

Unit XXXI Assignment II

By Nathan Windisch

Task I

PIV: Software Tools for Animation

There are four main tools that are used in making animation, and I will list all of them and explain what they do.

Frame

A frame is a still image within a video or animation. Every single video ever made has frames, and the standardized amount of frames per second, or `FPS`, in a feature movie is 24. This means that for every second in a movie, 24 still images are being played. Because this is moving so fast, the video looks like things are moving. The speed at which the frames are set at in the video is called the `framerate`. If you are filming a short animation and want the film to be longer, then you could lower the framerate to say, 5 FPS. This will mean that only 5 frames are shown per second, meaning that it will be 6 times faster than an animation shot at 30 FPS. The only issue with this is that 5 frames per second will be much more slow and *clunky*, meaning that it will look less realistic. A good balance is the amount of frames that makes the movie look fluid, but still keeping the movie at a reasonable length.

Layer

When creating an animation via a computer, you will not want to make many different objects all on the same group. For one, this will make it very unorganized and hard to work with, and for another it may mean that you delete the wrong thing by accident. This is where layers come in. Layers allow the user to separate different objects and other aspects of their project, meaning that it is easier to find different objects and edit singular parts of the animation. These layers can be locked so that they cannot be moved, hidden so that they are not distracting or placed in "solo mode" so that they be worked on individually.

Button

Buttons can be found in websites and applications that require a user's input. An example of this is a music player application that will have buttons that let the user play, pause, stop, rewind and fast-forward the currently playing song. These buttons are coincidentally also found within animation applications such as Flash and After Effects. These let you navigate your animation with ease, letting you go to very precise areas of your program.

Library

Libraries are like APIs within programming, they allow you to add new features and functionality to your animations. An example of this could be if you wish to add bouncing to your animation, you can just use `bounce.js`, a Javascript library that makes CSS have a more "warm, bouncy" feel to your shapes.

PV: Factors when Creating Animations for the Web

There are a few different criteria that need to be considered when designing and creating animations for the web, and they are as follows:

- House Style
- Output Devices (PC, Mobile etc)
- Email Attachment
- Size

I shall now go more in-depth with each of these points.

House Style

The first point is house style, which is the generic style that the place that you place your animation on will have. An example of this is that if you have a website with a very dark and minimalist theme, making an animation that has a bright and sunny theme and message may not be the best idea. Another example would be if you made an animation with lots of bevels and edges on a website that was very flat and used a Material Design philosophy.

Output Devices

Output Devices are also very important. If you are creating an animation for a PC version of a website, bandwidth may not be an issue, as will be discussed later. A PC also means that the screen will be bigger and the pixel density may be lower due to the fact that the screen is larger but the resolution stays the same. A mobile will be smaller and more compact, this meaning that the animation size will need to be smaller.

Email Attachment & Size

When sending an animation via an email attachment, then the file size of the animation will need to be smaller. Compression may be required for this, but it may also lower the quality of the animation. Another reason why the file size may need to be smaller is if it is being accessed on a mobile, where data plans may limit the size of the image that the user wants to download.

MIII: Techniques used to Minimize Animation File Size

There are a few different techniques that can be used to minimize the file size of animations. The following is a short list of a few techniques that can be used:

- Quality vs Quantity
- Lossy Compression
- Lossless Compression
- Auto Cropping
- Frame Disposal

I shall now explain what they are, as seen below:

Quality vs Quantity

One of the main ways to minimise a GIF's size is to reduce the quality of the image. This can be done by just lowering the resolution of the image, but other things can be done as I shall explain below. The important thing to remember is that you will need to balance the quality of the image with the quantity of the amount of images that you can save on your drive. This basically means that you could store one GIF that has an incredibly high resolution and framerate, but if it is 1TB in size then it will be very hard to save lots of them. The other side of the coin is that you could save an image as only eight pixels in width and height, but it would be very hard to make out what it is supposed to be, especially if the original source material is an image or series of images taken from real life. A way to solve this would be to use compression, a method to make files smaller. There are two categories of compression, lossy and lossless, and I shall elaborate on them now.

Lossy Compression

Lossy Compression is when the amount of pixels within the image are reduced. This can mean just setting a group of 4 pixels that are all roughly the same colour to be just one static colour, to reducing the size of the entire image by a distinct scale factor, such as reducing a 1920x1080 pixel image to a 1080x720 pixel image, or a 4000x4000 pixel image to 2000x2000 pixel image. While this can greatly reduce file size, it can also result in a more "blurry" or pixelated image due to the resolution being reduced. Another issue is that if an image is lossy compressed too much, "artifacts" may appear which can damage the image. To put lossy compression in layman's terms, reconstructs the data via approximation.

