

5-2016

Designing and Animating a Character Sprite with Modern Techniques

Wynton Redmond

Clemson University, wyntonr@clemson.edu

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

Recommended Citation

Redmond, Wynton, "Designing and Animating a Character Sprite with Modern Techniques" (2016). *All Theses*. 2381.
https://tigerprints.clemson.edu/all_theses/2381

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.

DESIGNING AND ANIMATING A CHARACTER SPRITE WITH
MODERN TECHNIQUES

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
Wynton Redmond
May 2016

Accepted by:
Victor Zordan, Committee Chair
Brian Malloy
Kathleen Thum

ABSTRACT

High-quality 2D animation for video game production is both strenuous and time consuming. Traditionally, 2D game animation consisted of drawing each frame by hand and processing it into a bitmap for use in-game. As every frame was individually drawn, it was difficult to create enough drawings for smooth animation as well as keep form consistent between frames. Although, this technique usually resulted in a strong sense of 3D volume and realism when well executed. Current technology allows for faster 2D animation workflows using interpolation and bone systems as well as greater consistency, smoothness, and efficiency, but oftentimes the results lose the sense of depth and quality found in traditional animation. This thesis explores efficiently creating, and animating a 2D sprite by utilizing a composite of traditional animation techniques and computer animation practices. Using Adobe Photoshop, Adobe Flash, and the Unity3D game engine, a short game was created to demonstrate this process in a finished work.

TABLE OF CONTENTS

	Page
TITLE PAGE.....	i
ABSTRACT.....	ii
LIST OF FIGURES.....	iii
I. INTRODUCTION AND MOTIVATION.....	1
II. BACKGROUND.....	3
2.1 Tools.....	3
2.1.1 Adobe Photoshop CC.....	3
2.1.2 Adobe Flash CC.....	3
2.1.3 Unity3D.....	4
2.2 Character Design for Games.....	4
2.3 Methods of Traditional 2D Game Animation.....	4
2.3.1 Key Poses.....	4
2.3.2 Inbetweens.....	6
2.4 Methods of 2D Computer Animation.....	8
2.4.1 Puppet Animation.....	8
2.4.2 Interpolation.....	10
III. ANIMATING.....	12
3.1 Preparation.....	12
3.1.1 Character Design.....	12
3.1.2 Asset Creation.....	13
3.2 Animation Process.....	16
3.2.1 Key poses.....	16
3.2.2 Interpolated Animation.....	17
3.2.3 Combined Process Animation.....	18
3.2.3 Drawn Animation.....	19
IV. RESULTS AND CONCLUSION.....	20
BIBLIOGRAPHY.....	24

LIST OF FIGURES

Figure		Page
1.1	Screenshot of the video game, Gunstar Heroes (© Treasure, © Sega).....	2
1.2	Screenshot of the video game, Megaman II (© Capcom).....	2
2.1	Key poses of Makoto and Q from <i>Street Fighter III:Third Strike</i> (©Capcom).....	5
2.2	Final Makoto Animations from <i>Street Fighter III:Third Strike</i> (©Capcom).....	7
2.3	Various character body parts illustrated in Photoshop.....	8
2.4	Image of how characters are put together in Flash.....	9
2.5	Different art assets over identical animation.....	10
2.6	Example of ineffective 2D image transform.....	10
2.7	Keyframes for walk cycle and interpolated frames in Flash (legs only).....	11
3.1	Character design sketch painted in Photoshop.....	12
3.2	Parts of character created in Flash.....	13
3.3	Placement of character parts.....	14
3.4	Key poses for main actions.....	16
3.5	Path of leg and lower body animation.....	17
3.6	Arm motion path, arm drawings at various angles.....	18
3.7	Poncho animation frames.....	19
4.1	Main character animation frames.....	21

List of Figures (Continued)

	Page
4.2 Enemy example frames.....	22
4.3 Screenshot of resulting game, <i>Timeraizer</i>	23
4.4 Screenshot of resulting game, <i>Timeraizer</i>	23

Chapter 1

Introduction and Motivation

The goal of this thesis is to create 2D video game sprites more efficiently. A sprite is a bitmap image commonly used for animation in older 2D video games. This paper explores using modern techniques to create animated sprites. The game being created to implement these animations is an action platformer. This genre of game typically consist of horizontal movement, jumping, and shooting. The game created is primarily inspired by the simplicity of Megaman (Figure 1.1) and the pacing of Gunstar Heroes(1.2). The simplicity of Megaman allows for a focus on clean clear, animations, while having fast movement grants the opportunity to make more dynamic animations. The game created for this thesis; “Timeraizer” follows the formula of the games mentioned in almost all areas but one. When the player avatar is hurt, the clock decreases by 1. When the player defeats an enemy, the clock increases. If the player character is hurt while the clock is at 00:00, the player loses and is forced to restart the level. This style of time constrained health was implemented with the intent of increasing the pace even more.

This paper first explains the traditional game animation process, explains the modern approach, and then walks through using both to create an animated sprite for implementation into the Unity3D game engine.



