



Hi-Bit Pixel Graphics – The New Era of Pixel Art

Olli Heikkinen

Thesis
May 2021

Degree Programme in Information Business Systems
Game Production

ABSTRACT

Tampere University of Applied Sciences
Information Business Systems
Game Production

Olli Heikkinen
Hi-Bit Pixel Graphics – New Era of Pixel Art

Bachelor's thesis
35 pages
May 2021

This bachelor's thesis studies how pixel graphics in video games are seen today, and what current trends make classic pixel graphics hi-bit. This thesis briefly covers the beginnings of pixel graphics, how pixel graphics in video games have changed over the years, as well as a few key techniques that make hi-bit pixel art.

To further demonstrate the elements of hi-bit pixel graphics, a short game demo "Mr. Skullerton's Vault" was created in the Unity game engine. In this demonstration a variety of different hi-bit pixel art techniques were tested, including pixel perfect settings, normal mapping, skeletal animation. The techniques tested in this demonstration proved to be significant elements, which distinguish classic pixel graphics from hi-bit pixel art

Key words: pixel graphics, pixel art, video game graphics

CONTENTS

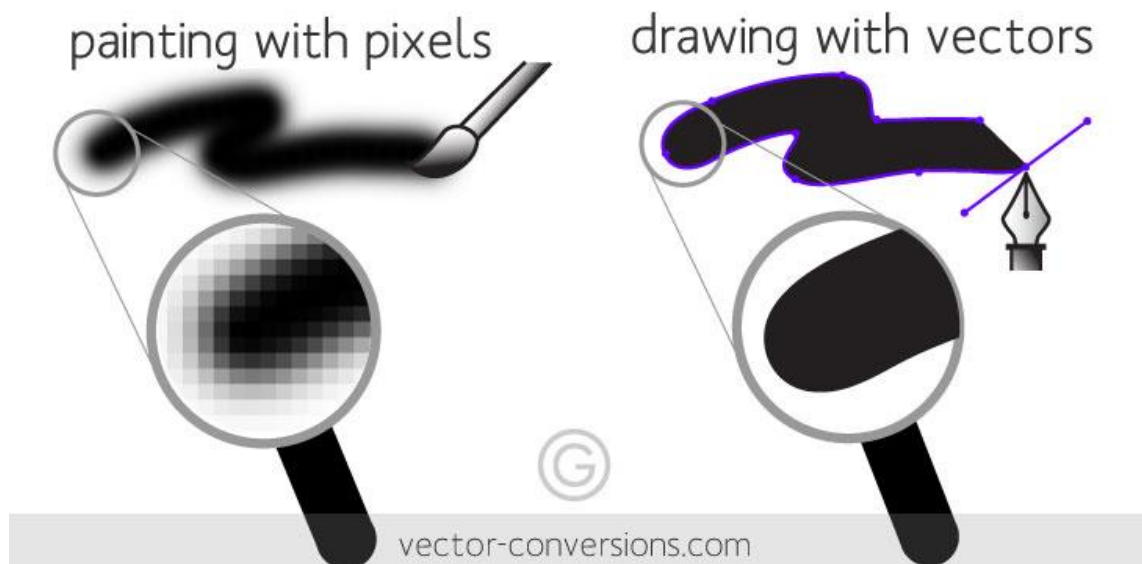
Introduction	5
1 Pixel art in general	7
1.1 Pixel art in video games today	8
1.2 Why Hi bit-pixel art?	10
1.3 Terminology	10
1.4 Purpose and methods of thesis	11
2 Pixel Perfect Resolution	12
3 Colour	14
4 Animation	16
4.1 Sprite Sheet Animation	16
4.2 Skeletal Animation	17
4.3 Using Skeletal Animation in Pixel Art	18
5 Light	20
5.1 Normal Mapping	20
5.2 Emission Mapping	22
6 3D elements in pixel art	24
7 The End Result: Mr. Skullerton's Vault	26
7.1. Used Software	26
7.2 Initial Settings	28
7.3 Colours of Mr. Skullerton's Vault	30
7.4 Light and Shadow	31
7.5 Animating <i>Mr. Skullerton's Vault</i>	33
8 Conclusion	35

TERMS USED IN THIS BACHELOR'S THESIS

Bit (colour graphics)	Storing method for information in an image file, so that each pixel is represented by certain number of bits.
(Display) resolution	Number of distinct pixels in each dimension that can be displayed on computer screen.
Demo	Short for "demonstration".
Game Engine	Necessary software required for a game to run.
Platformer	Video game genre, in which running and jumping on platforms is key element of the game.
Sprite	A bitmap integrated into a video game scene.
Texture	Image file applied to a game object.

Introduction

Pixel, literally "picture element", is the smallest controllable fragment of a picture on a computer screen. A pixel on screen represents a tiny colour fragment and a single image can consist of thousands of pixels. The idea of picture element itself predates computer technology for several decades and has been in use since the invention of film. All pixel-based graphics are also called raster graphics, which can be represented on a rectangular grid of pixels in opposition to vector graphics, which are composed of mathematical paths (picture 1). Nevertheless, pixels are an essential part of visual medium and computer games even today. (Lyon, 2006).



Picture 1. Difference between raster and vector graphics. (image by Vector Conversions)

While single, distinguishable pixels on a screen is a memento from a time since passed, pixel graphics games thrive among video game developers and enthusiasts alike. Many of the critically acclaimed and financially successful video games of the past decade are considered "*hi-bit pixel art*" games; games that use pixelated graphics and various other kinds of aesthetics such as chiptune music. (Madsen, 2020)

This bachelor's thesis will explore the current trends of pixel art and various methods that are used to create the nostalgic experience and further build on the frame

work and limitations of the older generations of video games. While pixel graphics and hi-bit pixel graphics are an ever-growing art form in the game industry, there is not much written information on the subject and the information is often fragmented around the internet and very little in-depth analysis exists. This thesis will explain hi-bit pixel art as it appears in video games today and explain the key factors that make hi-bit pixel art.

1 Pixel art in general

Pixel art, term coined by Adele Goldberg and Robert Flegal in 1982 to describe minimalistic style that constrains itself to creating distinguishable shapes from small amount of picture elements, is an art form that has a long history with video game design. Whether it's the simple yet iconic design of aliens in "Space Invaders" (Taito/Midway, 1978) or the recognizable features of Super Mario in "Super Mario Bros" (Nintendo, 1985), pixel art has been there to pave the way for the most recognizable designs in video games.

Pixel art in video games can be divided in number of different categories. Is the style isometric or non-isometric or rather is the style used to depict side-, top- or bottom view or if the designer is trying to create an illusion of 3-dimensional space? What is the bit depth or how many bits is used to indicate colour of a single pixel? Is every asset done orthodoxically drawing pixel by pixel on screen "by hand", or is it generated from pre-existing materials, such as lowering resolution of a real-life photograph to create pixelated graphics?

The key difference between conventional 2D graphics and pixelated graphics today is hard to determine, since the line between these two formats becomes harder to define as many designers get more and more creative with the use of pixels. Pixel resolution is one of the key distinguishing factors that separates these two art forms, but modern-day pixel art games can have such a high pixel resolution that it can be difficult to tell the difference, such as in case of *Iconoclasts* (picture 2). The most common consensus to this rule is that if individual pixels can be determined on screen, the game's art style can be considered pixel art (Keddy, 2019).



