Technical Design Document
2d Project
Jack Barnfield

Table of Contents

Technical Design Document
2d Project
Jack Barnfield

Character

Research –

RESEARCHING SPRITE SHEETS

SUPER MARIO BROS. (1985)

do not change. The sprite sheet also features sprites for the transition from 'small mario' to a larger Mario entitled 'Big Mario' in the case of the sprite sheet. There are 3 frames that fit between Mario been small and Mario being big. Going back to the amount of space on a NES Cartridge now, this may have been all they could afford to use for each character. As you can see Luigi has the same number of frames for growing and shrinking. Nintendo before creating any of the sprite sheets for the game could have allocated a certain amount of data for each sprite within the game. While this will not need to be a consideration for the games I am making this semester if I go onto making 2D games in the future with much higher scopes this may need to be something I consider.

Another thing of note is that Mario's Idle is only made up of one state. I am planning to have multiple when I create mine so that the character is never feels stationary. They may just not have wanted the characters idles to have animation or this again could have been something they had to choose to do to reduce the size of the game. Something which I do not have to think about thanks to the technology that is available today.

RESEARCHING SPRITE SHEETS

SUPER MARIO BROS. (1985)

In the 'Super Mario Bros.' Game from 1985 there are over 43 sprite sheets. These sprite sheets make up the entire games visuals from, seemingly mundane items like the 'brick blocks' right the way to main characters, like Mario and Luigi. Mario and Luigis Sprite sheet is below. The source of these images is "The Spriters Resource." These sprite sheets are not from Nintendo themselves but rather the admins from this website's idea of what the sprite sheets for the game might look like. As they have hundreds of these sprite sheets for hundreds of games I am going to analyse these as if they are the ones used for the games.

First of all, Mario and Luigi are made up of 3 colours each, this was more than likely because the console this game was made for was an NES. An NES cartridge could only hold 40 kilobytes of data. These colours make Mario Instantly recognisable and we can only assume succesful as Nintendo have chosen to stick with Marios 'style' throughout the Mario games. This highlights consistency within not just pixel art but games and characters in general throughout series as a good thing. This maybe slightly obvious but thoughout the game Mario does not change what he is wearing as well as the character not straying from the mainly red visuals. Something that does change within the game is Mario's scale. It is important to note that his proportions

Technical Design Document
2d Project
Jack Barnfield

RESEARCHING SPRITE SHEETS

SUPER MARIO BROS. (1985)

Another thing I think is worth talking about is how the difference between the characters big and small versions is actually very miniscule, apart from the differences because of their size allowing for more bend in the arms of the characters each frame of the animations are effectively scaled versions of each other. One noticable difference I can see, is in what I presume is the characters crouch. Looking at Luigi as an example, the big version depicts the character almost squating while the small version looks as if they are sitting. I can only pressume that these poses did not translate to a different scale well as this is the only difference in the two variations of Mario and Luigi hence why they might be so different in the big and small versions.

Each frame in the characters jump specifically are not that different from each other, so if my comments about the space on a cartridge are true, nintendo must have allowed so many frames for the jump. Mario's jump throughout the franchise is widely regarded as an excellent example and has been examined for years. The NES ran games at 59.94fps according to the Reddit user 'nodogo'. This means that each of the frames for the jump of the characters is played multiple times as a jump lasts around half of a second. Mario's jump lasts roughly half of a second, so roughly the frames that make up a jump are played for

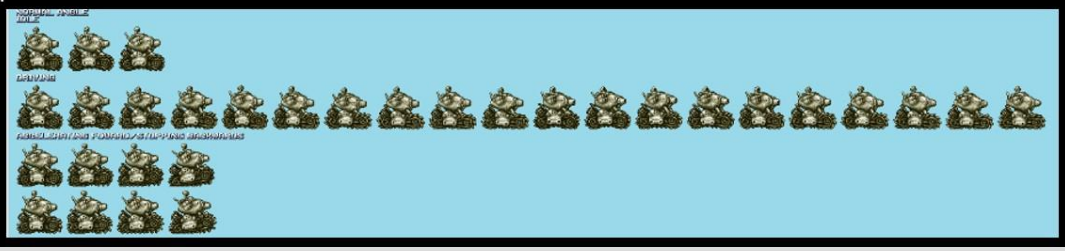
RESEARCHING SPRITE SHEETS

METAL SLUG

In Metal Slug, the players character, the tank has loads of different sprites allowing the player, to name a few enter, jump and move. The image below shows the sprites for when the player is idle and driving as well as sprite for when the player is either accelerating or stopping. To create this character the developers must have used montages. As the player would be performing some of the actions that the sprite sheet has visuals for at the same time.