Figure 1.1: Screenshot of the video game, Gunstar Heroes (© Treasure, © Sega)

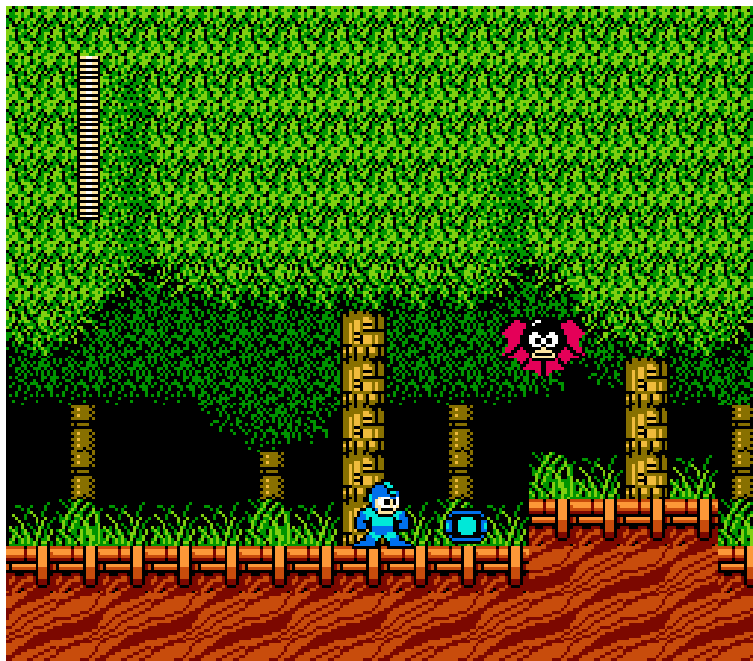


Figure 1.2: Screenshot of the video game, Megaman II (© Capcom)

Chapter 2

Background

To create 2D assets for games, a variety of software and methodologies should be understood at a working level. This section goes over the applications used, as well as thought processes and techniques for the art of character animation.

2.1 Tools

For the creation of all art assets in this process, three commercial softwares were used: raster graphics editor Adobe Photoshop, vector animation program Adobe Flash, and the Unity3D game engine.

2.1.1 Adobe Photoshop CC

Commonly known as a bitmap image editing software, Photoshop has also become the industry standard for drawing and painting digitally. Photoshop is exclusively used to create concept illustrations in this project.

2.1.2 Adobe Flash CC

Adobe Flash is a software commonly used to create vector art, animations, and web applications. It is also known as an affordable platform for creating 2D animations for television and user interfaces. In this thesis Adobe Flash is used for creating clean vector illustrations and animation.

2.1.3 Unity3D

Unity is a popular game engine with the ability to build games for a wide variety of hardware. Though primarily for 3D game development, as of Unity 4.3, many tools have been added catering to 2D game development as well. [4] Though this thesis is regarding to creating the animations for a game in the Unity engine, this thesis does not cover Unity game development.

2.2 Character Design for Games

When designing a character for production, there are some facts about the character and the game the character will be appearing in that must be taken into consideration. Primarily, a character must be designed to fit well in its environment. For example, a character that moves quickly or is commonly viewed from a distance would need simpler details and shapes to increase readability from afar. As a design for animation, strong consideration of joints, articulation, and secondary motions are required. The camera perspective is also important, as the primary viewing angle should be considered most when designing. [5]

2.3 Methods of Traditional 2D Game Animation

2.3.1 Key Poses

After generating a design for the asset, artists and game designers work together to create illustrations for movements and actions. In traditional 2D animation, these pictures would be considered the key poses of the animation. They are the most important positions in the animation and are meant to convey information as clearly as possible. [6] Clear communication is doubly important in game animation, as the poses are visual feedback from the player's actions. The illustrations in Figure 2.1 are key drawings from the production of Street Fighter III. These early sketches display different abilities the character has available to them in the game.