Picture 2. *Iconoclasts* (2018, Joakim Sandberg, Bifrost Entertainment) is considered a pixel graphics game yet the pixels per screen ratio makes it look like it could be conventional 2D art.

Tyriq Plummer, who has worked as an artist on video games such as *Epic Mickey: Power of Illusion* and *Crypt of the Necrodancer*, summarizes hi-bit pixel art, or pixel art as created today, as essentially taking what we've learned from video game graphics of the early generations of gaming and take it to new extremes. It is to enhance the visuals by multitude of improvements and additions that weren't possible to produce during the early days of pixel graphics.

1.1 Pixel art in video games today

In my opinion, despite the graphics technology in video games has taken major steps in creating nearing life-like graphical features, pixel art is still a thriving phenomenon in video game industry. Creating pixel graphics is especially favoured by independent game developers due to variety of reasons.

Pixel art editing software is relatively inexpensive. While you could create pixel art with any raster-based graphics editing software, many of the pixel art editors used in the video game industry today are either reasonably low-priced or completely free and easy to use. Though many graphics editors might have more features suitable for pixel art, such as layers and even animation, even the most

simple and basic graphics editor that allows you to draw singular pixels can be used to create pixel art that can be turned into game-ready assets.

Pixel art can also be used to create a feel of nostalgia. Many creators tend to rely on self-set restrictions based on the hardware limitations of previous console generations to make their games feel like a game from a different time. This can be done by, for example, limiting the use of colours and colour palette, resolution and animation frames. All the while excluding the hard limitations, which didn't enhance the gameplay or the visual style, such as limiting animated objects on screen. (Plummer, 2020)



Picture 3. Shovel Knight (Yacht Club Games, 2014) purposefully imitates the primitive pixel graphics of 8-bit and 16-bit era despite being released in the era of seventh generation of video game consoles. However, while it pays homage to earlier video game graphics, Shovel Knight expands from the limitations of its predecessors by variety of techniques, such as expanding the colour palette, increased frames of animations and having virtually no limit on animated objects on screen. By this definition, Shovel Knight can be considered a hi-bit pixel art game.

These kinds of restrictions can create a challenge on how to represent different characters and objects in small resolution. One way is to scale up all graphical assets' pixel resolution and widening screen resolution to match the resolution of current computer and television screens. More often, though, pixel art is about

overcoming the challenge of creating art within certain boundaries that is both visually pleasing and portrays the necessary details as efficiently as possible.

1.2 Why Hi bit-pixel art?

Hi bit pixel art is a term popularized by Jo-Remi Madsen, co-founder of D-Pad Studio and programmer for Owlboy (D-Pad Studio, 2016) to describe the current era of pixel art video games. In his words, video games that use this art style today are paying homage to the 16-bit era that while they're working beyond the limitations of the technology of that era. Simon Stafsnes Andersen, artist for Owlboy has stated that, like any other art form, pixel art needs to “push the medium even further than before to remain impressive”. (Madsen, J. 2020)

In theory, all current pixel art games are hi bit-pixel art. Things like hardware limitations are history with the current game development hardware and software. Today's restrictions to pixel graphics (such as resolution and available colours) are those the developers impose themselves as to follow the conventional rules of pixel graphics and to give their game a more nostalgic feel (Plummer, 2020).

As limitations for creating computer graphics diminish year to year, there is really no limit of what pixel art can be. The most common nominator between pixel art of the early days of computer graphics and the pixel art video games of the recent years is the pixel resolution; the fact that game that uses pixel art graphics should look like a pixel art to an extent.

1.3 Terminology

While the only known popular term to describe the art style, “hi-bit pixel graphics” may not be entirely accurate. By definition, anything represented on a current screen displays can be hi-bit pixel art as every pixel on screen is hi-bit, meaning that there are no real limitations of the use of colours or bits.

Along researching hi-bit pixel art and its terminology, a few new terms were suggested to describe the same art style: “neo retro pixel art” was used to describe

the new age of pixel graphics and all the current game design trends that borrow from video games considered retro, not just the art style. In addition, “neo retro” itself is a term to describe anything nostalgic and retro-aesthetic. Other terms to describe current trend of modernized pixel art were “second generation pixel art” and “modern pixel art”. (Hewson, 2018 & Oikkonen, 2020)

For the sake of clarity, in this thesis the art style will be referred to as “hi-bit pixel art” or in context of computer graphics as “hi-bit pixel graphics”.

1.4 Purpose and methods of thesis

The purpose of this thesis is

- To assess what hi bit-pixel graphics in video games is and how it shows in current video game trends
- To explore various elements and methods of what makes a game hi bit-pixel art
- To give aspiring video game developers and pixel art enthusiasts a brief glimpse on how certain methods can be used to improve their work

During the course of this thesis, I researched different types of games that have been developed using pixel art and studied their methods. The games I studied are from late 2000s to this day. In addition, I will briefly explore games of the 8-bit and 16-bit era for context. The study and assessment of different methods and techniques were done by qualitative methods which include testing and analysing different kinds of pixel art video games (both current and early pixel graphics games) and testing various hi-bit pixel art methods in practice.

As part of the final product of this thesis, a short game demonstration, *Mr. Skullerton's Vault*, was created to demonstrate some of these methods and techniques of pixel art in modern video games.

2 Pixel Perfect Resolution

In computer graphics pixel is the smallest individual unit of measure on screen. Pixel cannot be divided into fragments, such as a “half-pixel” or “ $\frac{3}{4}$ of a pixel”. The standard display resolution for Nintendo Entertainment System was 256 x 240 pixels and there was no way of increasing that amount. Compared to today, the 8K Full Format screens of today have display resolution that is over 500 higher in pixel density. This makes discerning individual pixels on screen impossible, which is why, to retain the visibility of pixelated outline, video games either zoom in or enlarge the pixel art assets on screen.

While higher display resolution obviously gives the artist more liberty and screen “real estate” than ever before, it can lead to a variety of different difficulties. The first problem that many starting artists fall for is the varying pixel resolution. On devices with higher display resolution, pixels “size” may vary greatly. This may not be just an oversight from the artist but a recurring problem if a game is played on different screen resolutions. Notice in picture 4 how the upper lantern is has the same pixel size as the wall texture while the pixel size of the lower lantern is doubled.



Picture 4. Two lanterns and background wall tiles from Mr. Skullerton's Vault. The upper lantern is in correct resolution compared to the wall texture while the lower doesn't match the same pixel size. In theory, the use of multiple pixel resolutions

can be used as an artistic choice. However, more often it will only break the illusion of a limited display resolution.

To counter this problem, it is important that the game is properly configured to display the same view on different display resolutions. Jo-Remi Madsen made this observation about the screen resolution of the most popular hi-bit pixel art games:

“I've noticed a possible new standard for 2D pixel games trying to achieve wide-screen support. By starting out with a 640x360 resolution, the developer is able to scale it up two times, 1280x720 (HD), 1920x1080 (Full HD) and even 4K, with the pixel graphics still looking great, and retaining the pixels just like they're supposed to!