Upon evaluation of the entire sprite sheet and not just what you see below, they have produced sprites for each orientation the player can be in. They have sprites for when the player is on a 10% incline as well as a 45-degree incline. It is interesting that they chose to draw new sprite for this and not just rotate the sprite below.

Metal Slugs sprites were not drawn on Photoshop and instead on a program called Deluxe Paint on a Amiga 3000, despite this the art in this game is referred to as having the best sprite art in some people's opinion on the Adventure game studio back in 2006..



What is the character based on?

As you can see from the images above, I did research into some other side-scroller games. I found that often the scale is not the same as it would be in real life. The tank in Metal slug for example the turret is near enough the entire length of the main body. In Mario the character especially when they are small is scaled not as you would expect.

I put this down to appeal, if the characters especially the tank was “realistic” the game would have lost its visual “charm”, another thing I notice looking back at some gameplay of

Metal slug is that when the player shoots, they do not actually shoot from the turret, instead the mid-section of the tank.

Likewise, Mario doesn't have the same metrics as a human might but again, I think this is down to appeal. Most people play games for escapism and the main character looking "less-human-like" will add to the players reason for playing which is likely to relax. I would assume it also helps the player separate themselves from the character they are playing as so when their character dies, they don't have that attachment. That may be slightly deep, but I feel it is something to think about when I start to create my character.

What are the abilities based on?

RESEARCHING CONTROLS IN 2D GAMES

CELESTE

In Celeste the player can dash. A dash allows the player to go faster and further than they normally would. The Player can perform this dash while they are in midair as well as after they have fallen off a platform, this is often referred to as a coyote timer. This term was coined when people looked at the animation style of Will E. Coyote, his animations often featured a slight delay between an action and the consequences of that action, hence why the community now refer to these types of mechanics as a "coyote timer."

To create the dash from celeste the developers must have added a force to the player's character, rather than changing the speed of the character as the dash is strongest as soon as the input has been pressed and tapers off in the next second or so. To refresh this dash the player must touch another object, for 10-15 frames.

The community have explored these given mechanics and took them further whether the developers meant it or not, I suppose this is true for all games. This exploration leads to online forums talking about a game, meaning that more people are going be exposed to your game when for example the forum is shared to reddit. So, despite the developers maybe not intending for some of these methods of play, the community have discovered some of these as methods to make the game easier. This often goes towards making the game more popular than it already was in the context of speed running)

RESEARCHING CONTROLS IN 2D GAMES

CELESTE

The dash in celeste is integral for progression inside of the game. The dash itself lasts for around 12 frames. In the image below the player needs to jump further to the right to exit the room. To jump to the platform to the right of the character the player must dash otherwise they would fall downwards, Thanks to the design within celeste in this case would probably not kill the player and instead take them to another level, this may be a level they have previously completed to get to this room, this may be a level in which there is a secret this is half of the charm for Celeste and is part of the reason for Celestes Popularity. I think what sets Celeste apart and why it is ranked so high in terms of quality is mainly for the level design, there are hundreds of different levels and not even one of them are the same.

If Celeste only had movement and jump keys it would be a drastically different game. This is something we see within all 2d games, they all have something that sets them apart, a USP (Unique selling point.)

RESEARCH INTO INTERACTS

MOONLIGHTER

Chests in Moonlighter can only be opened once all enemies in the room have been eliminated. Once all enemies have been eliminated and the player walks up to a chest, they will be given a prompt to open the chest using the A button if they are playing with an Xbox controller. When the player hits that button the piece of UI in the bottom right will appear. Within my game I plan on adding sticky notes that the player is prompted to read, these will give the player a brief narrative. When the player presses a button around an interactable note, a piece of UI will appear similar to how the UI appears in Moonlighter.

ATTACK AND INTERACT SPRITE SHEETS

INTERACT SPRITES

In most 2D games interacting with an environment or another character for example. Opens a piece of UI rather than showing an animation on your character. This piece of UI often hides the character from the player. The image below is a screenshot of what happens when the player interacts with a chest in Stardew valley. This piece of UI in my opinion prioritises usibility, just to pick one aspect out of it, the sprites of the bag and chest of the left edge of the UI make it clear to the player which screen is their inventory and which is the chest.

This Interaction will be triggered when the player is close to a chest and presses interact, typically 'E'. This piece of UI then pops up instantaniasly. Another thing intracting with a chest will do is slightly darken the background. This means that the player can pay more attention to what is in the chest. Another thing to not is that in the case of a chest, while bringing the UI up it also pauses the game, again meaning the player can pay more attention to what exactly is in a chest.

