Fur.

2.3.6 MEL Scripting

Scripting with the Maya Embedded Language (MEL) is the technique used to automate steps in a Maya workflow, and to create interfaces for plug-ins extending Maya functionality and controls for animation. MEL scripting is a desired skill for technical directors and 3D artists [WILK05]. Developing tools that other artists can leverage is valuable in production studios. Automating repetitive tasks with MEL streamlines workflows and saves time. In this thesis, MEL scripting is used to apply settings to fur descriptions, expose controls for quicker implementation and reduce keystrokes through a GUI built specifically for the dynamic grass effect.
**Variables:** Variables are temporary containers for integers, floats, and string values that are needed later in the MEL script [WILK05]. Variables are used throughout MEL to facilitate control of fur description and hair system attributes.

**Option Variables:** These variables exist across Maya sessions and store information based on user profiles, similar to system preferences, where each user on a computer can have different settings for such options as screen resolution and default web browser. Maya option variables work the same way, storing information specific to Maya and the state in which a user last left the software. Option Variables will be used to allow fur description and hair system settings to persist between Maya sessions; therefore, the MEL script is initialized with the last set of values used.

**Procedures:** Procedures are a collection of MEL statements that are grouped by a name or signature [WILK05]. Developing procedures for the Dynamic Fur FX MEL script functionality is the main part of MEL scripting for this project.

**Functions:** A function is equivalent to a procedure, but returns a value at the end of execution.

**Commands:** MEL commands are functions and procedures with flags for specifying execution parameters that are native to Maya. The commands are used inside the procedures developed for the Dynamic Fur FX MEL script. They take the values passed as parameters and create or update elements in the Maya scene.

**Source Script:** To source a script is similar to including a class file in the Java programming language. Some MEL scripts are not loaded when Maya opens initially. In order to use functions from another script, the user must source that script,
loading it into the Maya scene to define its functions. Some scripts from Maya are sourced to support the functionality of the dynamic grass effect MEL script.
CHAPTER 3
IMPLEMENTATION

The purpose of this chapter is to give an overview of the process for creating the dynamic grass effect in *Tartan Troubles*. With the background information in mind, we proceed to discuss the technical details of the fields, hair system, and fur descriptions used to produce the animated grass sequence in the closing shot of *Tartan Troubles*. In closing we automate tasks in scene set-up for applying grass with MEL scripting.

3.1 Dynamic Grass Effect in *Tartan Troubles*

In the final shot of *Tartan Troubles*, McDouling’s kilt is blown upwards by a forceful wind that reveals his boxer shorts (Figure 3.1). The tall grass that appears to his left and right must convey the same force of wind to support the kilt’s animation. This section reviews the steps for implementing this effect.
3.1.1 Scene Set-Up

The foundation model for the grass is the cliff where Patches and McDouling meet. Initially, the Maya scene file contained the entire Tartan Troubles set. The first step in setting up this scene was creating the Air Fields to simulate wind, followed by the hair system, and the attachment and design of the fur description to simulate grass.

3.1.2 Air Fields

To simulate wind, the project used the following Air options: Magnitude, Direction and Speed. The strength of the air field was controlled by the Magnitude attribute. As Magnitude values increased, the force applied by the air field increased as well. The initial Magnitude was set to 0.5. This value did not impact the grass much at the first frame of the scene file. When the action began, the Magnitude increased to 6.487. This value was greater than 0.5, but with the total range of Magnitude spanning -100 to 100, this value was relatively small and only intended to move the grass slightly. It was not until McDouling’s boxer shorts were revealed.
that the Magnitude climbed to 59.340, providing great force to the dynamic hair curves.

The Magnitude worked with the air field Direction to control where the curves were pushed. From the front of the cliff, the Fur Feedback was positioned down the negative Z-axis (i.e., Z was set to -1.000). The desired effect was to blow the grass backwards; therefore, the Y Direction value was left at 0.000. Based on the way the Fur Feedback rolled over the cliff, -0.250 was set as the X Direction value to adjust the direction the grass was blown to fit its position on the cliff. The Z Direction was the main Direction in which the hair and fur should move; thus, this value was set to a larger value than X and Y.

How quickly the dynamic curves react to the force of the air field was controlled by the Speed attribute. The full range for Speed was 0.000 to 200.000. The grass was to behave as if it were affected quickly but with natural, believable force. The value of 100.000 was chosen since it is a mid-range value that would yield an effect on the grass that is not overly intense.

### 3.1.4 Cliff Geometry

In the scene file, the main regions of interest, the Scottish town and cliff, were separated geometrically. The cliff was composed of a NURBS plane with displaced CVs, which stretched the UV spans in the process. In order to set up the hair system with enough curves to animate the grass effectively, the CV density was increased in specific locations. Theoretically, rebuilding the surface and increasing the number of CVs would have sufficed, but doing so would have slowed the Maya interface refresh rate. To resolve this issue, the cliff plane was modified to reduce the geometry.
necessary for each hair system. First, the cliff plane was duplicated and moved to a new location. The geometry was then split into smaller surfaces, which were removed until only the edge of the cliff remained (Figure 3.2).