I noticed this in Hyper Light Drifter, Iconoclasts and even Shovel Knight (known for strictly following the 8-bit NES restrictions)

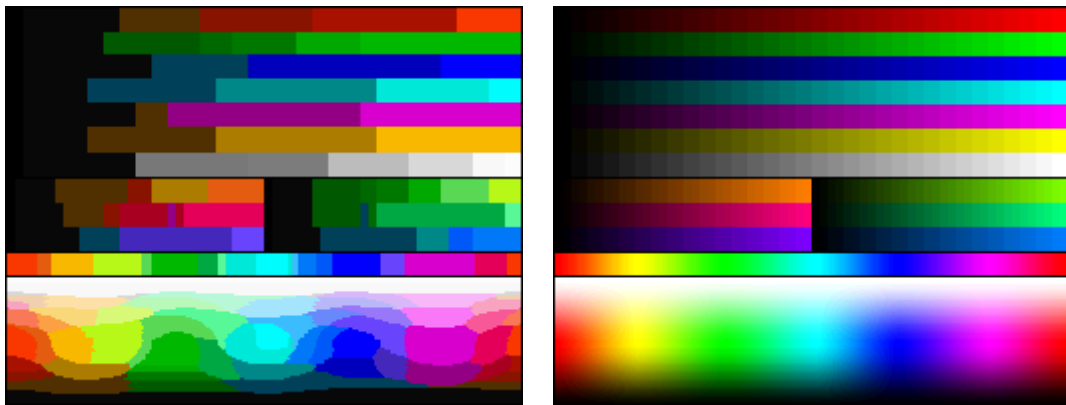
I don't blame them. The old screen formats simply [don't] work anymore. If Shovel Knight had opted to work with the old NES resolution, they would have discovered the same issue we did - the game would have looked terrible on widescreen - and players might've avoided buying it all together.”

By old NES resolution, Madsen means 4:3 format, or full screen format. This format was used in televisions all the way from invention of moving pictures to the 1990s. Madsen also points out that this was also the resolution which Nintendo used for all of its games until the popularity of wide screen overshadowed it.

3 Colour

At the time, the 8-bit Nintendo Entertainment System (NES) had a colour palette of 54 colours. Out of these 54 colours, the palette memory can display only up to 25 colours on screen at a time. Due to these limitations a single sprite can consist of only three visible colours, not including the transparent background colour. Sometimes these limitations could be circumvented by arranging the individual sprites in layers. Though this posed a completely new challenge with the limitation of sprites: out of the 64 sprites that could be shown on screen at the time, only 8 could be displayed on the same horizontal line without flickering. (Cunningham, 2013)

16-bit Super Nintendo Entertainment System (SNES) increased the palette to 32 768 colours, with 256 colours that could be shown on screen at once. Though the actual number of colours can vary greatly depending on the amount of colour layers used similarly to Nintendo Entertainment System. The difference can be observed in picture 5. Compared to more recent Nintendo DS, a handheld console using an 18-bit RGB colour palette, which was capable of displaying 262 144 possible colours of which 32 767 could be shown on screen at once.



Picture 5. The difference between a NES (8-bit console) colour palette and SNES (16-bit console) colour palette. (Colour palettes by Ricardo Cancho Niemietz).

Using the appropriate colour palette of aforementioned consoles is an effective way to emulate the feel of an 8-bit or 16-bit era video game and is often used in many hi-bit pixel art games of today. But while many video games use colour palettes of the older generations as a reference, they may not stick to it as strictly as one would with the limitations still present. *Shovel Knight*, for example uses

the NES palette as a reference, but the amount of colours on screen and even the some of the colours themselves are not strictly as one would remember seeing in an actual NES game. The colours have more depth to them and they're used more efficiently thanks to the current technology.

When designing pixelated game assets, it is also important to consider what kind of outline the assets have or do they have it at all. An outline is an effective way to create definition to any item on screen and can be used to make anything on screen more distinguishable from the background. They are not required to create clear and crisp pixel art but objects without outline need other ways to separate themselves from background, such as using different colour. (Keddy, 2016)

4 Animation

In video games animated elements can be used to form the parts that are the focal point of the player's centre of attention, such as the player character. A designer has to overcome multiple obstacles to make sure that their game objects pop up from the background and that the player character or anything that the designer wants the player to focus on, is always the focal point of the screen. With pixel art the greatest obstacle is the limited size of objects that can make animating and creating distinguishable movement a challenge.

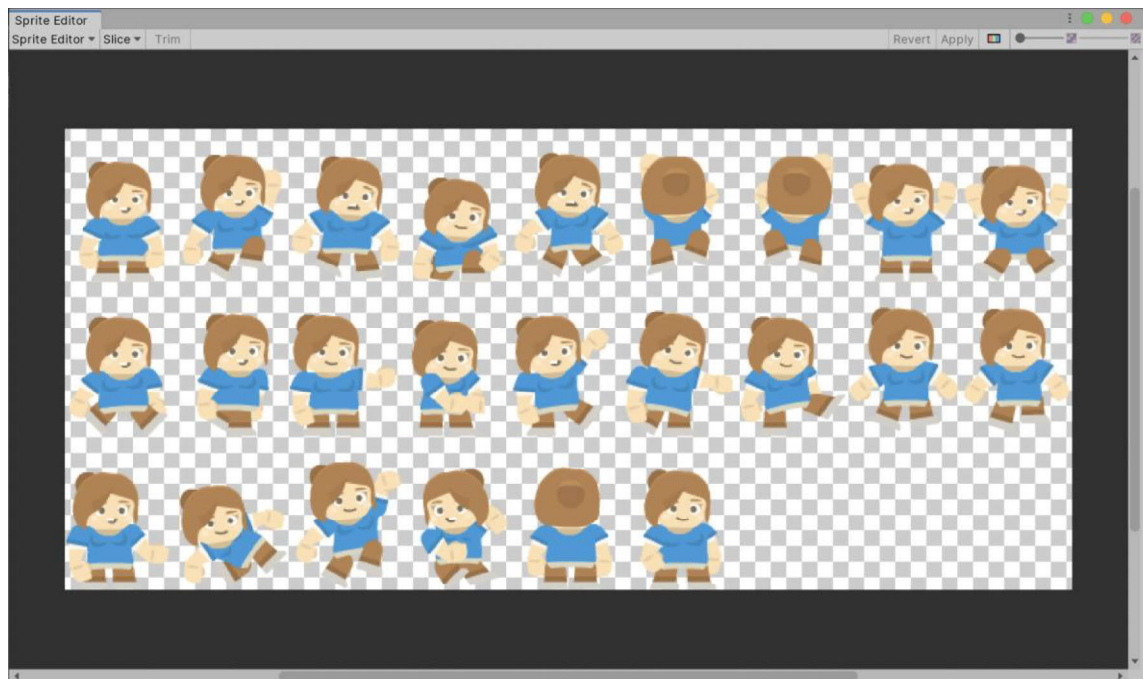
For example, with pixel art, the characters on screen always have more limited pixel resolution. Making something like a simple running animation can be really difficult if the character consists of 16x16 pixels. The most distinct parts that represent a character's movement, such as arms and legs, need to be clearly distinct from the rest of the body. The sprites of Mario in Donkey Kong (Nintendo, 1981) and later on Super Mario Bros. (Nintendo, 1985) are famous for giving character distinctive features which all have a function: Mario's hands and shoes are different colour from the rest of the body, overalls that make the movement of arms and legs more apparent and the pointy nose and hat shows clearly which way the character is facing. (NPR, Miyamoto, 2015)

4.1 Sprite Sheet Animation

Sprite sheet animation (picture 6) is the classic way of animating that has been used since the beginning of animation. In it, the animator draws every single frame of animation by hand. The technique was the only way of animating assets in the 8- and 16-bit era of video games until the use of skeletal animation was made possible in the later years.