With the knowledge I have I would build this system with multipul blueprints utilizing communication. I would do this because I doubt it is only when the player opens a chest that the game pauses. For example, in the context of stardew valley, when the player interacts with an NPC behind the counter of a shop, I would expect that to pause the game as well. I would have all these interactions call an event dispatcher. The UI would then bind to that, so a signal was sent through that blueprint once an interaction has happened. This could then open the UI as well as pause the games world.

In this game the player can interact with their environment. So above is some research on the interact inside of Stardew valley. Stardew Valley is a top-down 2d game so while the gameplay is likely to be different from my game the overall art is going to be drawn from a different perspective, I can still infer some things from;

a) The overall look of the animation
b) Have a glimpse at the system and what pressing interact does.

First, the player has to have the correct tool equipped to interact which each object. For example, in Stardew Valley the player can not clear grass with their pickaxe. Instead, they must use a scythe. Similarly, when the player is cutting down a tree, they will need their axe. This while making it complex for the player it adds an extra “mechanic” to the game as the player when they leave their house or wherever they keep their tools, needs to have the correct tools in their inventory. This then leads to the player trying to take as little as possible so that they can pick up everything that gets dropped on the floor around them.

As you can see from the image on the left. When the players inventory is full the player will not be able to pick the item up.

RESEARCH INTO ATTACKS

SHOOTAS, BLOOD, AND TEEF

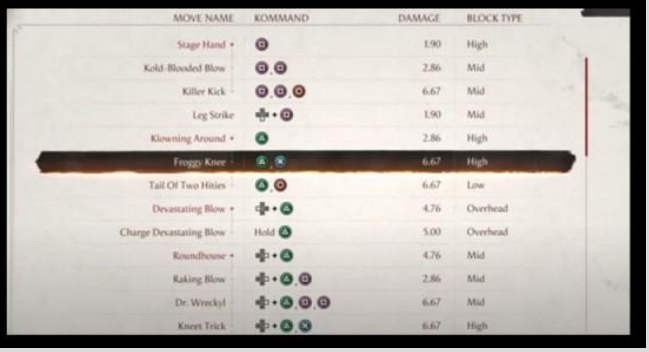
This game has a ranged attack, I plan to add a melee attack in my game. I have made this decision as I feel it fits with the ideas I have for the environment better. For this reason, I am going to look for a game in which, the player can only use a melee attack.

RESEARCH INTO ATTACKS

MORTAL KOMBAT

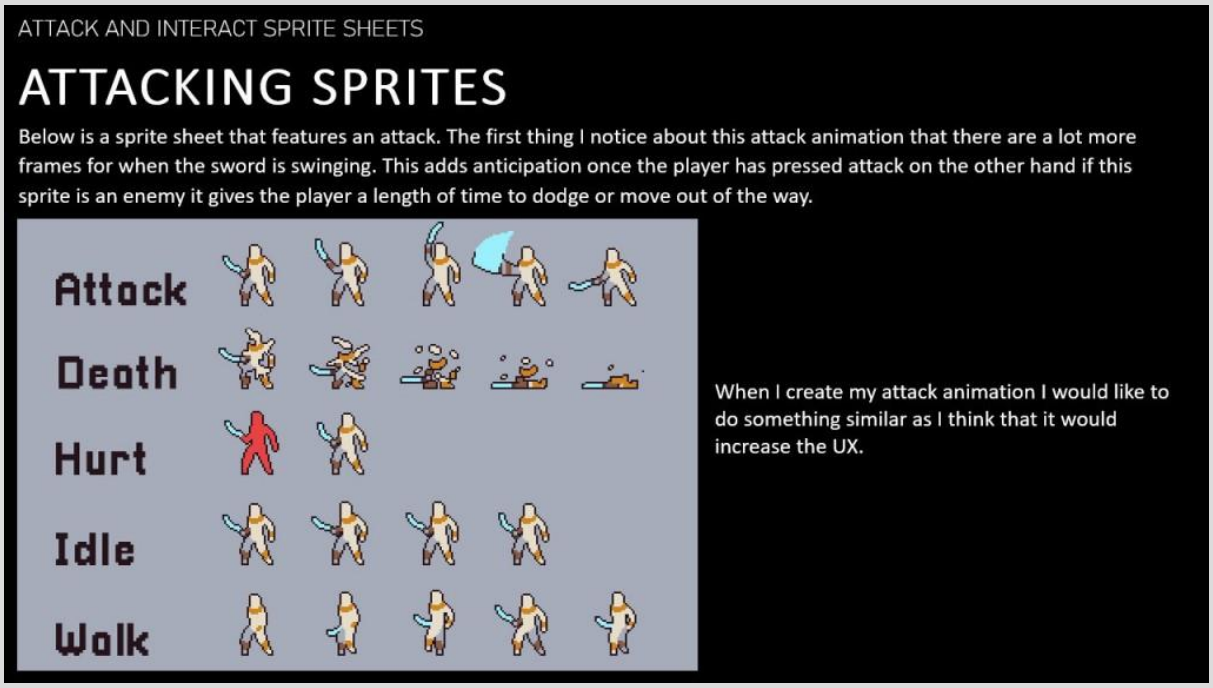
Mortal Kombat is an example of a game in which the player can use melee attacks. Each attack has a different animation and throughout the Mortal Kombat series if the player presses a series of buttons they will do a special attack. Special attacks are different for each character.