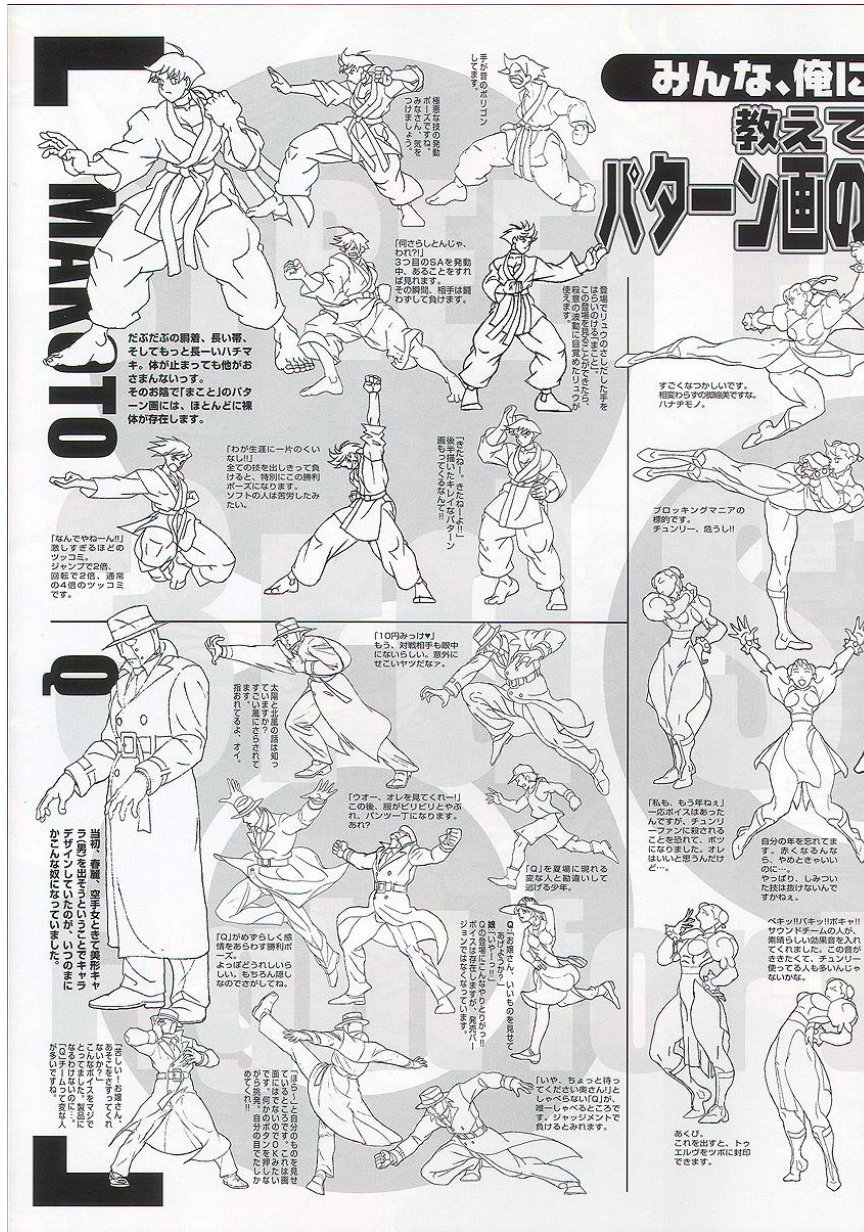


Figure 2.1: Key poses of Makoto and Q from *Street Fighter III: Third Strike* (© Capcom)

2.3.2 Inbetweens

After the keyframes are created, “inbetweens” are added. Inbetweens are additional pictures added to create smooth motion between the key poses, and they are also drawn by hand. When these frames are created, secondary motions such as fabric or hair also tend to be animated. [6] After the drawings are completed, they are reproduced as bitmap images called sprites in order to be implemented into the game. Many early games utilized graph paper for initial drawings to make them easier to recreate on a limited resolution screen. The images in Figure 2.2 are final animations from SFIII, including the inbetween images. This animation is from one of the poses in Figure 2.1. The sprites on the left are the completed frames transitioning into the key pose, and the sprites on the right are the frames transitioning back into a default standing pose. Characters animated in this way can take months to complete, since each inbetween had to be created accurately by hand, and enough inbetweens had to be drawn to ensure smooth animation.



Figure 2.2: Final Makoto Animations from *Street Fighter III:Third Strike* (©Capcom)

2.4 Methods of 2D Computer Animation

2.4.1 Puppet Animation

“If drawn ‘classical’ animation is an extension of drawing, then computer animation can be seen as an extension of puppetry- high tech marionettes.” [6]

Similar to posing an action figure, in puppet animation, pieces of a computer animated character are moved individually to create key poses. Because the animation is independent of the art asset, it is relatively easy to change the character art while keeping the same animations. In 3D animation software such as Maya, a character is ‘rigged’ with a bone structure, and these bones can be moved into different positions. A similar technique can be utilized in 2D computer animation as well, albeit some slight changes.



Figure 2.3: Various character body parts illustrated in Photoshop

To build a character for this method of 2D animation, pieces are illustrated separately and combined in an animation program. The result is a puppet whose limbs can be moved individually to create poses. As the positioning takes place on a 2D plane, movement of parts is limited to translation, rotation around the z axis, and scaling. If the 2D assets are skewed or rotated on another axis, the key pose loses its illusion of volume. (Figure 2.6) Though this method is efficient and allows for a high level of detail in individual drawings, it also results in very flat posing. However this method requires significantly less drawing, and the art remains consistent in each keyframe.

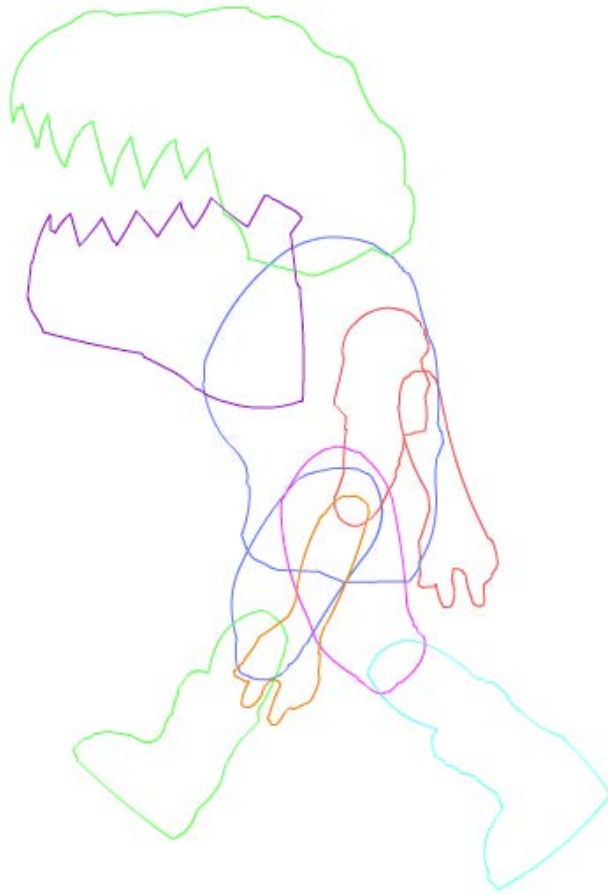


Figure 2.4: Image of how character is put together in Flash



Figure 2.5: Different art assets over identical animation



Figure 2.6: Example of ineffective 2D image transform

2.4.2 Interpolation

Though both traditional and computer animation have key poses, computer animation’s version of the inbetween drawing is vastly different. Whereas in traditional animation, each frame would have to be drawn, in computer animation, the inbetweens are calculated and filled in automatically. [7] This is called interpolation. There is no limit on the number of frames that can be interpolated, each keyframe could have as many inbetweens as it needs to ensure the smoothest animation possible. Though this can save time, it often leads to misinterpreted motions, and in 2D animation, this ability is further limited by the transforms allowed.