One of the benefits of sprite sheet animation is that every single frame can be anything the artist wants, within the limits of the canvas and the amount of animation frames. This gives a lot of freedom in the creation process and the results can be highly vivid. In addition, sprite sheet animation requires no armature configuration or rigging, which is essential to 3D and skeletal animation.

The drawbacks of sprite sheet animation are that if you want more fluid movement, it requires more animation frames to draw. Drawing every single frame of animation is very time consuming and currently there is no way of properly generate frames between key frames automatically.



Picture 6. A sprite sheet in Unity's Sprite Editor. All the necessary frames of animation can be in one sprite sheet image. (Image: Unity Technologies)

4.2 Skeletal Animation

Skeletal animation is an animation technique in which an object is animated using interconnected parts or “bones”. These bones are what is called armature or rig that is used to animate the object. Essentially, each bone is assigned to a different part of the animated object (called “mesh” or “skin”), which can be body parts, such as arms and legs, which can be operated and animated individually.

As such, skeletal animation can be used to animate individual body parts without affecting the entire mesh or, more importantly: without requiring the artist to draw every single frame of animation individually, but by only setting up the key frames.

Skeletal animation is essentially the general way to animate 3D characters in gaming and other media. As a tool for 2D animation it has become widely popular

especially with game engines such as Unity, which has built-in tools for skeletal animation (picture 7).



Picture 7. Unity's bone editing tool for skeletal animation in use. (Image: Unity Technologies)

4.3 Using Skeletal Animation in Pixel Art

Combining sprite sheet animation and skeletal animation isn't out of the ordinary. Different parts of a character's body that can be considered smaller details and which are hard to animate by bones, for example character's mouth, can still be animated by sprites.

Using skeletal animation can create a clear distinction to classic pixel art, since most if not all 8-bit and 16-bit era video games relied on sprite animation. It can be used to create a flowing and vivid movement that wasn't possible during the early days of pixel art. This is due to the skeletal animation ability to freely rotate game objects 360 degrees on a singular axis without needing to draw every single animation frame by hand. This makes the animation seem more fluid without the necessity of drawing every single frame of animation and interpolation, which is a process animation frames between key frames, can be done automatically in skeletal animation.

Skeletal animation also allows the use of deforming a game object. This is a helpful property that, again, reduces time to animate multiple objects but in case of pixel art where animatable objects have a small number of pixels can be problematic. Deforming an object with small pixel density can lead to scruffy and blurry antialiasing form, which may break the illusion of otherwise crisp and sharp edges used in pixel art.

While it has its benefits, in pixel art and in 2D animation in general, skeletal animation is not completely flawless and it makes some sacrifices in favour of smoother movement. Relying only on skeletal animation can give the impression of lifeless and robotic motion.

In addition, as skeletal animation creates a clear distinction between the classic sprite sheet animation of 8- and 16-bit era games and current pixel art games, it can draw player away from the feel of nostalgia.

It can also be very difficult to animate objects or details that consists of small number of pixels, though it is usually completely unnecessary. In video games, where the resolution of animated characters is small and the representation of an animated object on screen is meant to be tiny, it is often not needed to rig every individual joint in fingers or every single facial muscle. In fact, it is often easier to create multiple additional sprites to represent different facial expressions and hand poses.

5 Light

In the early days of video game graphics, something like realistic lighting wasn't a thing. Light could be represented by a single-color shape and light source could be represented as simply as a few flickering pixels that were meant to depict fire in a brazier. This kind of light was usually pre-drawn on the sprite sheet itself, rather than using an active light source. Later on, games would create difference between light and shadow by creating dark gradient layer over the game view or scaling down the draw distance if the game was in 3D.

Nowadays, creating reactive and dynamic light in pixel graphics games is not out of the ordinary and can be used to put the retro aesthetic to a completely new level. Games such as Megasphere (AKGames, 2015) uses several different types of lighting and normal maps to give the game world and character's a very energetic feel while retaining the pixel graphic art style.

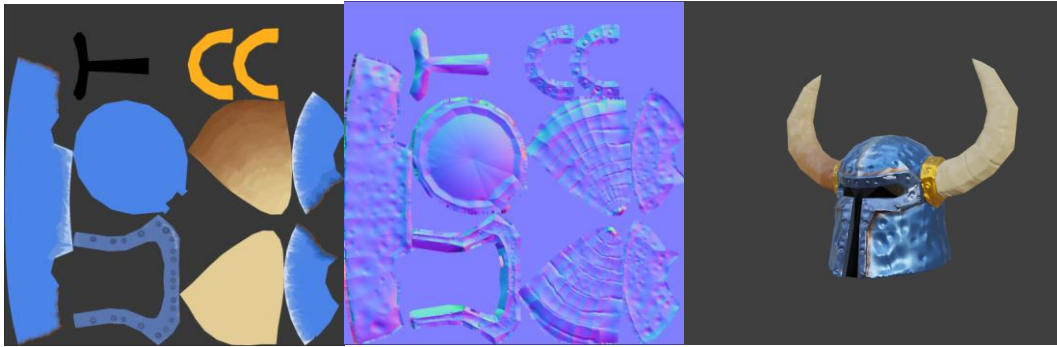


Picture 8. Megasphere

5.1 Normal Mapping

In its most common form, normal mapping is used to create an illusion of depth, lighting and detail on low-poly 3D-objects without affecting the object's resolution. It can be made used to, for example, create bumps and cavities on to otherwise

flat surfaces, without affecting the actual shape of the object. In practice, this is done by creating an additional image texture that represents the depth of the different areas of the object (picture 9). Rather than bump mapping, which uses grayscale values to provide information about the vertical information of the area on the texture, normal mapping uses RGB values to correspond with X, Y and Z axis. (Pluralsight, 2014).



Picture 9. From left: Base colour texture of a 3D helmet, the corresponding normal map and the low-poly 3D object with both textures enabled.

Nowadays, normal mapping is an essential industry practice in 3D graphics (Mikkelsen, 1). Its use in 2D graphics, while the use of normal maps is fairly similar to 3D graphics, is not as common and rarely used in 2D pixel art games. In games that use 2D graphics, normal mapping can be an effective tool to use with dynamic lighting. Essentially, normal mapping in 2D graphics is used to create fake depth to 2D textures that reacts to light just as a 3D object with a normal map texture would. This is especially effective tool to create details on simple surfaces.

There are several ways of creating a normal map. One way would be to generate them using image texture with a computer software that generates the normal map automatically, but, especially in case with pixel graphics, may not turn out as accurate as such programs can be inaccurate due to the significantly lower pixel resolution. It can also be difficult for a software automatically determine the exact depth of any given detail, sometimes even not being able to determine the correct angle from which light is supposed to react to.