Attacks in the Mortal Kombat Games will thrust the players forward so that their characters fists or feet can hit the player when they stretch. This must mean that the amount the character reaches for the attacks is an integer inside of the engine, if the amount of units between the two characters is less than that integer the engine will move the attacking character so that the distance between the two characters is less than the range of the attack.



The attack in this game will be no where near as complex as the attacks in the mortal combat games however I can still draw from them. I'm not sure there is a single example of when a player takes damage without a visual cue from the enemy. This is true of most games where the player can take damage, the player very rarely will take damage out of the blue. Therefor I need to have an attack flipbook present in my game.

As the character in my game is "squished" I think it would make sense to use the characters leg, kicking as the attack animation. This should make it clear to the player exactly what they have done.



Production –

Movement Sprite sheet-

Attack and Interact Sprite sheet-

What are the controls based on?

RESEARCHING CONTROLS IN 2D GAMES

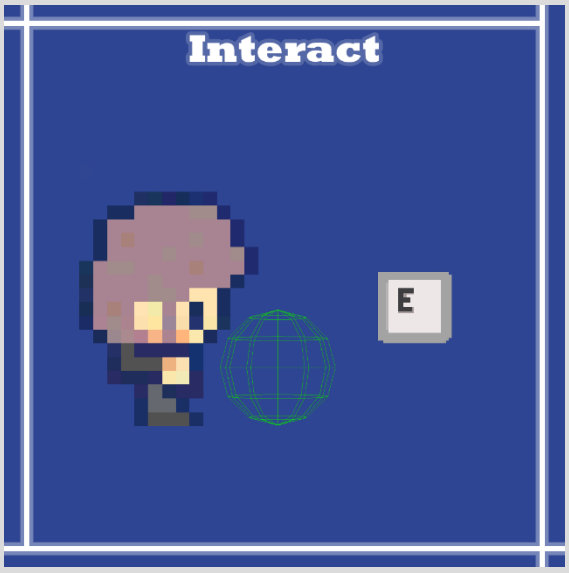
GETTING OVER IT

In Getting over it, a game by Bennett Foddy, the player tries to ascend the level as quick as they can. This is another game with a huge community around it. What makes this game unusual though is the controls. I say controls, what I mean is control. This game does not use a keyboard at all despite it being a PC game, instead it relies on the movements of the player's mouse. It will not track left clicks, right clicks or the space bar being pressed. When the player moves their mouse the characters hammer will follow. The further the players mouse is away from the character the more the character will stretch, this does reach a limit though, so throughout the level the player has to "hook" onto objects and "swing" their mouse this will propel their character in the direction against which every way they are swinging their mouse.

To achieve a mechanic like the one in this game. I would first track the position of the mouse I would then add an empty GameObject on the furthest end of the hammer in the characters hand, as well as one on the character themselves. The GameObject furthest from this character would match the position of the player's mouse without the distance from the game object that is positioned on the character being greater than the number of units I want the player to be able to hook onto units from.

The controls in my game will be the stereotypical keys for movement when a player is using a keyboard, WASD. The Interact key will be E, which while playing various games I have found to be the most common.

