

AA4S10: Technology and
Engine
Cyberman
Challenge 5: Now Live Client
Brief
Personal Journal

Cole Underwood

TABLE OF CONTENTS

Module Introduction.....	4
Module Aims.....	4
Module Challenge.....	5
The Brief.....	5
Central game concept and progression.....	5
Playing format/platform.....	5
Game World / Visuals.....	6
Character / Player.....	6
Power-ups.....	6
Portrayal of threats.....	7
Music.....	7
Controls.....	7
Scoring / Shareability.....	7
Originality or inspired by (either is fine):.....	7
Group Assembly.....	8
Group Analysis.....	8
Group Coordination.....	8
Research.....	9
Game Engine.....	9
Overview.....	9
Unreal engine.....	9
Unity.....	11
Game maker studio 2.....	13
Classic retro video games.....	15
Overview.....	15
Asteroids.....	15
Pacman.....	16
Dizzy.....	17
Game Breakdown Analysis.....	18
Overview.....	18
Art style 16bit/32bit Colour.....	18
Game World.....	19
Xeodrifter.....	19

Bit.Trip Runner	23
Devolvement.....	25
Overview.....	25
Early stages of Devolvement	25
Early Character Design.....	27
Early Level Design	28
Development of Cyberman	30
Analysis.....	30
The beginning phase of Cyberman	30
Designs.....	32
Character Design	33
Enemy Design	34
Implementation.....	35
Character animations	35
Level Design	39
Screen Designs.....	44
Sound Design	47
Cyberman.....	48
Appendix A.....	49
Appendix B.....	51
Appendix c	55
Appendix D.....	58
Appendix E	59

MODULE INTRODUCTION

This module allows the student to explore the practical aspects of games production using existing engine software.

Responding to a series of focused briefs, students identify suitable game technologies to implement their design proposal. Evaluating the technical and practical constraints of the software.

An appreciation of the differences between competing technologies, their support networks, and the practicalities of the student's own abilities to use them should be evaluated.

During the module students will undertake a series of game challenges producing functional prototypes within engine, demonstrating key aspects of the game mechanic and/or aesthetic intent.

Module Aims

Individual: Produce a portfolio of work, evidencing the contribution towards each game jam (visual journal). This should include research undertaken (visual, design & process) as well as evidence of the work undertaken. A written critical evaluation (approx. 1500 words, 500 words per challenge) which evaluates the strengths and weaknesses of the challenge outputs, as well as the justification of actions, decisions and contributions taken in support of the group's game challenges.

Module Aims

- To complement and build on the student's existing skills as designers and makers of digital games
- To support and encourage the student's continuing exploration of game design, specifically the software 'engines' used to produce them.
- To enable the student to research and critically evaluate an ongoing process of games production, demonstrating relevant techniques and selection of tools
- To enable the student to develop their games production skills within a constrained brief.

Module Challenge

Live Client Brief

ThirdSpace would like to offer USW students a project to design an in-browser / online retro video game to complement and support the launch of a new service – ThirdSpace’s Cyber Threat Protection Service.

ThirdSpace (www.thirdspace.net) is an IT consultancy specialising in identity, security and cybersecurity solutions for large UK organisations and institutions. We would like to develop a simple, quick, and fun retro video game that can be used as a draw at our industry events (to attract potential clients to its stand). ThirdSpace also believes that the retro video game available from its website could also be a useful addition for social media sharing/interaction to draw people onto the website.

The Brief

Central game concept and progression

The simple premise of the game is that you are trying to protect your character (or an object) from cyber threats. These cyber-threats are doing one of the following:

- Trying to reach an objective that you need to protect.
- Trying to hit you.
- Trying to get past you (don’t let anything past!)
- All threats need to be caught within a time period.

The player will receive points for eliminating (could be destroying or catching) cyber threats (optional for more points to be scored if threats are destroyed quicker / or levels are completed quicker). We’d like our game to have levels/progression, whereby the number of threats to destroy/catch increase and/or the speed of threats increase. To keep the game quick and simple we believe your character/object should only have one life. The game ends when the first threat reaches the objective – end message “You protected your organisation with a score of XXXX” (don’t want to convey a negative message like “you’ve been hacked!”)

Playing format/platform

- Classic retro arcade video game style (16- or 32-bit style)
- Suitable / simple enough to play on Chrome/Edge/Firefox browsers – no downloads required, please

Game World / Visuals

The video game must incorporate our brand logo, our brand colours, and the use of the ThirdSpace orange brackets where possible/relevant.