Another way of creating normal maps is to simply draw them “by hand”. This can be easily done by drawing the desired normal map on a new layer over the base

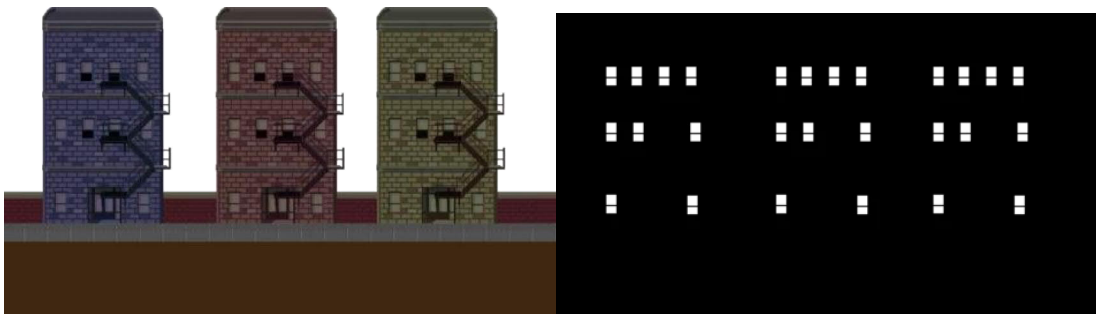
layer using any graphics editor that utilizes layers. While this can be time consuming, in the end it is probably the best way to get the exact result you wish. It also doesn't rely on the use of any additional software, and can be done by the same graphics editing software used to draw the base textures.

Third way to create a normal map is basically the process how you would do so to any texture of a 3D object: you first create the 3D model of the texture and use the 3D editing software to bake a normal map based on that model. This, while might get the most accurate of all the methods mentioned, is by far the most time consuming, since you need to create both the base image texture and a 3D model based on that texture, solely for the purpose of baking the normal map.

In Unity, adding a normal map to a 2D object happens similarly as you would add a normal map to a 3D object. The object in question requires the base texture and normal map which is set up as a secondary texture. To enable the use of normal maps with the 2D lights, it is also required to enable them from the property window of the 2D light object itself. This will activate the normal maps on objects and makes them react to any light objects in the Unity scene.

5.2 Emission Mapping

Emission maps are used to indicate areas on textures that are meant to emit glowing light. Emission maps functions similarly to normal or bump maps as secondary texture (picture 10). This grayscale image texture is drawn on the basis of the base texture by colouring any glowing areas with white or lighter shades of grey and non- or less glowing areas as black or darker shades of grey. (Keil, 2018)



Picture 10. Base texture with emission map texture. With emission map, only the windows of the buildings will emit glow. (Image: Alyssa Keil, Mega Cat Studios. 2017)

Like normal mapping, emission mapping is used in 3D texturing, but unlike normal mapping, it is used in both traditional 2D art and pixel art more generally. Emission maps can be used in tandem with actual light sources to generate more realistic looking light sources.

6 3D elements in pixel art

If you look at the current highest selling and most popular video game titles, you notice that many of them feature 3D graphics and most of them have graphics that are near life-like (NPD, 2021). While graphical aspects of a game may not directly correlate with the game's popularity, many developers in the gaming industry push for more realistic graphics every console generation. In my opinion this may be one of the reasons why 3D games are more popular as the video gaming audience has certain expectations for video game graphics and replicating realism on a 2D plane can be very difficult.

While using 3D elements in 2D games and pixel graphics games seems like a recent addition to the art style, it has been used since the 5th generation of video game console era. Games such as *Doom* (id Software, 1993) and *Duke Nukem (3D Realms, 1997)* mix 3D environments with 2D sprites. More recent uses of this kind of mixed use of 2D- and 3D assets can be found in games such as *Octopath Traveller* as seen in picture 11 (Square Enix, 2018).



Picture 11. *Octopath Traveller*'s developers coined the phrase HD-2D to describe the art style of their game. It features pixelated sprites and textures with low-polygonal environments combined with high-definition effects such as use of dynamic lighting, depth of field and multiple particle effects.

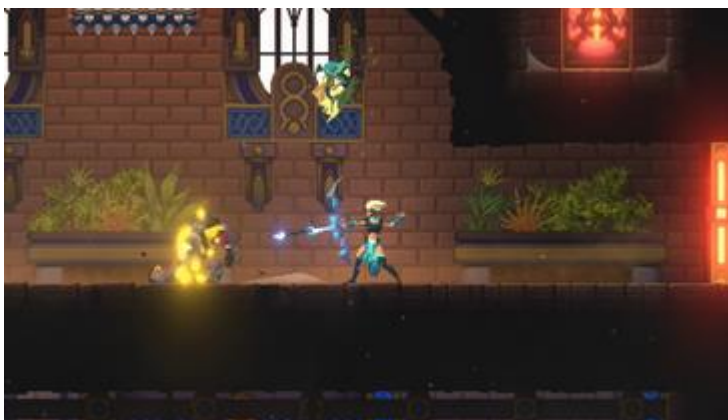
Minecraft (Mojang Studios, 2011), while not a pixel art game, heavily relies on the use of pixel art in its textures. Similarly the wooden crate in picture 12 is a 3D cube but the textures are low-resolution pixel art. While the framework of the

game is 3D, the pixelated resolution used in the textures of the 3D objects may be considered pixel art. While pixel art is considered 2D graphics, there are ways to use 3D elements in pixel art without breaking away from the 2D format, including the limited use of 3D objects or rotoscoping.



Picture 12. A 3D object which uses low resolution pixel art textures.

Rotoscoping is the technique of tracing motion picture footage frame by frame to create realistic looking animations. It relies on 3-dimensional or real-life footage used to create more life-like animation. Jordan Mechner famously used rotoscoping to translate real life imagery into their pixel art animations for his game *Prince of Persia* (Mechner, J. & Broderbund, 1989). Today, a modernized version of this technique might be to use filters and textures to make actual 3D-objects look like 2D-pixel art. Games such as *Foregone* (Big Blue Bubble, 2021) use similar technique to create player and non-player characters using 3D-objects and use filters on 3D object to make them seem pixelated, while the background is created in 2D pixel graphics (picture 13).



Picture 13. *Foregone* (Big Blue Bubble, 2021).

7 The End Result: Mr. Skullerton's Vault

One of the goals of this thesis is a short tech demo meant to represent different techniques of hi-bit-pixel graphics used in video games. It especially is meant to demonstrate the use of 2D tools of Unity's render pipeline, skeletal animation, normal mapping and lighting in pixel graphics setting. The game is called *Mr. Skullerton's Vault*.

Mr. Skullerton's Vault is a simple 2D pixel art platformer in which player controls a character "Mr. Skullerton". Player's must navigate Mr. Skullerton through a dark maze-like dungeon using only a lantern and other various light sources to help them see their surroundings. The player has an ability to run, jump and pick up or throw around lantern. The player must avoid hazards such as pits and spikes while also trying to stay out of darkness as much as possible. If the player stays in darkness for too long, Mr. Skullerton gets frightened and the game is over. The game is won by successfully navigating to the end of the dungeon.