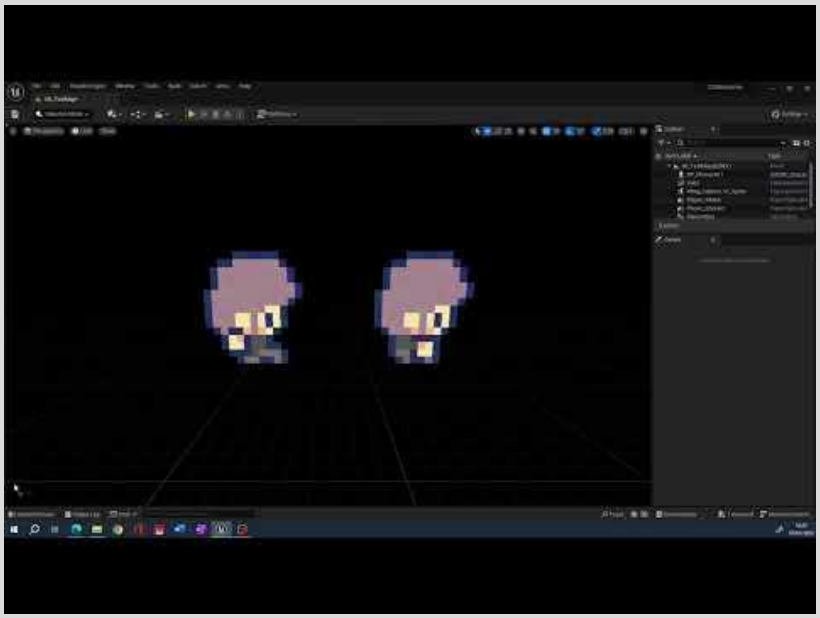
To play the game the player will not need a mouse as to attack they should press F on the keyboard. I think this should make it easier for players who do not always play with keyboard and mouse to play the game.



Implementation –

Youtube – Sprites

Youtube – Attack Sprites



Camera

Research-

Within 2d Platforming games the players character is most often close to the middle of the screen. In Super Mario Bros U…

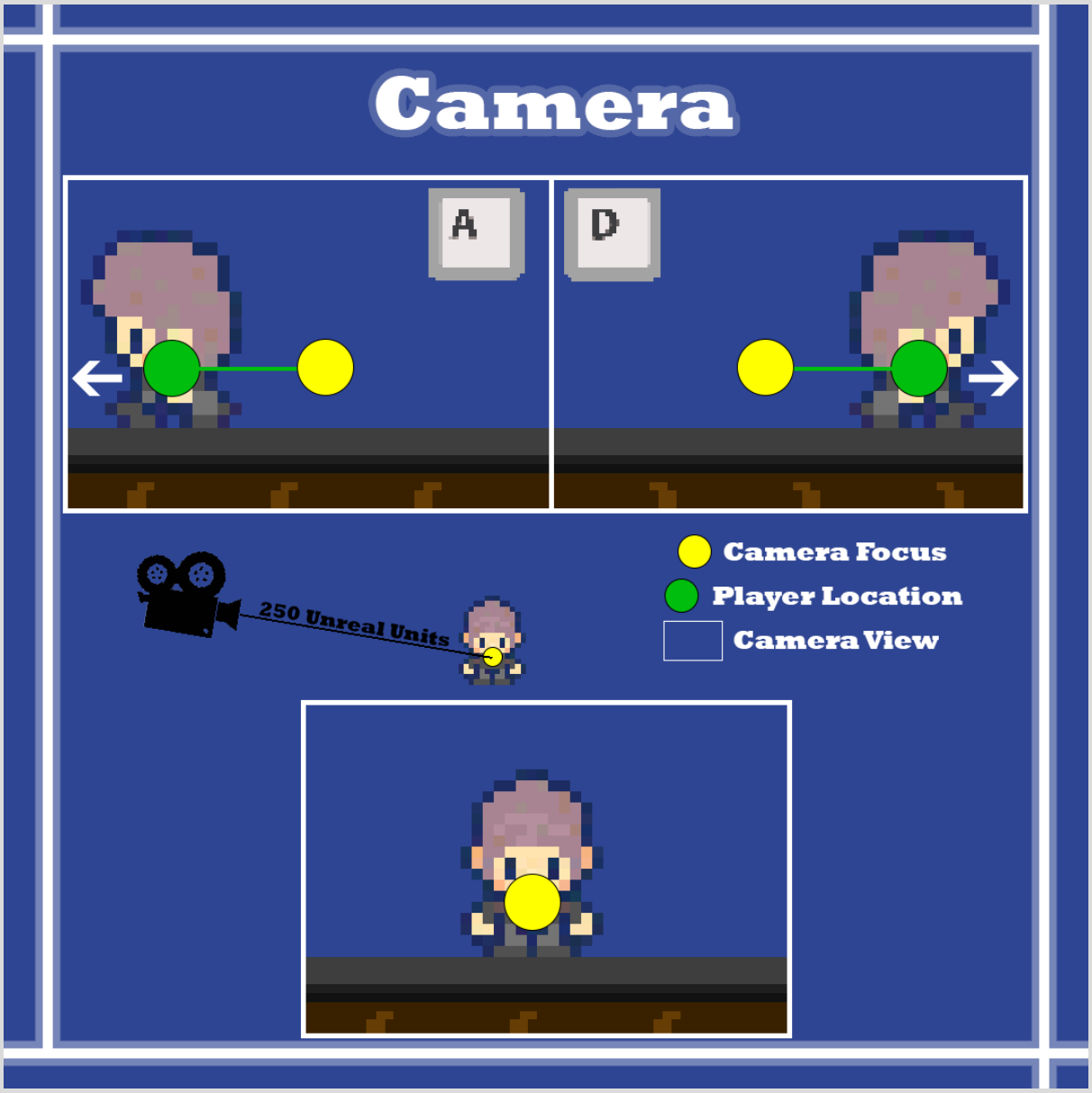
The Player is ever so slightly to the left, this lets the player look ahead of them. This is implying that to the players right might be where the danger will be come from. This allows players to see what they might be facing before they are fighting with the enemy.