Lossless Compression

Lossless compression is a more "reliable" method of compressing images, due to the fact that only excess pixels are removed. This is similar to auto cropping, which I shall elaborate upon later. Lossless compression allows for perfect reconstruction of data, meaning that no parts of the image are lost, which can lead to a larger file size. An example of this is the compression of the string "AAAAAABBBBBBCCCC", which can be compressed into "6A5B3C" due to the fact that there are 6As in the string, followed by 5Bs, followed by 3Cs. This means that the string can be expanded with ease, but the compressed version will take up less space.

Auto Cropping

Auto Cropping is when an animation or image has excess parts of an image removed. For instance, if an exported video of an animation is two minutes long, but the actual animation in question is only one minute long then the auto cropping tool will remove the excess frames by default. Another example is if the project file for an animation states that the image is 1920x1080 pixels in size, but the actual animation only takes up the top 500x500 pixels, then the animation can be cropped to remove these excess pixels. Auto cropping is useful as it reduced file size without reducing the actual quality of the image.

Frame Disposal

Frame Disposal is a method of reducing file size by removing the current frame before the next frame should be displayed. This effectively reduces file size as it means that only one frame can be seen at any one time. Also, it means that there are less pixels in the animation and frames are less likely to overlap, meaning that there is less wasted space and more files can be stored on the drive as the file size is lower.

Colour Density

Colour Density is the total amount of colours that can be used in a colour palette. An example of this is the GIF file format, which is a 16bit colour scheme as there are a maximum of 256 colours that can be used. A higher colour density means that there are more colours that can be used in an image, but it also increases the file size as the index of all of the colours that can be used needs to be larger. Files can deliberately reduce their colour density if some of the colours are not being used at all, thus reducing the size of the image.

DI: Compare Specialist Animation Software

In the following segment I shall make a list of 3 different specialist softwares that are used for animation and list a few things about each of them such as ease of access, price, what I liked about it, what I would improve if I could and what did not work at all. This will be in a table based format.

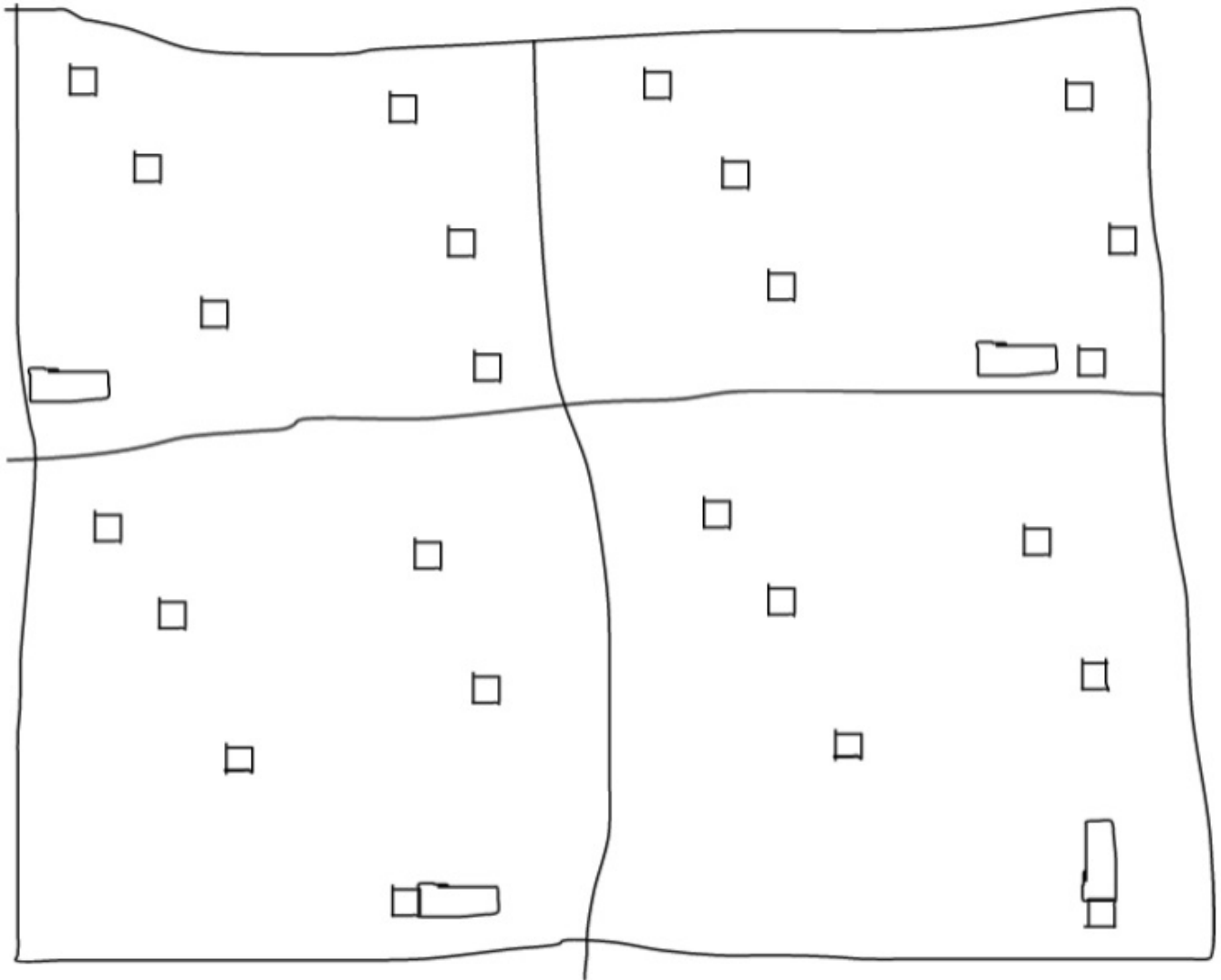
| Software Name | Price | Ease of Access | What I Liked | What I Would Improve | What Didn't Work |
|---------------------|--------|--|---|---|---|
| Flash | £20/mo | Very easy for beginners. | Expansive, helpful community. | Add more file formats that can be worked with. | Strong and stable with regular updates, I've found no bugs yet. |
| Blender | Free | Steep learning curve. | Relatively easier to pick up in comparison to others. | UI Design is cramped. | Can be prone to crashes. |
| Adobe After Effects | £20/mo | Relatively straight forward for beginners, with lots of tutorials. | Make a cheaper version for students. | Some types of motion graphics are difficult to use. | Strong and stable with regular updates, I've found no bugs yet. |