Figure 3.2: A screenshot of the modified cliff.

Seven patches of grass were applied to the cliff for the final shot of Tartan Troubles. These patches provided an adequate field of grass to allow for horizontal pans of the area. Only two patches of grass were actually included in the final shot; therefore, two hair systems were needed: one to the left and another to the right of the center of the cliff. The modified cliff provided an identical model with dense UVs and geometry to apply hair systems to specific areas, while the remaining NURBS plane was removed (Figure 3.3) for efficiency purposes.
3.1.5 Fur Attribute Image Map Templates

To create the grass texture, Maya’s 3D Paint Tool was used to texture the plane for later processing in Photoshop. The right edge of the plane was painted red, the top sky blue, the bottom edge navy, and the left edge purple. Further, a green outline was painted to signify grassy areas, as seen in Figure 3.4. These markings served as reference points for orientation in Photoshop and where grass should appear. A similar process was used to paint the second NURBS plane.
3.1.6 Hair System Set-Up

In addition to fur, Maya hair systems were also used to simulate grass. Using the Artisan and Paint Hair Curve tool in Maya, hair follicles were painted on both sides of the modified cliff, as shown in Figure 3.5. A separate hair system was painted on each plane. The follicle density for U was denser than V at a ratio of 2 to 1. The direction in which the wind blew the grass was along the U Direction or Z-axis; therefore, more collisions occurred in that direction and more follicle density was needed. With more hair curves along U, the grass simulation could better detect collisions and avoid intersections that would be noticeable to the audience from the camera’s perspective.
3.1.7 Hair System Settings

Once all dynamic systems were in place, simulation of grass movement could proceed. The air fields and hair settings interacted to create the calculated movement of the dynamic grass effect. The hair system employed a set of attributes that controlled the animation. The attributes most significant for the final shot of Tartan Troubles were collide, stiffness, and drag. The collide attribute controlled the
calculations of the hair curve collisions with the NURBS plane. In the physical world, when a blade of grass touches the ground, it stops without penetration. Similar behavior was needed for hair curves and the ground plane in the simulation to keep all grass blades above the cliff; therefore, collide was activated. Natural grass also appears to have some stiffness as well. Blades do not lie flat on the ground. Instead, they grow up from the ground and move when the wind blows without completely succumbing to its force. The stiffness attribute controlled how the dynamic curves flex or bend and was set to 0.35. Drag simulates friction with the air and helps stabilize the simulation. To slow the grass movement and impact of the air fields, drag was set to 0.0500. This figure is a relatively low value because the grass blades should react quickly, but with less movement than, say, a strand of thread. The entire set of attributes and values are listed in Figure 3.6. With the hair system in place, taking care of Maya fur was the next phase in the dynamic grass pipeline.
Figure 3.6: Attribute values for hair system.
3.1.8 Maya Fur

For the Maya fur description used in simulating grass, the values listed in Figure 3.7 were used. The fur description settings for the left and right grass patches were identical. The hair system was linked to the corresponding fur description on each NURBS plane. A number of the fur description attributes were used to design the grass. Most notable were the Base Color, Tip Color, Length and Baldness attributes, which were all controlled by image maps. The Tip and Base Color attributes controlled the color of the grass. Applying a different color map to each attribute provided a gradient from base to tip of each blade. The Length and Baldness Maps controlled the height and location of the grass on each respective plane. The Length Map adjusted the length or height of the grass scaled by the value of the corresponding pixel. White represented full length, medium gray represented half of the specified length or a scale of 0.5, and black pixels represented a length of 0. A similar approach was employed for the Baldness Map, except this map controlled the location of the grass on the surface. In the Details section of the fur description, noise was added for variation in the height of the rendered grass. Once all the maps were added, these attributes were baked such that the Fur Feedback could interpret the information provided and convert the maps to simple file textures.
Figure 3.7: Fur description attributes and values.

The amount of curl at the top of each blade was controlled by the Tip Curl attribute. To bend the grass tips on cue, the Tip Curl was keyframed to bend back
and forth with Magnitude to enhance the windblown grass effect further. An illustration of the effect of Tip Curl on the grass is shown in Figure 3.8.

![Figure 3.8: Illustration of Tip Curl attribute [AUTO09].](image)

Setting the value to 0.5 indicates no curl; therefore, the values used here flipped back and forth from less than to greater than 0.5 to bend the grass tips back and forth as the wind blew. Table 3-1 lists keyframes and the respective Tip Curl values used. With the fur behavior set, making the fur look like grass was the next step in creating the dynamic grass effect.