As there are multiple different enemies in this game it also lets the player plan how they want to fight an enemy, the factors informing the players decision could be if the enemy can attack with range, how fast the enemy is, the enemies attack speed…Do they have time to try and avoid the enemies' attacks? Can they skip this enemy altogether?

The camera lets the player see their environment while keeping their character big enough for the player to understand what they are doing. This is something I am going to try and take into the game I make.

Production-

The Players camera will follow the player. When the player moves, the camera will follow them with a slight delay. That delay will increase the longer the player moves until It gets to roughly the players width away from them. If the player stops moving it will then return to focusing on the players location.



Opposition –

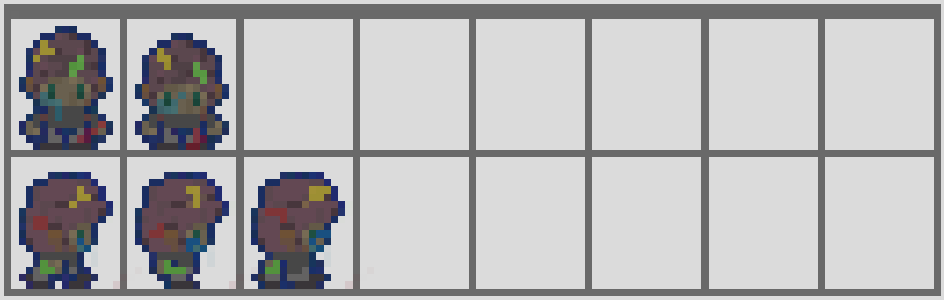
Research –

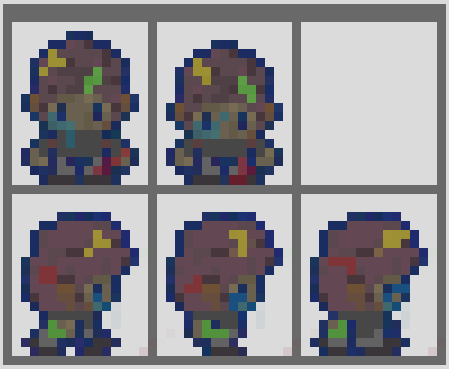
Broforce is a 2D platformer, the enemies in this game all follow a common theme, this helps the player identify them as enemies. It also would make the player feel as if they are fighting an organisation rather than a few enemies. This might make a player feel powerful.

Another common theme with in Broforce is that the enemies, all bar one has a mask over their faces, this means that the player can identify them easily. The one enemy that can fly so is still easily identified. This will also separate the player from the enemies adding to players feeling of power. This feeling could lead to the replay ability of the game.