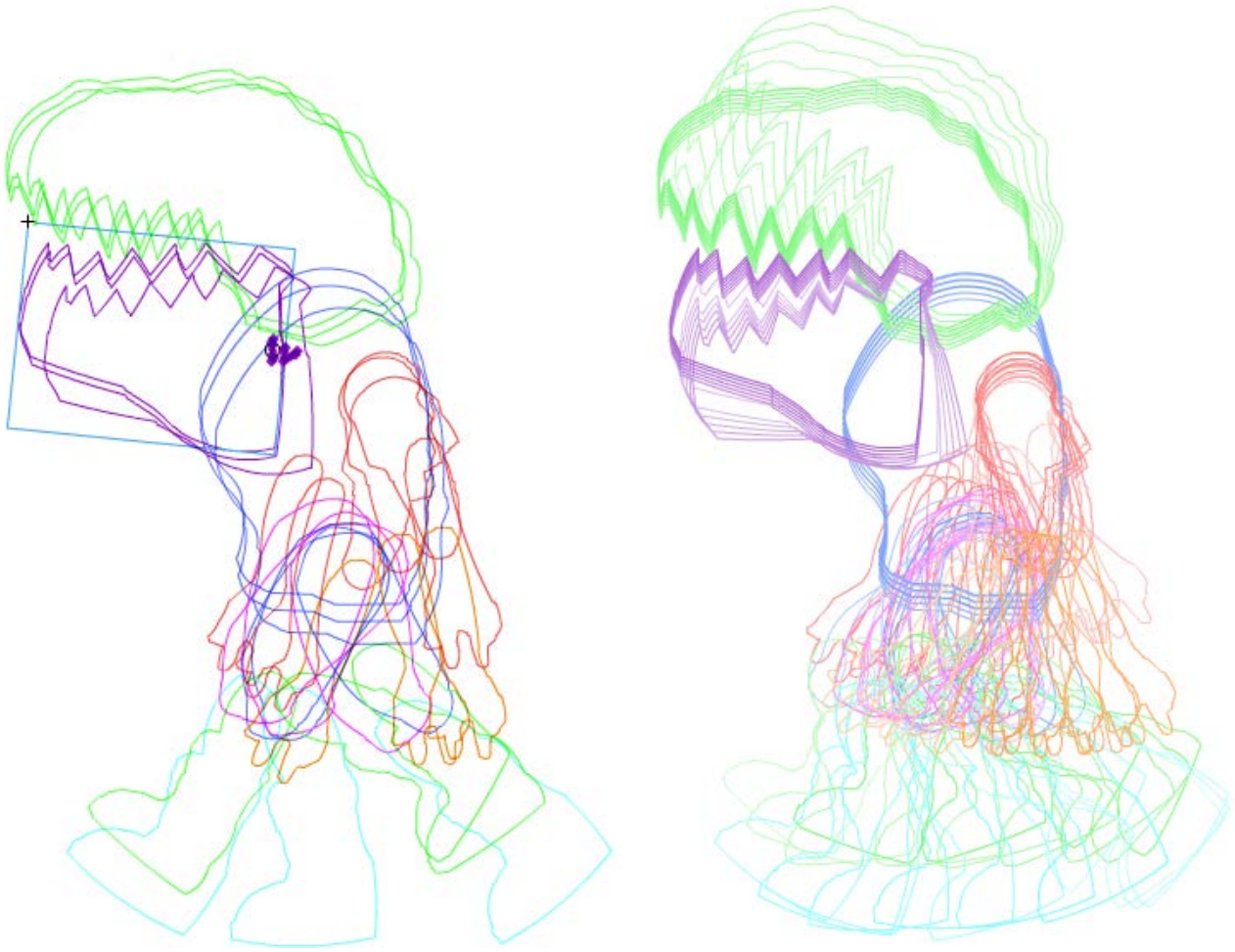


Figure 2.7: Keyframes for walk cycle and interpolated frames in Flash (legs only)

Chapter 3

Animating

3.1 Preparation

For the main character and enemies, rough drawings were created in Photoshop, and the in-game assets based off of these sketches were recreated in Adobe Flash.

3.1.1 Character Design

For this project, the game is a 2D side scroller. Traditionally, this means the game will be from a fixed perspective with the majority of action being in a profile view. As per the titles it is inspired by, the actions are relatively simple, and don't consist of much more than running, jumping, and shooting. A relatively simple design for the main character was painted in Photoshop. As she is intended for a fast paced game, most of the anatomy consist of simple, easily readable shapes, with a bright red poncho added to emphasize movement.



Figure 3.1: Character design sketch painted in Photoshop

3.1.2 Asset Creation

To create the in-game assets from the original design, the vector illustration tools in Flash were used to recreate the character as separate building blocks. As a modern 2D game, using a vector based program allows the final assets to be exported at any resolution. The created cutouts are intended to be combined and used as a puppet for the key poses.



Figure 3.2: Parts of character created in Flash

Considering that the character's primary actions are intended to be running and jumping, the legs are the most disjointed part of the character, whereas the arms consist of a single unit. The pieces were separated by the size and the amount of movement the part required. As the game is intended to be fast, very small details such as individual fingers and ankle movement were omitted.



Figure 3.3: Placement of character parts

After the assets are created, the animation process is started. First, keyframes are created by posing parts of the figure and drawing over them. Next, inbetweens are created by utilizing a combination of interpolated transforms and hand drawing. These animations are created at 20 frames per second, allowing smooth animation from interpolated transforms while also eliminating some drawing time from the 2D animation standard of 24 fps.