<table>
<thead>
<tr>
<th>Keyframe</th>
<th>Right Fur Patch Tip Curl</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>285</td>
<td>0.343</td>
<td>Initial curl towards cliff edge</td>
</tr>
<tr>
<td>365</td>
<td>0.588</td>
<td>Minor backward bend due to breeze</td>
</tr>
<tr>
<td>405</td>
<td>0.353</td>
<td>Less movement before big gust</td>
</tr>
<tr>
<td>472</td>
<td>1.529</td>
<td>Slight delayed bend with strong wind</td>
</tr>
<tr>
<td>529</td>
<td>0.930</td>
<td>Less bend due to decreased wind</td>
</tr>
</tbody>
</table>
3.1.9 Fur Attribute Maps

Once the painted file textures for the cliff were created in Maya, they were exported to Photoshop for further processing. Photoshop image manipulation features facilitated map creation for the fur descriptions in terms of simplicity and efficiency. The Tip Color Map shown in Figure 3.9 was painted with two layers. The background layer was completely black and the foreground layer contained the color for the tip of the grass. Photoshop allowed the foreground layer of this image to be duplicated and changed to another color as shown in Figure 3.10, the Base Color Map, and Figure 3.11, the Length and Baldness Map.

Figure 3.9: Tip Color attribute map.
The images are identical except in color. These maps applied to the fur
description shaded the grass green on the bottom, which faded into a yellowish
green at the tip, only visible where white was included in the Length and Baldness
Map. Creating these maps in an application external to Maya also aided in adjusting
the maps when necessary. With the images broken into a combination of layers, managing changes to the images and applying new versions of an attribute map to the fur description was easy. Photoshop’s precision and variation in paintbrush patterns were valuable tools for this project since producing the same image maps in Maya would have been time-consuming and laborious. The brush settings used to create the image maps in Figures 3.9, 3.10, and 3.11 are shown in Figure 3.12.
Figure 3.12: Adobe Photoshop brush settings.
3.1.10 Wind Animation

The wind gusts were created with keyframed Magnitude values for the air field, as listed in Table 3-2.

<table>
<thead>
<tr>
<th>Keyframe</th>
<th>Air Field Magnitude</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>0.500</td>
<td>Initial setting</td>
</tr>
<tr>
<td>330</td>
<td>6.487</td>
<td>Increase wind to introduce breeze</td>
</tr>
<tr>
<td>345</td>
<td>1.610</td>
<td>Reduce wind for gentle breeze effect</td>
</tr>
<tr>
<td>353</td>
<td>15.792</td>
<td>Increase wind to build up anticipation</td>
</tr>
<tr>
<td>364</td>
<td>1.765</td>
<td>Reduce wind so curves drive animation</td>
</tr>
<tr>
<td>376</td>
<td>0.398</td>
<td>Rest for anticipation of wind</td>
</tr>
<tr>
<td>414</td>
<td>0.560</td>
<td>Wait for gust of wind</td>
</tr>
<tr>
<td>417</td>
<td>3.769</td>
<td>Increase wind slightly to tease audience</td>
</tr>
<tr>
<td>461</td>
<td>59.340</td>
<td>Increase wind to bend grass backward</td>
</tr>
<tr>
<td>476</td>
<td>2.382</td>
<td>Reduce wind</td>
</tr>
<tr>
<td>488</td>
<td>13.135</td>
<td>Increase wind to continue breeze</td>
</tr>
<tr>
<td>494</td>
<td>3.139</td>
<td>Reduce wind to ease out of strong gust</td>
</tr>
<tr>
<td>500</td>
<td>2.181</td>
<td>Reduce wind to ease out of strong gust</td>
</tr>
<tr>
<td>507</td>
<td>4.735</td>
<td>Increase slightly to continue breeze</td>
</tr>
<tr>
<td>511</td>
<td>10.221</td>
<td>Increase wind for last small gust</td>
</tr>
<tr>
<td>517</td>
<td>3.671</td>
<td>Continue light breeze</td>
</tr>
<tr>
<td>527</td>
<td>4.055</td>
<td>Continue light breeze</td>
</tr>
</tbody>
</table>
The air field Magnitude was the catalyst for the dynamic grass effect. Variation in the Magnitude intensity simulated gentle breezes and strong gusts of wind. The Magnitude was keyframed, which created a linear windblown grass effect in the final shot. Because the grass was set up in a different Maya scene file from McDouling’s kilt simulation, synchronizing the results of these animations was easier to do with linear results. The keyframes spanned from frame 300 to 527, which provided nine seconds of animation to work with for this shot. The final edit was only 5 or 6 seconds with the extra frames preventing re-rendering due to an inadequate number of frames to lead or close the shot.