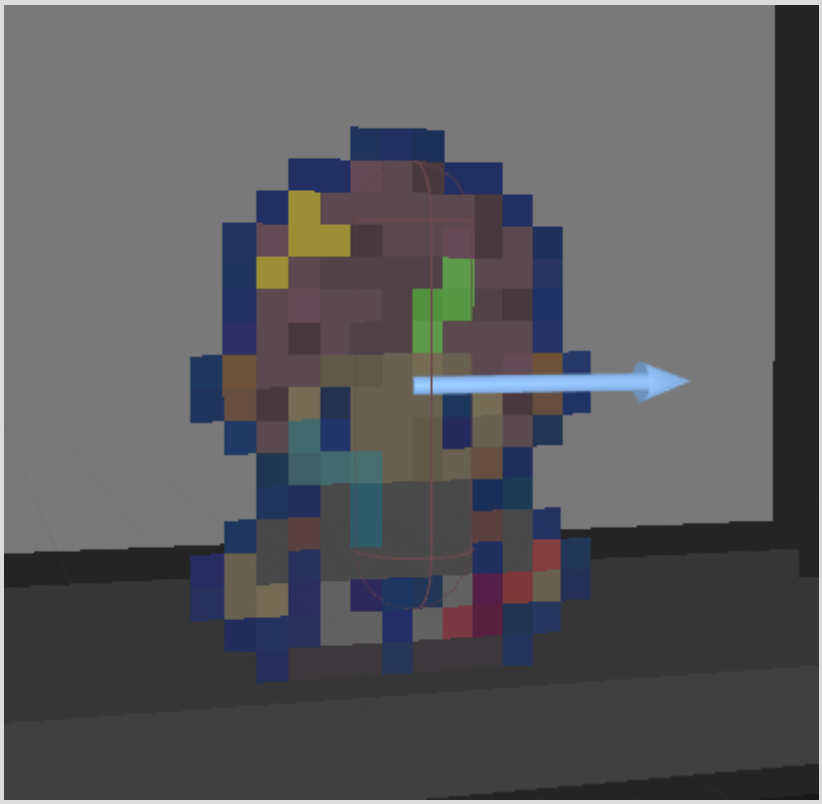
Production –

I used photoshop to create these sprite sheets, using the players character as a base, I then made everything darker and added splashes of colour. I intended to add sparks but ended up with this instead. I quite like it. The whole narrative behind my game is that the enemies are bugs. I think the splashes of colour create inconsistencies which I feel is a visualisation of a bug.





Implementation –



While I managed to get these in engine, I struggled to make them move back and forth. For this reason, they are stationary in the game, they will also not deal any damage to the player.

World

Research –

In the Nintendo game ‘Spelunker’ the player can interact with multiple different objects in the game’s world. As soon as the player starts a new save in the game, their character will be in a lift that they can move with the up and down keys. The player never actively plays as the lift however when the player is on it, they will control it.

In the image above the player is more than likely moving downwards as their character is looking that way. This means that when the player is pressing down to control the lift, the players state is being affected as well as the transform for the lift.

To get this working how it does the lift may have a collider over it. If I were creating this inside of Unreal engine. The lift and the character would be pawns, the player would “play” as a controller then when the player is in the lift the controller would possess the lift. Then when the player moves out of the lift, the controller would stop “possessing” the lift.

Spelunky also features ladders and rope these however cannot be controlled by the player. Instead, the player uses these to traverse around levels. These items would have a collider over them just like the lift. These objects however, if the game was made in unreal, would not need to be pawns as they do not need to be controlled. They would instead be actors which have colliders as their children. Then the functionality for them would check for the players collision and when it registers that, make the players animation one for climbing as well as allowing the player to move directly upwards and downwards.

There are also items in Spelunky that the player must move around. Stalagmites and Stalactites grow from the ground and hang from the ceiling. The player must jump around them.

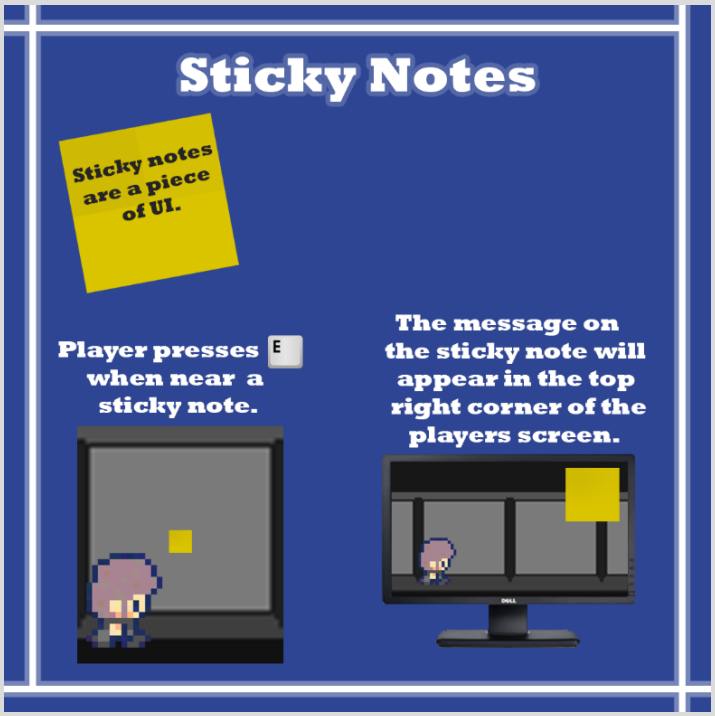
Mine cars, work the same as lifts except they move horizontally as appose too vertically.