Task II

PVI: Designing Computer Animations Using Different Techniques

Animation I

The following image is my story board on the stop motion animation that I am planning on making using a digital camera and some editing software to stitch the images together.



The following is a brief list of the timing of the animation:

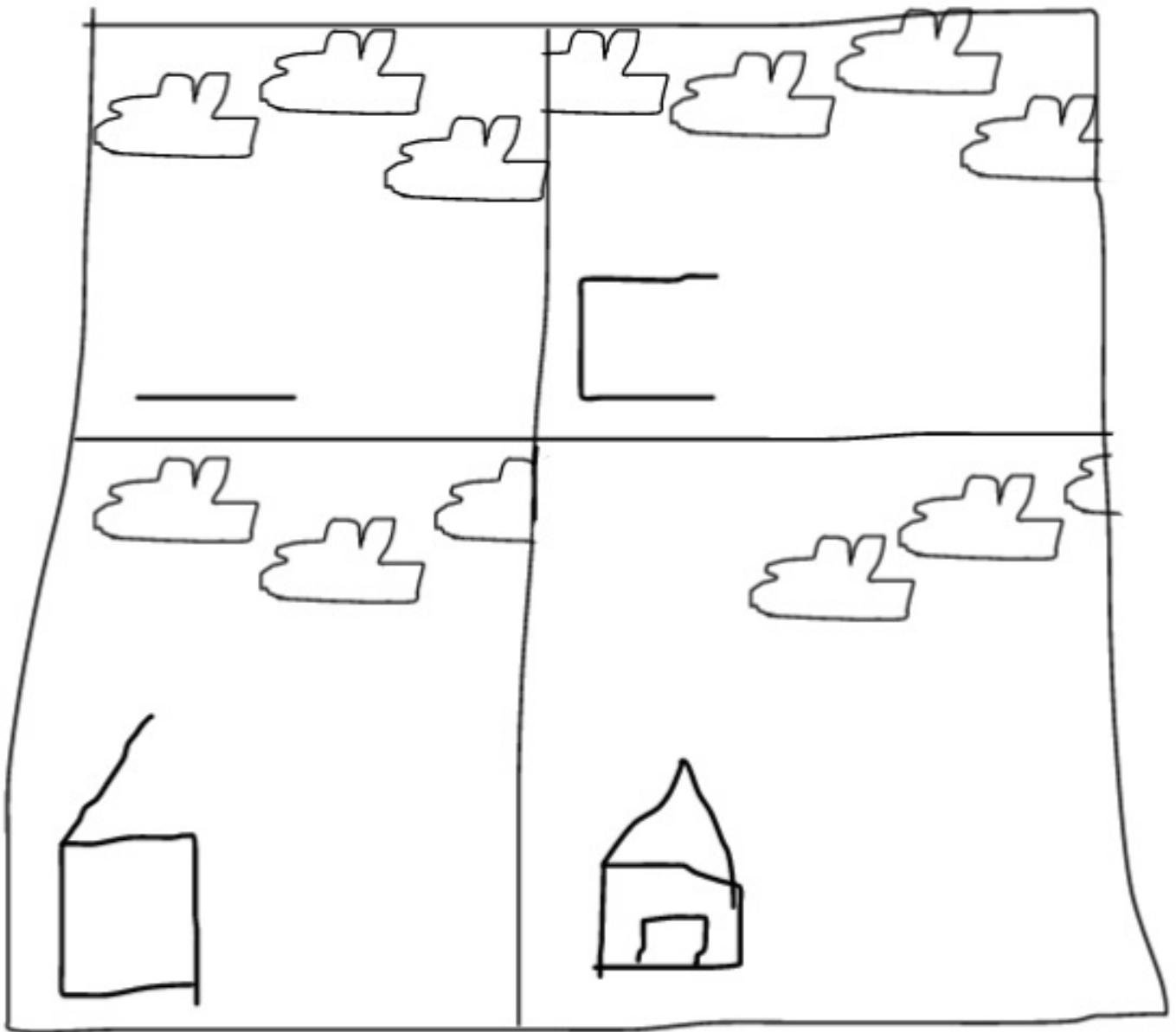
- The clouds all start at the left side of the screen. (0:01)
- The clouds individually move across the screen, taking 12 seconds each. (0:01 - 0:012)
- When the cloud hits the right side of the screen, it is "teleported" to the left side of the screen again. (0:13)
- The clouds teleport individually of one another, and keep moving for the duration of animation, which is infinite as it is similar to a GIF. (INF)
- The grounds starts off as just green grass (0:00)
- Every half a second, a part of the house is drawn. (0:00 - 0:045)
- The bottom of the house is generated (0:00 - 0:005)
- The left side of the house is generated (0:005 - 0:01)
- The top of the house is generated (0:01 - 0:015)
- The right side of the house is generated (0:015 - 0:02)