3.2 A Field of Grass

The grass effect was used in Tartan Troubles to add thick tall grass on the edge of the cliff. The original plan for the dynamic grass effect was to cover all grassy areas of the cliff. During the latter stages of production, the team scaled back the scope of the grass. The short grassy look where Patches and McDouling meet, seen in Figure 3.13, was achieved through texturing. The technique for creating dynamic grass is scalable and feasible for providing interactive grass for larger areas. All of the previous concepts and approaches are employed with similar results on a larger scale.
Applying grass to the entire ground plane would have required image maps for Length, Tip Color, Base Color, and Baldness, similar to those used in the smaller grassy areas. Image maps for covering the ground with grass would also require larger Map Width and Height values i.e., 1024 by 1024. The Length Map would need to have more variation in color from white to black to differentiate between tall and short grass as shown in Figure 3.14.
Figure 3.14: Length and Baldness Map for full cliff fur description.

The area where black meets gray in the top half of Figure 3.14 corresponds to the edge of the cliff. All of the white area represents the tall grass set to the maximum Length value for the fur description. Variation in gray and dark gray comprise the rest of the field and result in shorter grass. The dusting of lighter gray disrupts the even tones of gray, adding noise to the length of the resulting grass and scattering slightly taller blades of grass across the field.

Painting hair follicles would have been more strategic with dense hair curves along the edge of the cliff and the area where Patches and McDouling stand (see Figure 3.15). The remaining field would have used fewer hair curves and even
incorporated fill curves in place of dynamic curves for reducing the computational load of dynamically animating a field of hair curves.

Figure 3.15: View of full cliff hair system, Fur Feedback, and Collide Sphere.

To incorporate character interaction with grass, Collide Spheres parented to McDouling’s feet and Patches’ paws would squash grass where the characters stood. Collide Spheres are hair constraints. Since the grass animation was driven by hair systems, any feature of Maya that has an effect on hair systems could be used to animate the dynamic grass. Figure 3.15 shows dense dynamic hair curves after a Collide Sphere has bounced off the ground. The red area just below the gray pill-shaped form represents the location where the Collide Sphere and dynamic curves intersected.

Color maps for covering the ground plane would incorporate more variations in color also. Figure 3.16 could be used to control the color of the grass with various shades of green, tan, and brown. This image would be used for the Base and Tip
Color attributes, allowing the random colors to show through when rendered.

Whether the situation calls for a small patch or full field, the process for producing the dynamic grass effect is the same. Automating some of the repetitive steps would shorten the process of applying the effect in the future.

![Color map for full cliff fur description.](image)

Figure 3.16: Color map for full cliff fur description.

### 3.3 Scene Set-Up with MEL

The Dynamic Fur FX MEL script was created after the production of *Tartan Troubles* to automate the common steps in implementing the dynamic grass effect. This MEL script would have saved a lot of time during the trial and error phase of learning how to control and design dynamic grass. Creating air fields, hair systems,
fur descriptions and attaching hair systems to fur descriptions are the main features of this MEL script. The GUI for this script is pictured in Figure 3.17.

![Dynamic Fur FX GUI](image)

Figure 3.17: Screenshot of GUI for MEL script.

When running the Dynamic Fur FX MEL script, air fields were created by entering values for the translate, Direction, Magnitude and distance settings followed by clicking the create air field button. Hair systems were created in a similar way. The target geometry for attaching the hair system must be selected prior to using the create hairsystem and apply buttons. Fur descriptions are created from the default fur values stored in Maya, with the hair system created most
recently attached to the fur description. The script also allows user-created presets
to be used for fur set-up instead of the default values.

This script makes use of option variables (optionVar) stored with Maya
property settings, which are specific to each user. Option variable values persist
between Maya sessions; therefore, the values are available each time Maya is
launched. The code snippets in Figure 3.18 show the MEL statements for storing and
querying.

```mel
//Store float values for air Direction XYZ in corresponding option variables
optionVar -fv "airDirectionX" $directX
    -fv "airDirectionY" $directY
    -fv "airDirectionZ" $directZ;
...

//Query values stored in option variables
$directX = `optionVar -q "airDirectionX"`
$directY = `optionVar -q "airDirectionY"`
$directZ = `optionVar -q "airDirectionZ"`
```

Figure 3.18: Storing and querying option variables in MEL.

The Dynamic Fur FX MEL script also uses other MEL scripts that are native to
Maya. To load these functions and procedures in Maya, the scripts must be sourced,
or loaded, into the current session. The MEL procedure listed in Figure 3.19 shows
the code for sourcing scripts.

```mel
global proc sourceScript (string $path_to_mel_script_including_filename) {
    string $command = "source "$path_to_mel_script_including_filename +"";
    eval $command ;
}
```

Figure 3.19: Procedure written to source scripts.

Global variables are used to make GUI attribute values accessible to MEL
functions and procedures that create and apply values to fields, hair, and fur. Figure
3.20 includes code snippets to illustrate this technique.
// Hair System Global Variables and default values
global string $uCountIntSliderGrp;  // Slider Group for uCount value
$uCount = `optionVar -q "createHairUCount"`; // default value from optionVar

global string $vCountIntSliderGrp;
$vCount = `optionVar -q "createHairVCount"`;
...