3.2 Animation Process

3.2.1 Key poses

Once the character has been split into its separate parts, the parts are then posed in Flash to create the key poses necessary for the game. For the main character in this project, the only required poses are standing, jumping, running, and being hit. As moving the parts using only basic transforms is limited, some parts were also drawn by hand to provide more variety and interest to the key frames. The facial expression and hair were painted over in the jumping, and getting hit keyframes. The combining of posing keyframes via cutout and drawing over them let the shapes remain almost exactly the same for each key pose, while still allowing some natural movement via draw overs.



Figure 3.4: Key poses for main actions

3.2.2 Interpolated Animation

The first parts that are animated are the largest. [6] Working from general to specific makes it easier to see progress and the direction the work is headed. In this project, the leg movements are animated first. As they are from a side view, they do not require a change in volume or size to look believable. These are simply animated by letting flash calculate the inbetweens with linear interpolation of rotation and position transforms. Other simple movements on the face and hair were created exclusively using transforms.

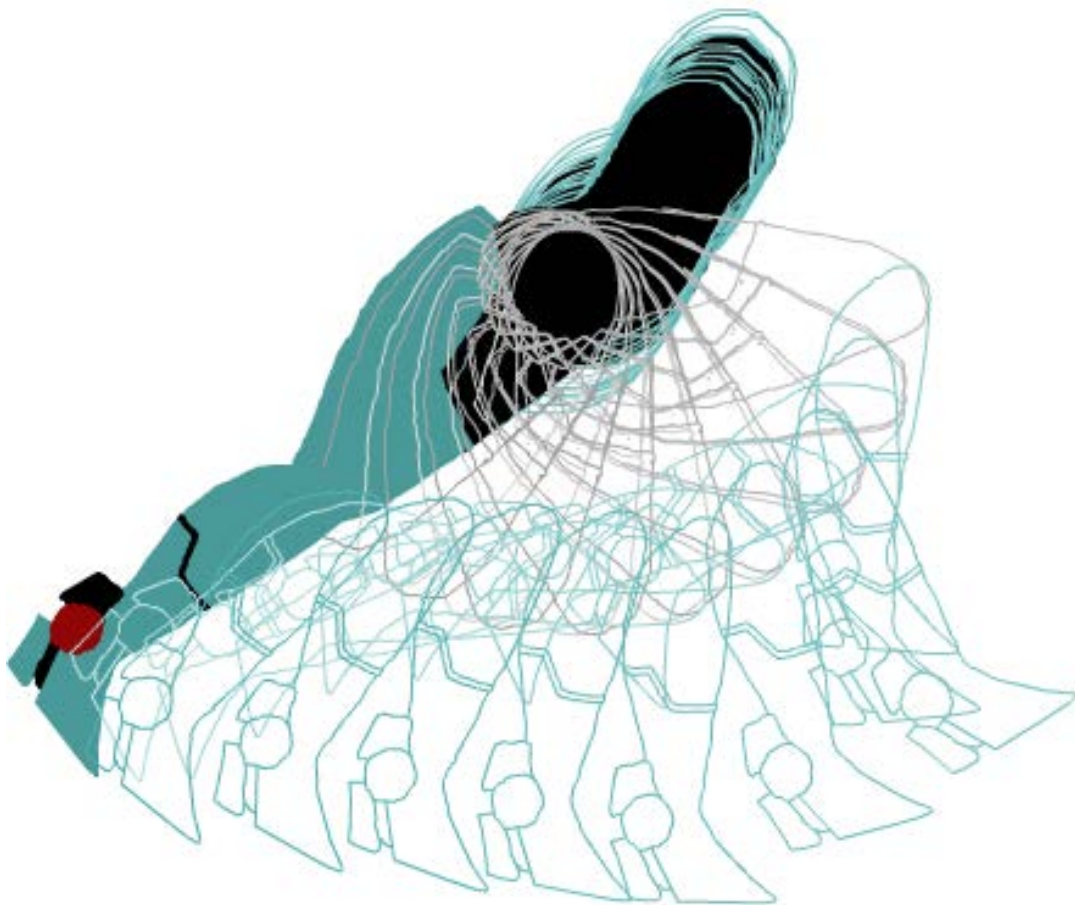


Figure 3.5: Path of leg and lower body animation

3.2.2 Combined Process Animation

For animation parts that extend towards or away from the camera, using both techniques is required. Dependent on the speed and angle of the animation, the drawing should be changed while it is transformed. Initially, the animation was a basic swing. To imply rotation towards and away from the camera, two other drawings of the arm were added inside the interpolated transform. This created an animation that utilized three hand drawn frames to imply 3D rotation, while the primary movement was a simple 2D transform.