- The left side of the roof is generated (0:02 - 0:025)
- The right side of the roof is generated (0:025 - 0:03)
- After the house has been generated, the clouds repeat. (0:03 - 1:00)

While there is a door in the final product, I may not add it in as I want the animation to be minimalistic.

Animation II

The following image is my story board on the stop motion animation that I am planning on making using a digital camera and some editing software to stitch the images together.



The following is a brief list of the timing of the animation:

- Small Lego "Snake" is at the bottom of the board. Around it there are lots of 2x2 bricks. (0:00)
- The Snake starts moving towards the nearest brick. (0:01)
- When it touches the brick, the touched brick joins onto the end of the Snake. (0:02)
- The Snake continues to move around the board, collecting each brick as it goes. (0:02 - 0:37)
- The Snake moves to the left side of the board, straightened tuning out (0:36 - 0:41)
- The Snake starts to move down the left side of the board, now taking the shape of a right angle (0:41 - 0:45)
- The Snake moves off the screen (0:45 - 0:51)
- The animation ends

PVII: Implementation of Computer Animations Using Different Techniques

Animation I

Test Cases

Due to the fact that the animation uses P5js, it will work with all modern browsers including Chrome, Firefox, Safari and Opera, along with the mobile versions of all of the latter.

Feedback

When getting feedback, I spoke to one of my colleagues, Ryan Krage. He gave me feedback with my animation, and told me to add a door to the house to make it more obvious as to what it is. I did so, as can be seen in the final design. The changes that I made can be seen in the change log below.

Change Log

- The left side of the door is generated (0:03 - 0:035)
- The top of the door is generated (0:035 - 0:04)
- The right side of the door is generated (0:04 - 0:045)

Animation II

Test Cases

Due to the fact that the animation is encoded in the MP4 format, it will work with all modern browsers including Chrome, Firefox, Safari and Opera, along with the mobile versions of all of the latter. It may not work on Linux machines due to the fact that MP4 is not opensource.

Feedback

When looking for feedback, I spoke to my colleague Daniel Easteal. After viewing the animation, he was unsure as to what it was. I explained that it was a simulation of the popular game "Snake" represented by Lego in a stop motion animation. He understood after I explained, but I felt that I needed to verify what the animation actually was before it started, which lead me to the change that I made in my change log below.

Change Log

My only change was to add a title to the animation to explain what it was.

DII: Tools and Techniques Used to Create Animations

In the following segment I will go through the two animation techniques that I used to create each animation and I will list what I liked, what I would improve and what did not work. Like the previous criteria, it will be in a table format.

| Technique Name | What I Liked | What I Would Improve | What Did Not Work |
|----------------|---|--|--|
| Frame By Frame | How it was each to go back and change things. | I would make the User Interface easier to use. | My laptop is not very powerful, and so the program ran slowly. |
| Stop Motion | That it was much more hands-on | I would increase the length of the animation. | I kept on having to readjust the camera, so I should invest in a camera stand next time. |