// Slider Group from above used for GUI attribute fields and set to default values
$uCountIntSliderGrp = `intSliderGrp -min 0 -max 500 -label "U Count:" -field true -value $uCount`; // set to value from optionVar

$vCountIntSliderGrp = `intSliderGrp -min 0 -max 500 -label "V Count:" -field true -value $vCount`; // set to value from optionVar

...

// button for creating hair system needs access to global variables to pass // values to createHairSystem procedure. Notice the $uCountIntSliderGrp and // $vCountIntSliderGrp variables passed as parameters
button -label "Create Hairsystem" -command "createHairSystem($uCountIntSliderGrp, $vCountIntSliderGrp, $passiveFillIntSliderGrp, $randomizationFloatSliderGrp, $dynamicRadioButtonGrp, $pointsPerHairIntSliderGrp, $lengthFloatSliderGrp)";
...

// procedure for creating hair system uses parameter values to create hair system // according to the values set in the GUI attribute fields
proc createHairSystem(
    string $uCountIntGrp,
    string $vCountIntGrp,
    string $passiveFillIntGrp,
    string $randomizationFloatGrp,
    string $dynamicRadioGrp,
    string $pointsPerHairIntGrp,
    string $lengthFloatGrp)
{
...

Figure 3.20: Using global parameters to pass values to procedure.
User-defined procedures, such as `createHairSystem` in Figure 3.20, were created to enable the script to work as shown. These procedures take values from the GUI controls as parameters and place those values in MEL commands. The entire Dynamic Fur FX MEL script is available in Appendix A.
CHAPTER 4
RESULTS

The dynamic grass effect can be adapted to various situations. The following sections will tie the implementations in chapter 3 to their respective visualizations, with an explanation of the technical considerations involved.

4.1 Grass in Tartan Troubles

The following figures show rendered frames from the final Tartan Troubles animation. The selected shots contain grass created with the implementation discussed in Section 3.1.

Figure 4.1 contains Patches, sleeping on a rock, surrounded by dynamic grass. To apply grass to this scene, Length Maps and color maps were created to control the grass around the rock. To reduce render time, grass was only included where visible by the camera. The air field's Magnitude was keyframed at lower intensities for a gentle breeze effect.
Figure 4.1: Introduction of Patches the dog.

The tufts of grass lining the edge of the cliff can be seen in Figure 4.2. On the right side of this figure, anomalies from hair systems are seen among the tall grass. Each tuft is an individual fur description and hair system. The movement in this shot is subtle compared to Figure 4.1. All of the attention is focused on the McDouling’s struggle.

Figure 4.2: McDouling struggles for freedom from Patches’ bite.
Figure 4.3 shows the grass and characters before the gust of wind. The grass is relatively straight and tall. Figure 4.4 shows wind-blown grass and McDouling's kilt flapping in the wind. For the grass, the air field Magnitude combined with the fur Tip Curl attribute worked together to give the effect of strong wind.

![Figure 4.3: Shot of grass and kilt before strong wind.](image)

![Figure 4.4: Shot of grass and kilt affected by strong wind.](image)

### 4.2 A Field of Dynamic Grass

A scalable solution is one that can be applied in multiple scenarios of size and complexity. The dynamic grass effect achieved with Maya fur and hair can also cover
an entire field of grass. Figure 4.5 shows the main acting area covered in grass with a model of McDouling, using the implementation presented in Section 3.2. A Collide Sphere was parented to the gray oblong sphere to simulate a foot stepping on the grass. The Collide Sphere was not set to render, but its impact on the grass is shown in the dark oval of grass underneath the sphere.

![Figure 4.5: Full field of grass technique applied to cliff.](image)

The tall blades along the edge of the cliff, as well as the short grass around McDouling’s feet, were created with the same fur description. Figure 4.5 was rendered using the attribute maps from Figures 3.14 and 3.16. The variation in colors seen in the image maps is evident in the tall grass and the shorter grass around McDouling’s feet. The light gray values scattered over Figure 3.14, the Length and Baldness Map, appear here as uneven areas in the grass. These areas are most noticeable in the lower left quadrant of Figure 4.5 in front of the tall grass. Having a full field of grass for the characters and wind to interact with would
enhance the role of the environment in *Tartan Troubles*, further supporting the action in each scene.
CHAPTER 5
CONCLUSION AND FUTURE WORK

Technical direction involves researching and evaluating techniques for special effects. For Tartan Troubles the effect involved providing a solution for creating windblown grass. By combining field, hair, and fur components, the resulting system provided both a flexible and scalable solution. Tufts of grass, as well as fields of grass, are achievable with this dynamic grass effect.