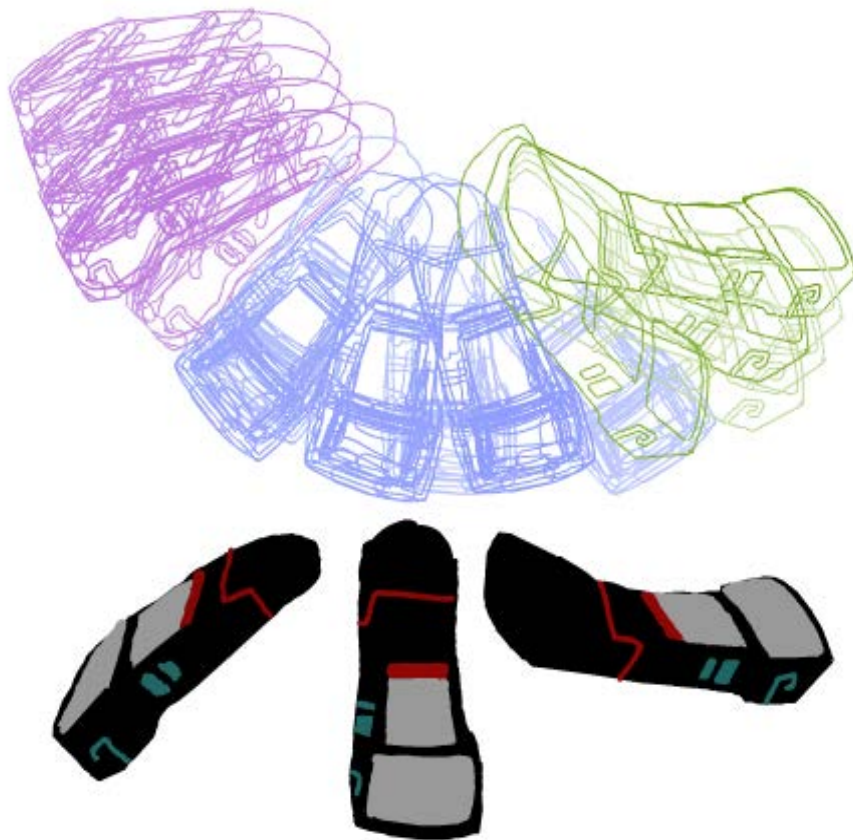


Figure 3.6: Arm motion path, arm drawings at various angles

3.2.3 Drawn Animation

The secondary animation on the fabric was done completely with traditional animation techniques. Because the fabric was resting on the arm, which is already moving in a pseudo 3D space, the fabric had to be animated to cover the volume of the shoulder. Without using a physics based simulation, 2D interpolation of the cloth would be nearly impossible. Because of this, it had to be hand animated and inbetweened completely.

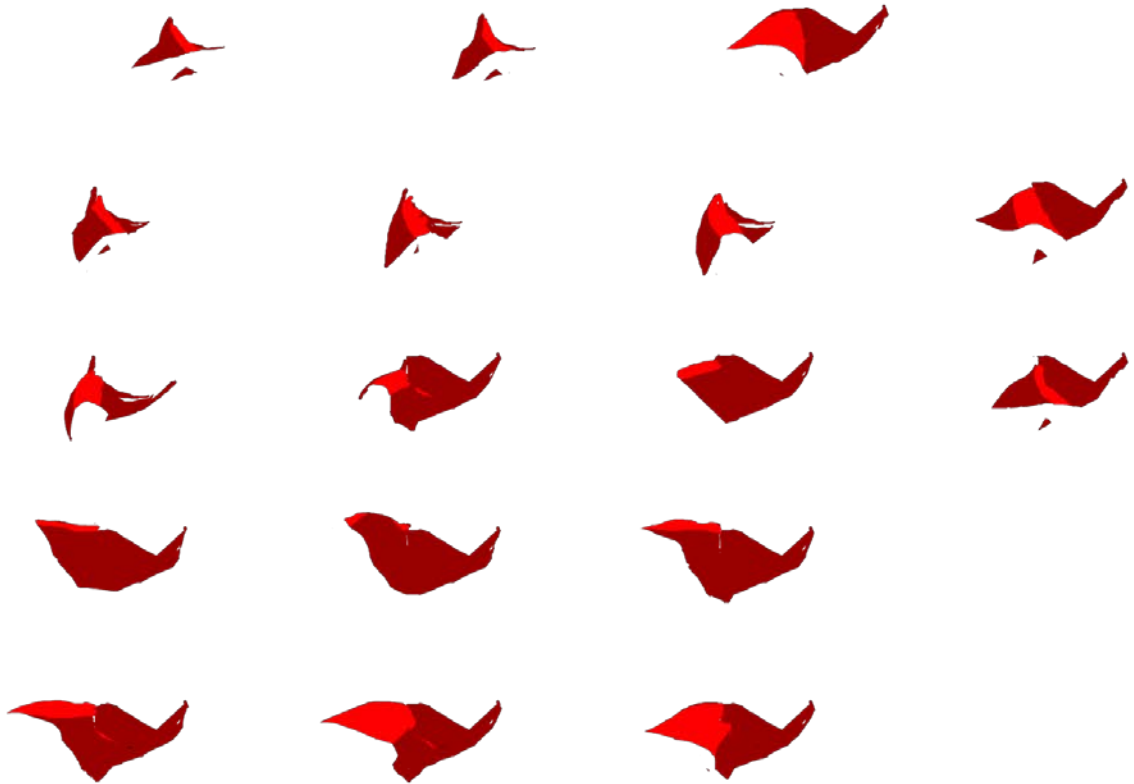


Figure 3.7: Poncho animation frames

Chapter 4

Results and Conclusion

Working with multiple animation techniques allowed the creation of many more frames at a much greater efficiency than traditional animation, while hand animating certain areas let the character retain some traditional animation nuances. Generally, movements that stayed relatively flat, were simple enough complete with keyframe interpolation alone. Over the course of development secondary motions seemed to benefit the most from traditional animation. Objects that rotated or transformed on an axis not perpendicular to the surface were also drawn by hand to give them a greater sense of natural motion and depth. All of these frames were then exported into a sprite sheet to be used in game. The complete set of main character animations is displayed in Figure 4.1, consisting of 40 individual frames, representing five actions. Simple animations of other characters were also created. They are displayed in Figure 4.2. All of the animations were used to create the game “Timeraizer”. Figures 4.3 and 4.4 are screens from the game.