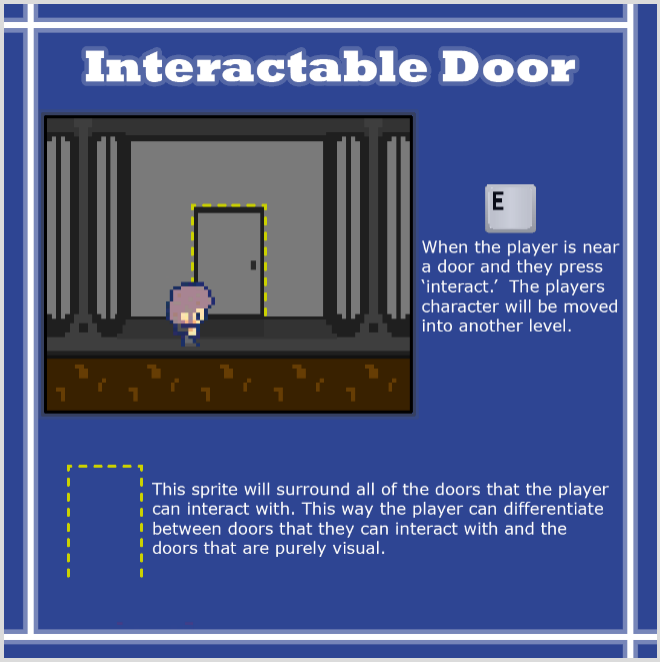
In the game I am making I am going to have things that the player must jump over. I also want to add something that the player should collect throughout their journey in the game. I could then make a blueprint that counts these items as they are collected. Then at the end of the game I could show the player how many they have collected.

Production –

Sticky Note –

Sticky notes will give the player a reason for moving through the level. (provide the story)

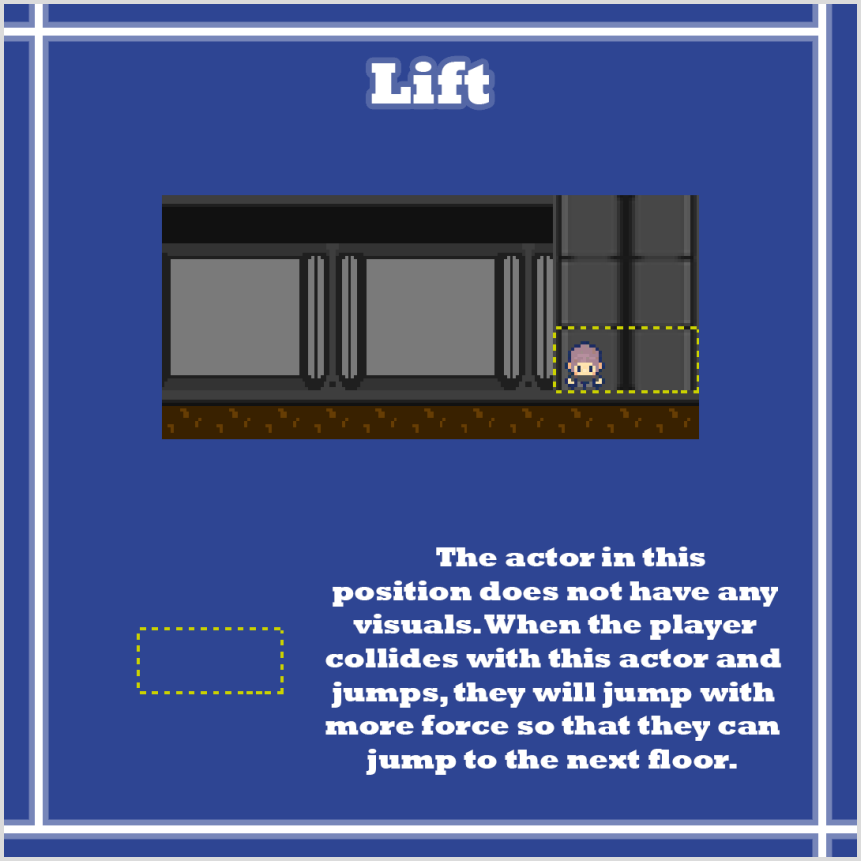




Interactable Door-

Once the player has interacted with a door they will be moved to the following level.

Lifts-



Collectables-

When the player is stood next to a collectable, and they press ‘e’ they will pick it up.