Creating windblown grass initially involved significant user interaction. MEL scripting reduced some of the steps through automation. Using presets reduced the time and tedium required for applying attribute settings. Fur presets, however, do not save attribute maps when the preset is created. A dialogue box offering the option to save attribute maps would be an improvement. Additionally, the dynamic fur effect MEL script could include a list of presets for air fields, which would reduce the amount of time spent setting initial values for attributes animated with keyframes. The grass effect could also benefit from noise adjustments. The windblown grass moves correctly, but would be more convincing if the blade tips randomly separated from each patch as they bend and flap in the wind.

This thesis describes the process for creating the dynamic grass effect in Tartan Troubles. With emphasis on the final shot, each step involved in setting up the grass effect was discussed as well as specific settings used for key attributes in air fields, hair systems, and fur descriptions. This effect can be applied to animal fur, facial hair, bristles for a dense hairbrush, and many other situations.
REFERENCES


APPENDIX A
DYNAMIC FUR FX MEL SCRIPT

global proc sourceScript (string $path_to_mel_script) {
    string $command = "source \""+$path_to_mel_script+"\"";
    eval $command;
}

string $script1 = "/Applications/Autodesk/maya2011/Maya.app/Contents/scripts/others/createHair.mel";
//in order to use the global procedure updateHSPlaceMenu, we need to source performCreateHair.mel

string $script2 = "/Applications/Autodesk/maya2011/Maya.app/Contents/scripts/others/performCreateHair.mel";
sourceScript ($script1);
sourceScript ($script2);

// Hair System Global Variables and default values
global proc dynamicFurSystemUI(){
    string $window_name = `window -title "Dynamic Fur FX" -menuBar true`;

    // Hair System Global Variables and default values
    global string $translateFloatFieldGrp;
    $directX = `optionVar -q "airDirectionX"`;
    $directY = `optionVar -q "airDirectionY"`;
    $directZ = `optionVar -q "airDirectionZ"`;

    global string $directionFloatFieldGrp;
    $transX = `optionVar -q "airTranslateX"`;
    $transY = `optionVar -q "airTranslateY"`;
    $transZ = `optionVar -q "airTranslateZ"`;

    global string $magnitudeFloatSliderGrp;
    $magnitude = `optionVar -q "airMagnitude"`;

    global string $mDistanceIntSliderGrp;
    $maxDistanceValue = `optionVar -q "airMaxDistance"`;

    global string $mDistanceRadioButtonGrp;
    $maxDistanceRadio = `optionVar -q "airMaxDistCBox"`;

    // Hair System Global Variables and default values
    global string $uCountIntSliderGrp;
    $uCount = `optionVar -q "createHairUCount"`;

    global string $vCountIntSliderGrp;
    $vCount = `optionVar -q "createHairVCount"`;

    global string $passiveFillIntSliderGrp;
global string $randomizationFloatSliderGrp;
$randomization = `optionVar -q "createHairRandomization"`;

global string $simulationType = `optionVar -q "createHairSimulationType"`;

global string $pointsPerHairIntSliderGrp;
$nu
mCvs = `optionVar -q "createHairPointsPerHair"`;

global string $lengthFloatSliderGrp;
$length = `optionVar -q "createHairLength"`;

// Fur Description Global Variables and default values

global string $myFurPresetOptionMenuGrp;
global string $myLastFurPreset = `optionVar -q "myLastFurPresetOptionUsed"`;

global string $usePresetRadioButtonGrp;
$myUseFurPreset = `optionVar -q "myUseFurPreset"`;

// AIR SECTION
rowColumnLayout;
frameLayout -collapsible true -label "Air Field Values";
rowColumnLayout;

// frameLayout -label "Translation";
$translateFloatFieldGrp = `floatFieldGrp -label "Translate XYZ:" -numberOfFields 3 -value1 $transX -value2 $transY -value3 $transZ`;

$directionFloatFieldGrp = `floatFieldGrp -label "Direction XYZ:" -numberOfFields 3 -value1 $directX -value2 $directY -value3 $directZ`;

$magnitudeFloatSliderGrp = `floatSliderGrp -min 0 -max 500 -label "Magnitude" -field true -value $magnitude`;

$mDistanceIntSliderGrp = `intSliderGrp -min 0`
-max 500
-label "Max Distance"
-field true
-value $maxDistanceValue`;

$mDistanceRadioBtnGrp = `radioBtnGrp -
  numberOfRadioButtons 2
-label "Use Max Distance?"
-labelArray2 "Yes" "No"
-select $maxDistanceRadio`;

rowColumnLayout -nc 2;
  button -label "Create Air Field" -command
  "buttonCreateAir("create"), $translateFieldGrp, $directionFieldGrp, $magnitudeFloatSliderGrp, $mDistanceIntSliderGrp, $mDistanceRadioBtnGrp);"

  button -label "Apply" -command "buttonCreateAir("apply"), $translateFieldGrp, $directionFieldGrp, $magnitudeFloatSliderGrp, $mDistanceIntSliderGrp, $mDistanceRadioBtnGrp);"

  setParent..;
  setParent..;
  setParent..;