Mr. Skullerton's Vault is designed for the purposes of demonstration of the specific features of hi-bit -pixel graphics and to test these features in practice. The process of creating *Mr. Skullerton's Vault* is described further in this thesis.

7.1. Used Software

The following software products were used to create *Mr. Skullerton's Vault* and to explore various aspects of hi-bit -pixel art.

Unity

Unity is a game engine developed by Unity Technologies. It is regarded as highly accessible and easy to use game engine which supports over 25 different platforms from consoles to mobile systems. While mainly focused on 3D games, such as *Fall Guys: Ultimate Knockout* (Mediatonic, 2020), *Subnautica* (Unknown Worlds Entertainment, 2018) and *Pokémon Go* (Niantic, 2016), Unity can be used equally well to produce successful high-end 2D games, such as *Hollow Knight*

(Team Cherry, 2017), Among Us (Innersloth, 2018) and Cuphead (Studio MDHR, 2017).

While there might be more suitable and simpler software for creating pixel graphics games, Unity was chosen specifically for the task to test the abilities and techniques of hi-bit pixel art, which can require more high-end solution than any basic game development software could provide. Most of properties are either built-in Unity or its add-ons or can be replicated in various ways.

Unity's Universal Render Pipeline (URP) is a Scriptable Render Pipeline for Unity engine introduced in Unity 2019.3. URP provides workflows that let you quickly create optimized graphics across any Unity supported platforms (PC's, consoles, mobile and even VR devices). Unity's Universal Render Pipeline provides optimized performance for developers targeting a broad range of platforms by making some trade-offs with the lighting and shading. Essentially URP or any other rendering pipeline is used to mathematically calculate what steps a software needs to take to render an image that is displayed. This process is used to maintain fluid performance while minimizing the costs of the use of heavy assets that might hinder the process. (Unity, 2021)

Unity's current render pipeline supports both 2D and 3D games development, with various tools and add-ons that support and enhance 2D graphics and even specific tools for pixel-perfect graphics. It doesn't require any additional settings to create 2D games, although it is advisable to configure the project as a 2D project in the editor, which can be done easily from Unity's Project Settings menu. (Alvarado, 2020)

Other suitable software for 2D game engines:

- Godot
- Game Maker
- Solar 2D

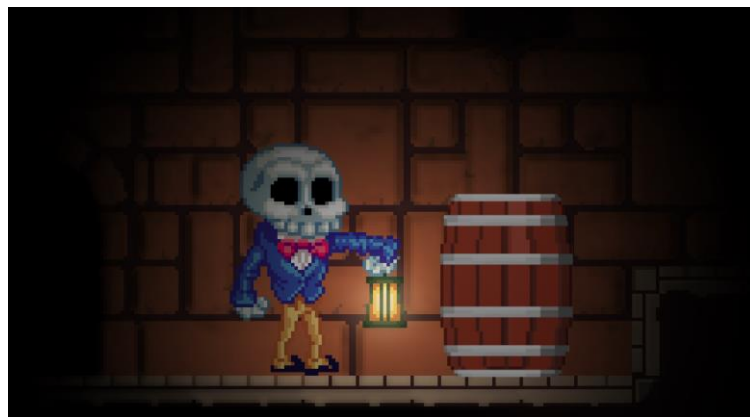
Adobe Photoshop

Adobe Photoshop is a raster graphics editor software developed by Adobe Inc. Adobe Photoshop was used to create all the visual assets such as character sprites and tile maps for *Mr. Skullerton's Vault*. Adobe Photoshop was chosen due to its versatile features in graphics editing, which include proper tools for creating pixel art, such as pixel perfect editing, use of layers and animation tools.

All assets were created in Adobe Photoshop CC 2019 version 20.0.5.27259.

Other suitable software for pixel art:

- Aseprite
- Gimp
- Krita
- Piskel
- Pyxel Edit



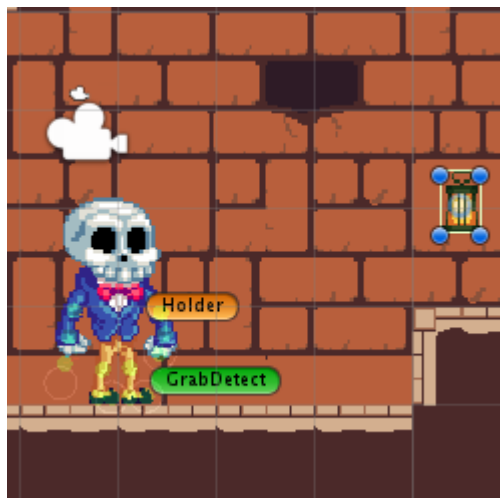
Picture 14. Mr. Skullerton's Vault.

7.2 Initial Settings

When starting the project, the goal was to create it in the Unity's Universal Render Pipeline. This can be done immediately at the project start up or by selecting it via Unity's Package Manager. It is preferable to choose the proper render pipeline at the start of the project so that it is enabled from the very beginning. To help you configure the pipeline, URP Asset can be created in the project window.

Unity's base settings for any 2D assets are not applicable for pixel perfect graphics straight away. However, any 2D asset can be applied to perfect pixel art settings. For the most basic settings any pixel art asset any compression and filter mode settings should be removed (or Filter Mode to be set to "Point (no filter)"). Additionally, the pixels per unit (PPU) needs to be switched so it corresponds to the proper size of the pixels viewed on screen, preferably so that the size of each pixel is the same on screen for all the assets used.

Unity's 2D tools offer a variety of helpful tools that will help to create a pixel perfect resolution of a game. Pixel Perfect Camera is a fairly new addition to Unity's 2D components. Its basic function is to keep the integrity of pixels of all pixel-based motion and visuals on screen. This could be one of the most essential tools to maintain the pixel integrity and correct pixel size when working with pixel art assets as it keeps the assets from any blurring or deforming. As seen in picture 15, all the art assets share the same pixel size on screen.

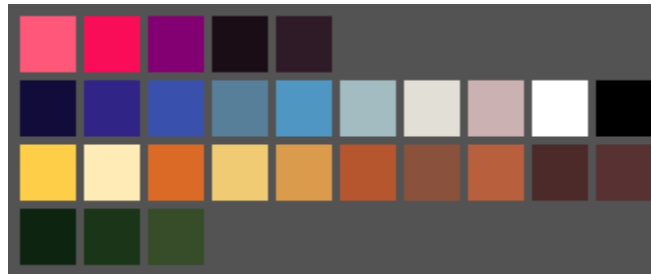


Picture 15. The aim for Mr. Skullerton's Vault was, that, whether it was the individual objects that were animated in skeletal animation or the edges of light and shadow, each pixel on screen was portrayed in consistent size.

7.3 Colours of Mr. Skullerton's Vault

For the visuals, the plan for *Mr. Skullerton's Vault* was to use a colour scheme (picture 16) that was a sort of homage to the 16-bit era video games without fundamentally using the exact same colour palette or colour scheme of a certain game. The idea was to use a retro-style colour palette without restricting the use of colours. The most important about choosing the colours for *Mr. Skullerton's Vault* was that the use of colours was as practical as possible, meaning that the characters and objects would stand from the background.