This thesis implies utilizing modern animation shortcuts and traditional animation can be a very effective workflow for game animation, allowing artists to work much more quickly without a sacrifice in quality.

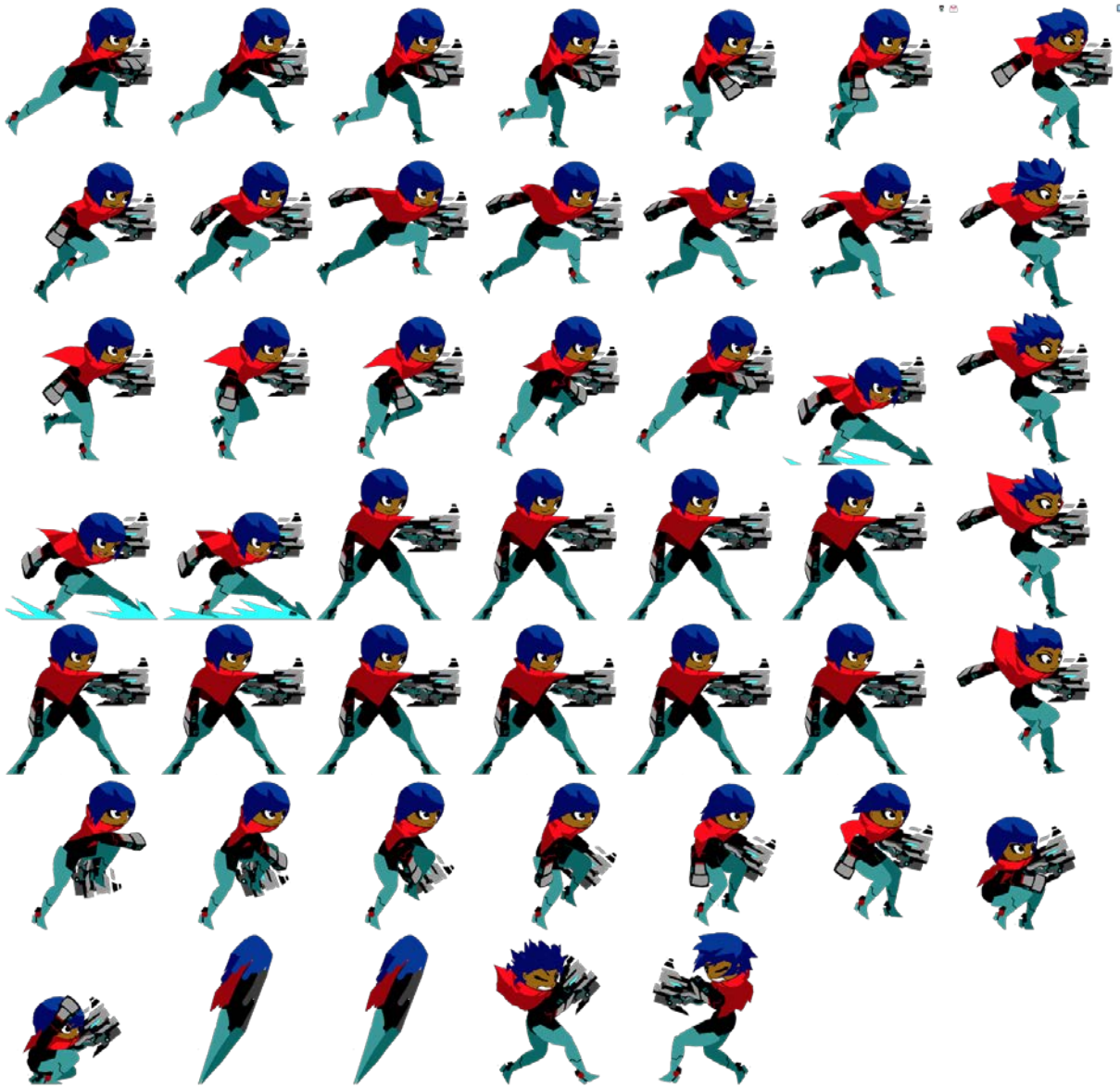


Figure 4.1: Main character animation frames

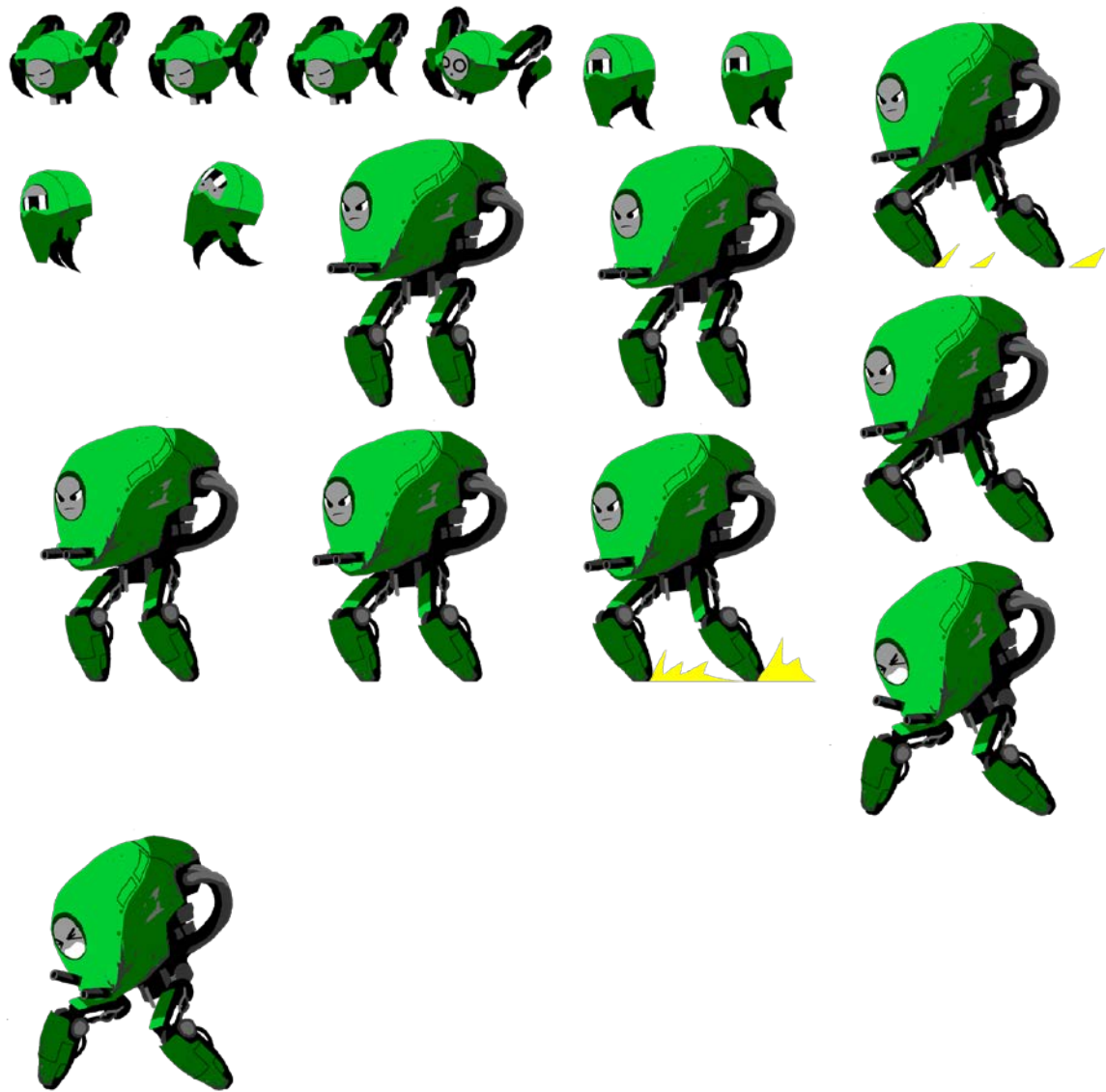


Figure 4.2: Enemy example frames

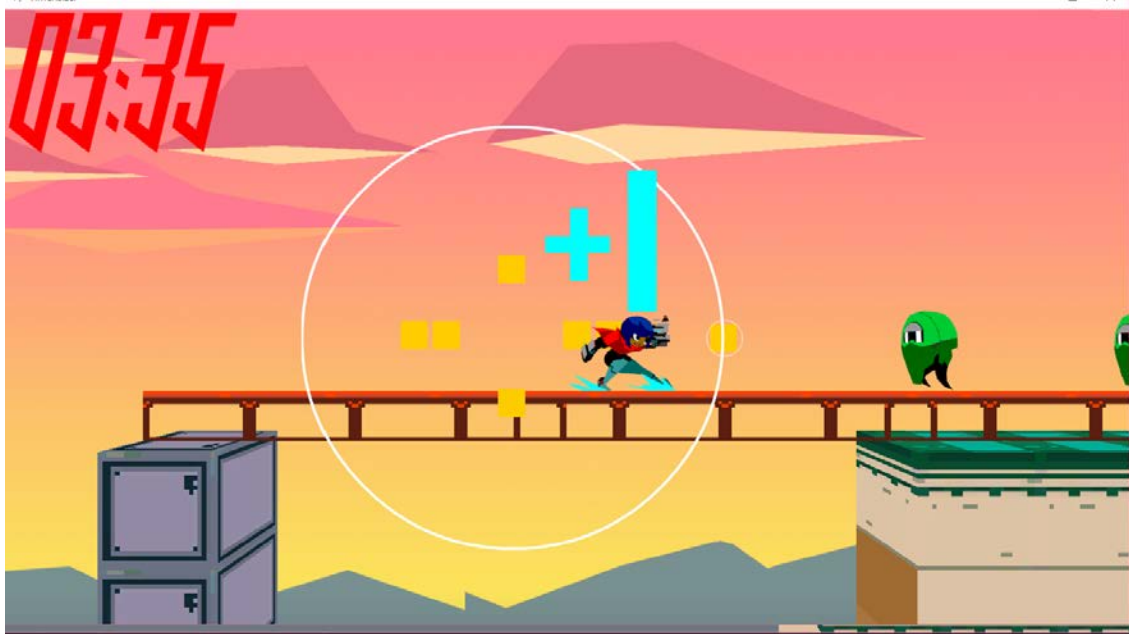


Figure 4.3: Screenshot of resulting game, *Timeraizer*



Figure 4.4: Screenshot of resulting game, *Timeraizer*

Bibliography

[1] Gunstar Heroes. Treasure, Sega. 1993.

[2] Megaman. Capcom. 1993.

[3] Street Fighter III. 3rd. Capcom. 1997.

[4] Unity3D 4.3 release notes. <https://unity3d.com/unity/whats-new/unity-4.3>.

[5] Seegmiller, Don. Digital Character Design and Painting : The Photoshop CS Edition. Charles River Media, Hingham, MA. 2004.

[6] Williams, Richard. The Animator's Survival Kit. Faber and Faber Inc. Farrar, Straus and Giroux LLC, New York. 2001.

[7] Lasseter, John. Tricks to Animating Characters with a Computer. 1994.