// HAIR SECTION=----------------------------------
frameLayout -collapsible true -label "Create Hair Options";
rowColumnLayout;

  $uCntIntSliderGrp = `intSliderGrp -min 0
    -max 500
    -label "U Count:"
    -field true
    -value $uCount`;

  $vCntIntSliderGrp = `intSliderGrp -min 0
    -max 500
    -label "V Count:"
    -field true
    -value $vCount`;

  $passiveFillIntSliderGrp = `intSliderGrp -min 0
    -max 500
    -label "Passive Fill:"
    -field true
    -value $passiveFill`;

53
$randomizationFloatSliderGrp = `floatSliderGrp -min 0.0000
        -max 1.0000
        -fieldMaxValue 1000.0
        -label "Randomization:"
        -field true
        -value $randomization`;

dynamicRadioButtonGrp = `radioButtonGrp -numberOfRadioButtons 2
        -label "Curves:"
        -labelArray2 "Dynamic" "Static"
        -select $simulationType`;

$pointsPerHairIntSliderGrp = `intSliderGrp -min 2
        -max 500
        -label "Points per hair:"
        -field true
        -value $numCvs`;

$lengthFloatSliderGrp = `floatSliderGrp -min 0.0000
        -max 20.0000
        -label "Length:"
        -field true
        -value $length`;

optionMenuGrp
        -label
        (uiRes("m_performCreateHair.kPlaceHairsInto")) hsPlaceMenu;

menuItem -label (uiRes("m_performCreateHair.kNewHairSystem"));

rowColumnLayout -nc 2;

button -label "Create Hairsystem" -command
        "createHairSystem($uCountIntSliderGrp,
                        $vCountIntSliderGrp, $passiveFillIntSliderGrp,
                        $randomizationFloatSliderGrp, $dynamicRadioBlurButtonGrp,
                        $pointsPerHairIntSliderGrp, $lengthFloatSliderGrp)";

button -label "Apply" -command
        "createHairSystem($uCountIntSliderGrp, $vCountIntSliderGrp,
                        $passiveFillIntSliderGrp,
                        $randomizationFloatSliderGrp, $dynamicRadioBlurButtonGrp,
                        $pointsPerHairIntSliderGrp, $lengthFloatSliderGrp)";

updateHSPlaceMenu( true );
scriptJob -p hsPlaceMenu -e "SelectionChanged"
        ("updateHSPlaceMenu false");

        setParent..
        setParent..;
        setParent..
        //FUR SECTION=-----------------------------
frameLayout -collapsible true -label "Fur Description";
rowColumnLayout;

$usePresetRadioButtonGrp = radioButtonGrp -
    numberOfRadioButtons 2
    -label "Use Fur Preset?"
    -labelArray2 "Yes" "No"
    -select $myUseFurPreset;
proc createFurSystem(string $usePresetRadioButtonGrp,
string $myFurPresetOptionMenuGrp) {

// select object first before running this procedure

string $geometrySelected[] = `ls -sl`;

// if no geometry selected throw error
if(size($geometrySelected) < 1) {
    error "Must select the geometry before clicking the create button.";
}

if( `nodeType $geometrySelected[0]` != "transform") {
    // nurbs and polygon primitives are of type transform
    error "Must select nurbs or polygon geometry before clicking the create button.";
}

string $geometry = $geometrySelected[0];

HfCreateAndAssignHDF 0;

string $furPresetDirectory = "attrPresets/FurDescription/";
string $myPresetDir = `internalVar -userPresetsDir`;
string $myfurPreset = $myPresetDir + $furPresetDirectory + $myFurPresetOptionMenuGrp;

// get a list of all Fur Descriptions and reference the last Fur Description created
string $furDes[] = `ls -type FurDescription`;
string $lastFurDescriptionCreated = $furDes[size($furDes)-1];

// get a list of all Hair Systems and reference the last Hair System created
string $hairSystemShape[] = `ls -type hairSystem`;
if(size($hairSystemShape) < 1) {
    error "This script requires 1 hair system to exist before creating and connecting fur description(s).";
}

string $lastHairSystemCreated = $hairSystemShape[size($hairSystemShape)-1];

if($usePresetRadioButtonGrp == "Yes") {
    applyPresetToNode $lastFurDescriptionCreated "" "" $myfurPreset 1;
}

// connect hair system to fur description
assignHStoFD $lastFurDescriptionCreated $lastHairSystemCreated;
}

proc createHairSystem{
string $uCountIntGrp,
string $vCountIntGrp,
string $passiveFillIntGrp,
string $randomizationFloatGrp,
string $dynamicRadioGrp,
string $pointsPerHairIntGrp,
string $lengthFloatGrp)
{
    $uCount = `intSliderGrp -q -value $uCountIntGrp`;
    $vCount = `intSliderGrp -q -value $vCountIntGrp`;
    $numCvs = `intSliderGrp -q -value $pointsPerHairIntGrp`;
    $passiveFill = `intSliderGrp -q -value $passiveFillIntGrp`;
    $randomization = `floatSliderGrp -q -value $randomizationFloatGrp`;
    $simulationType = `radioButtonGrp -q -select $dynamicRadioGrp`;
    $length = `floatSliderGrp -q -value $lengthFloatGrp`;