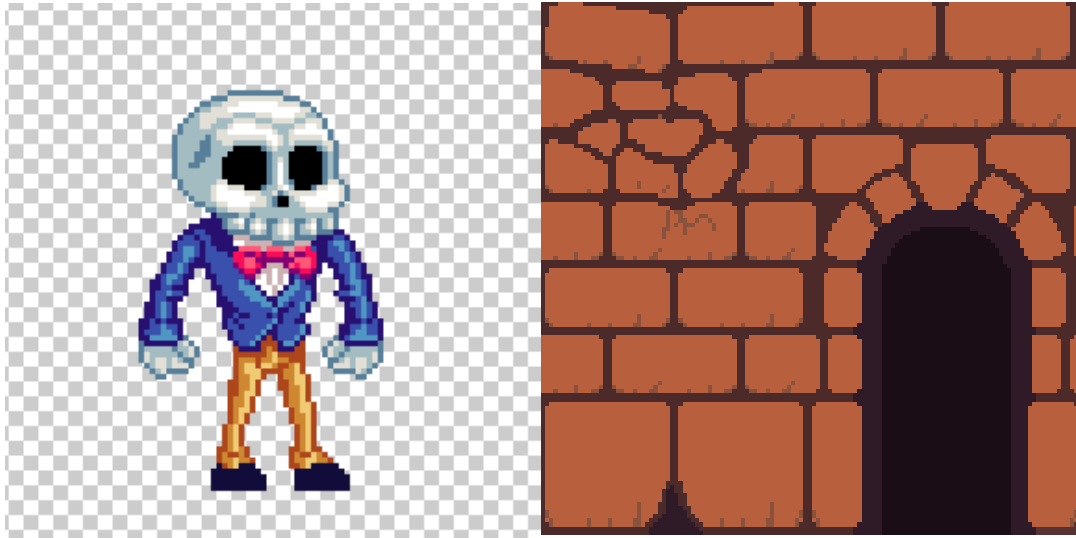
As the game is set in a dungeon, the tile set was given a more reddish brick feel, with lighter tones on the floor. The tile set's base texture was given a rather plain look, while more precise details, such as cracks and crevices were done by normal mapping. The idea behind this was to create a more vivid background that reacted to light. Mr. Skullerton, the player character itself being an actual skeleton, in contrast, was given more of a colder outlook with more bluish colours, with the exception of the bright magenta bowtie that would help player pay attention to the character.



Picture 16. The initial colour palette of "Mr. Skullerton's Vault". While not trying to limit the use of colour too much, the colour palette that was used in Mr. Skullerton's Vault is reminiscent to the 16-bit era games with added colours.

From animation perspective, it was important that Mr. Skullerton's movement was readable not only from the background but also from that his own colours would not clutter with each other. This would mean that the character's colours would need to be distinctive from each other so that any animation is clear to the viewer, as seen in picture 17. The outline of Mr. Skullerton was drawn in the darkest tone of the surrounding base colour to further drive the 16-bit era angle to the character

and to give limbs and detail more definition. Completely black outline or no outline at all would probably work better for a game that emulated the feel of an 8-bit era game.

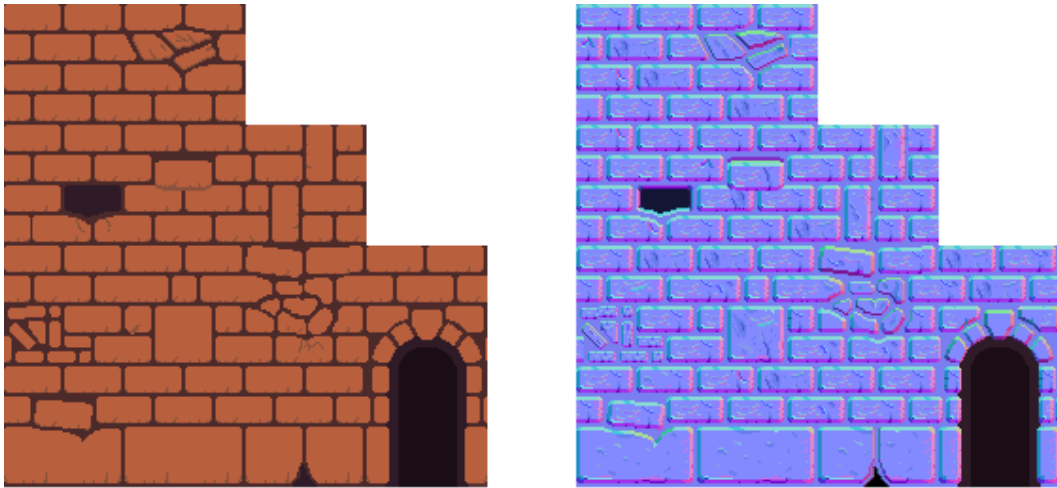


Picture 17: the player character Mr. Skullerton's colours are chosen to make clear distinction between different body parts. His legs and torso are counter coloured, his head and legs are contrasted more highly compared to the background.

7.4 Light and Shadow

The use of lights and normal mapping may be the most distinguishing factor that separates classic pixel art games hi-bit pixel art. This is because the technology to produce realistic light and shadow was nearly non-existent during the 8-bit and 16-bit era of gaming. As such, adding dynamic light to a pixel art game might be one of the most efficient ways to implement hi-bit pixel art aesthetic.

In *Mr. Skullerton's Vault* the idea was to have the details on background tiles done mostly by normal maps as seen in picture 18. This meant doing cracks and holes on brick wall to be done by normal maps, giving them a slightly increased illusion of depth. As light is a crucial element in *Mr. Skullerton's Vault*, the light would react naturally to surroundings.



Picture 18. The base texture for background tile map and the corresponding normal map.

Normal mapping can also be implemented to organic objects such as the player character. This will further enhance the depth of the game scene as seen in picture 19. The process of implementing them is similar to applying them to any other kind of texture and the results are fairly similar.



Picture 19. Mr. Skullerton's head in a light Unity scene with normal maps (left) and without normal maps (right).

While the player character's portable lantern object is an active source of light itself, emission maps were used to create a more emissive light source. A simple emission map texture was created on the base lantern texture that would light the parts of the lantern that would, realistically, emit light to give it a slight glow effect.

With Unity's 2D light tools, pixel art can be elevated to a completely new level. It gives a modern look to a classic video game art style without completely breaking away from the retro aesthetic. The process of drawing the normal maps on individual assets can be time-consuming but will lead to satisfactory results.

7.5 Animating *Mr. Skullerton's Vault*

The animation and specifically the character animation in *Mr. Skullerton's Vault* was done in skeletal animation to test how would a new type of animation work in a retro-aesthetic setting. While sprite animation would have been more traditional to pixel art, skeletal animation can be used to create more fluid animation even with pixelated assets.