    $restCurves = 0;      // no
    $edgeBounded = 0;     // no
    $equalize = 0;        // no
    $outputMode = 2;      // nurbs curves is the output type 1 =
                          // paintFX, 2 = nurbs curves, 3 = nurbs
                          // curves + paint FX
    $hairCreateType = 1;  // 1 = grid hairs, 2 = Hair at selected
                          // surface points/ faces

    //createHair 7 6 10 0 0 0 0 5 0 2 1 1;
    createHair $uCount $vCount $numCvs $restCurves $passiveFill
    $edgeBounded $equalize $length $randomization $outputMode
    $simulationType $hairCreateType;

    // Save values to be used a default values for the next time the
    script loads
    // This feature saves the values per user so values persist per user
    // not for all users on one computer.
    optionVar -iv "createHairUCount" $uCount
                -iv "createHairVCount" $vCount
                -iv "createHairPassiveFill" $passiveFill
                -fv "createHairRandomization" $randomization
                -iv "createHairSimulationType" $simulationType
                -iv "createHairPointsPerHair" $numCvs
                -iv "createHairLength" $length;
}
proc printTranslate(string $translateGrp){
    $transX = `floatFieldGrp -q -value1 $translateGrp`;
    $transY = `floatFieldGrp -q -value2 $translateGrp`;
    $transZ = `floatFieldGrp -q -value3 $translateGrp`;

    print("transX = " + $transX + " \n");
    print("transY = " + $transY + " \n");
    print("transZ = " + $transZ + " \n");
}

proc buttonCreateAir(string $button, string $translateGrp, string $directionGrp, string $magnitudeGrp, string $maxDistanceSliderGrp, string $maxDistanceRadioGrp){
    global int $maxDistanceForCommand;
    $transX = `floatFieldGrp -q -value1 $translateGrp`;
    $transY = `floatFieldGrp -q -value2 $translateGrp`;
    $transZ = `floatFieldGrp -q -value3 $translateGrp`;

    $directX = `floatFieldGrp -q -value1 $directionGrp`;
    $directY = `floatFieldGrp -q -value2 $directionGrp`;
    $directZ = `floatFieldGrp -q -value3 $directionGrp`;

    $maxDistanceInt = `intSliderGrp -q -value $maxDistanceSliderGrp`; // value from User Interface
    // we want to keep this value for future use
    $maxDistanceForCommand = $maxDistanceInt;

    // preserve the value and pass it on to the command
    $magnitudeFloat = `floatSliderGrp -q -value $magnitudeGrp`;

    $maxDistanceUse = `radioButtonGrp -q -select $maxDistanceRadioGrp`;

    // Use Max Distance? Yes is 1, no is 2.
    if($maxDistanceUse == 2){
        global int $maxDistanceForCommand;

        $maxDistanceForCommand = -1;
        // -1 indicates that the field has no maximum distance.
    }

    if($button == "apply"){
        string $field[] = `ls -sl`; //get selected airField
        // should be an airField, let's check

        // if no airField selected throw error
        if(size($field) < 1){
            error "Must select the airField to apply values to before clicking apply button.";
        }

        if( `nodeType $field[0]` != "airField"){
            // further actions...
        }
    }
}
error "Must select the airField to apply values to before clicking apply button."

}

string $airField = $field[0];
print("airField name = "+$airField);

// Apply values to the selected airField
// by setting the attribute values to user defined values
setAttr ($airField +".directionX") $directX;
setAttr ($airField +".directionY") $directY;
setAttr ($airField +".directionZ") $directZ;
setAttr ($airField +".translateX") $transX;
setAttr ($airField +".translateY") $transY;
setAttr ($airField +".translateZ") $transZ;
setAttr ($airField +".magnitude") $magnitudeFloat;
setAttr ($airField +".maxDistance") $maxDistanceForCommand;

}
else{
// create an air field
air -pos $transX $transY $transZ -m $magnitudeFloat -dx $directX -dy $directY -dz $directZ -mxd $maxDistanceForCommand;
}

// Save values to be used a default values for
// the next time the script loads
// This feature saves the values per user
// so values persist per user
// not for all users on one computer.

optionVar -fv "airDirectionX" $directX
-fv "airDirectionY" $directY
-fv "airDirectionZ" $directZ
-fv "airTranslateX" $transX
-fv "airTranslateY" $transY
-fv "airTranslateZ" $transZ
-fv "airMagnitude" $magnitudeFloat
-fv "airMaxDistance" $maxDistanceInt
-fv "airMaxDistCBox" $maxDistanceUse;

}