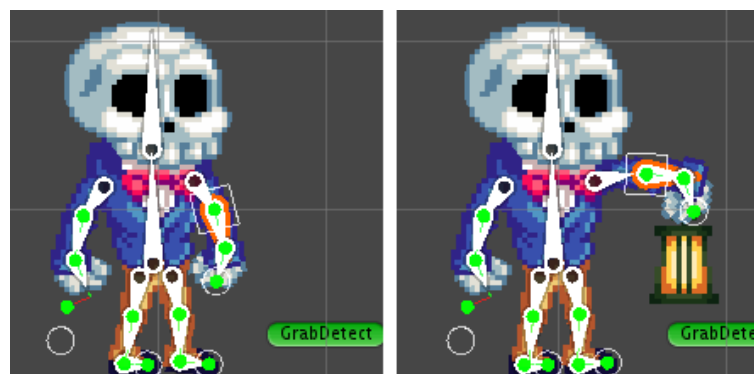
The player character, Mr. Skullerton, is nearly fully animated in skeletal animation. Mr. Skullerton has total of 2 primary animations: idle animation and running animation and 1 secondary animation, holding out his lantern, which can be viewed in tandem with any of the primary animations. The only exception being the character's hand which holds the lantern, which changes to a different sprite if he is holding the lantern or not. This allows a more fluid movement of and freedom in animation for each individual limb.

The process of animating a pixel art character using Unity's own animation tools and skeletal animation follows the same conventions as animating any 2D character. The process includes drawing each animatable item separately on a different layer using a graphics editor (using Unity's PSD Importer plugin, an editor such as Adobe Photoshop recognizes each layer in a PSB-document as an individual item or "body part") or having each animatable item separated in a PNG image file.

To obtain non-distorting effect on individual pixels, it is vital that that the basic pixel perfect settings of Unity are enabled and that pixel perfect camera is enabled. This will keep the look of animated objects consistent and also makes sure

that no pixel is deformed or blurred due to anti-aliasing even when rotating animated items. These settings proved to be mandatory to retain the feel of a pixel art game.

Rigging the different limbs of the player character was essentially similar to rigging any 2D character in Unity. Each body part of the player was pre-cut in existing layers in Adobe Photoshop, which made the process of working on individual limbs easier in Unity. As the animation on the limbs was fairly simple and I tried to avoid as much of deforming and blurring of pixels as possible, all areas of an individual limb have nearly the same weight to them. This gave the limbs of the player character a little stiff or paper-doll-like movement, but it felt appropriate for the retro aesthetic that was tried to achieve. The process of the animation can be observed in picture 20.



Picture 20: The player character in *Mr. Skullerton's Vault* is fully animated in skeletal animation. Each part is represented by a different mesh object and "bone" object so that no deforming may occur. The idle animation is the same in both instances, with the default idle animation (left) and the running animation while holding the lantern (right). The idle animation is unaffected by the secondary lantern holding animation, which are both separate animations.

Mr. Skullerton's armature also uses inverse kinematics or IK which, in the context of skeletal animation, is an animation tool to calculate the proper movement of joints of a character. In the case of *Mr. Skullerton's Vault*, the IK is used on player character's arms and legs to make animating these limbs easier and to make the movement of these said limbs move more naturally as each joint in the chain lives in synergy with each other. The use of IK is an effective way to prevent character's bones to exceed their motional limits and to create more realistic movement.

8 Conclusion

The pixel art trend in video games has come a long way since its inception. The current generation of game developers and artists have developed the art style further than one could've ever imagined during the early days of pixel art. While heavily lending techniques and elements used in traditional art 2D games and even 3D games the simple shapes used in pixel graphics has pushed the medium to another level.

The software used to test and enhance the pixel art graphics in *Mr. Skullerton's Vault* proved successful in every regard. While its artistic scope was simple, the features that make pixel art hi-bit pixel art enhanced the simple graphics to a new level. Each different technique of hi-bit-pixel art that was sought to work, could be put into the project.

While there might be software and tools that are better suited for pixel graphics games, Unity Game Engine was most certainly useful to test and use the different mechanics such as lighting, normal mapping and animation, among any other enhancing features used in the process of this thesis. While the use of some of these techniques were difficult at times, in the end Unity turned out to be efficient and reliable tool in creating enhanced pixel graphics.

During the process of this thesis and end product *Mr. Skullerton's Vault* it was realized that while pixel art can be a very confined and constrained art form, its limits are what are iconic to the format and is what has kept it going for decades. Pixel graphics in gaming are still popular among game developers and players alike and current trends show that it will continue to be so as long as it maintains its availability to the masses. And as is the case with all the other forms of art it needs to keep pushing beyond the current norms of itself to maintain its popularity and to this day it certainly has.

REFERENCES

Alvarado, B. 24.4.2020. Configuring a project to use the new 2D Pipeline and 2D lights from the Universal Render Pipeline in Unity. Read on 20.5.2021

<https://www.bryanalvarado.xyz/posts/unity-2d-lights-urp>

Cunningham, A. 15.7.2013. The NES turns 30: How it began, worked, and saved an industry. Read on 17.3.2021

<https://arstechnica.com/gaming/2013/07/time-to-feel-old-inside-the-nes-on-its-30th-birthday/2/>

Hewson, R. 4.6.2018. Neo-Retro: A design philosophy for Hyper Sentinel

https://www.gamasutra.com/blogs/RobHewson/20180604/319235/Neo-Retro_A_design_philosophy_for_Hyper_Sentinel.php

Keil, A. 8.11.2018. Emission Map Guide for Artists. Read on 5.3.2020

<https://megacatstudios.com/blogs/game-development/emission-map-guide-for-artists>

Keddy, S. 14.6.2019 Pixel Art: Where to Start. Read on 10.1.2021

<https://lospec.com/articles/pixel-art-where-to-start/>

Keddy, S. 4.4.2016 Pixel Art Outlines Tutorial. Read on 26.4.2021

<http://samkeddy.com/pixel-art-outlines/>

Lambert, S. 26.11.2013. An Introduction to Spritesheet Animation. Read on 1.2.2021.

<https://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>

Lyon, R. F. 2006. A Brief History of 'Pixel'. IS&T/SPIE Symposium on Electronic Imaging.

<http://dicklyon.com/tech/Photography/Pixel-SPIE06-Lyon.pdf>

Madsen, J. 2020. The High-Bit Era. Read on 7.1.2021.

<http://dpadstudio.com/Blog/postHibit.html>

Mikkelsen, M. 2008. Simulation of Wrinkled Surfaces Revisited
<http://image.diku.dk/projects/media/morten.mikkelsen.08.pdf>

Normal Maps for Pixel Art. 5.1.2019. Watched on 16.1.2020.
<https://www.youtube.com/watch?v=gUkY8ZoRfuQ>

NPD. Best Selling Video Games. Updated 30.4.2021. Read on 31.5.2021

NPR. 19.6.2015. Shigeru Miyamoto On The Origins Of Nintendo's Famous Characters. Read on 13.3.2021
<https://www.npr.org/sections/alltechconsidered/2015/06/19/415568892/q-a-shigeru-miyamoto-on-the-origins-of-nintendos-famous-characters?t=1618330243549>

Oikkonen, T. 2020. Interview material.

Plummer T. 9.9.2020. Skill-Building Series: Pixel Art and Animation in the Hi-Bit Age. Watched on 16.1.2021
https://www.youtube.com/watch?v=ZR_JBgNEeVM

Pluralsight. 2014. Eliminate Texture Confusion: Bump, Normal and Displacement Maps. Read on 7.11.2020
<https://www.pluralsight.com/blog/film-games/bump-normal-and-displacement-maps>

Universal Render Pipeline overview. Read on 9.3.2020. Updated on 9.3.2020
<https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@11.0/manual/index.html>