As an organisation we like the idea of combining two distinct and different styles:

- A space theme incorporating stars, planets, moons, asteroids etc.
- Elements of business/security – office buildings, cityscape, computers, Cyberthreats

The reasoning behind space + business theme is due to 'space' / 'outer space' feeling like a good fit with a retro/arcade game while also feeling futuristic/modern. The 'space' theme also plays on the ThirdSpace name and has associations with cyberspace. We'd like to include references and associations with business/office/city in here to keep it relevant to our industry/services.

Brand guidelines are provided separately for all colours and assets.

Character / Player

Some suggestions for player characters.

- A security analyst
- An object like a ship / a computer / a network
- An organisation/building "MegaCorp" or "GigantaCorp" something like that

Power-ups

To make the game more interesting and to give players that much needed 'edge' to progress to the next level we'd like the player to be able to collect power-ups. These should be called/named after services and tools ThirdSpace offer that help organisations improve their security.

Power-up names (and potential characteristics):

- Sentinel / Azure Sentinel – slows time or increases character speed
- MTP – improves accuracy / or destroys all threats on screen
- ThirdSpace MSS (this grants invulnerability / a shield – please portray this with the ThirdSpace brackets protecting or forming a shield around the character/object)

We are happy to entertain other power-up ideas.

Portrayal of threats

To make this relevant to our threat protection/cybersecurity service the threats in this game should either be named or visually designed to represent commonly known threat types:

- a bug,
- a worm,
- a trojan horse,
- a robot (botnet),
- a mail icon (phishing),
- ransomware

Music

Music and sound effects will help bring the game to life. We'd like this to be classic /retro style sound effects and a real retro/cheesy background music.

Controls

Basic options: keyboard control utilising arrow keys, space bar and/or one option special attack/defence option button (shift or other)

More advanced options: Add mobile response controls if possible.

Scoring / Shareability

At its most basic level, we want to provide each player with a final score, and we'd like to encourage the player to share this score by screenshotting and sharing on social media.

- Great job on protecting your organization
- Your final score: XXXX
- Screenshot and share on social media to see how you compare #LetsDevelopACoolHashtag

We don't believe it's needed/or feasible at this point to run a database behind the game that would provide a leader board or the ability to enter a name. We are interested to know if this is possible within the scope of a project like this.

Originality or inspired by (either is fine):

ThirdSpace do not believe there are any issues/concerns with copyright on taking inspiration from the ideas and gameplay of classic retro arcade games such as Asteroids, Space invaders etc. If taking inspiration from these games we would ask that where possible we consider changes to sounds, graphics and some parts of gameplay, to ensure games have a visual/playable difference.

Group Assembly

Group Members: Cole Underwood, Sam-Alexander Ford, Tira Pace and Cavin Pimm

We took part in a Blackboard Collaborate meeting where we placed ourselves into groups to be able to research the potential game concepts. We used this as our main source of communication and to keep in contact regularly.

Cole Underwood – Developer, Programmer, Designer

Sam-Alexander Ford – Artist

Tira Pace – Artist

Cavin Pimm – Artist

Group Analysis

Strengths

- Each person from the team has skills and experience in different areas.
- We worked together prior (except Cavin who was new to the team)
- Organised and informed on the brief.
- We work strongly as a group and know each other on a personal level.

Weaknesses

- New member to get to know.
- Working on an engine I have very little experience in.
- I am not a strong programmer and my weakest area.

Resolve

I will work on researching and watching as many videos on GML programming language to try and get ahead of the game as I know this is where I struggle the most. I will get familiar with Cavin and have a discussion to see where his strengths and weaknesses lay.

Group Coordination

After assembling into groups, we created a discord channel that we use as a the main form of communication and created a shared folder on OneDrive to access uploaded documentation that each of us can edit.

RESEARCH

Game Engine

Overview

As we are developing a pixel style retro game, I will be exploring engines and researching on what tool would be best for the job. Although I am most familiar with Unreal Engine I will be breaking out of my comfort zone as I believe different tools are required for different jobs. Even though I explored a vast range of engines such as RPG Maker, Godot, Adventure Game Studio, Visionaire Studio plus many others, I will break it down to Three which I considered most practical and my thought process and which one I chose and why I chose that particular engine.

Unreal engine

Although I am most comfortable and familiar with this engine I feel it falls short for 2d pixel arcade games even though it can be done, the engine is built with powerful 3d in mind, while there are engines out there which are dedicated especially for 2d. As discussed on the lesson's live brief that "Dann Rees" built a 2d retro game using the engine, but the engine needs to be rolled back a few versions for the conversions to work. Unreal does have a pixel 2D engine port which did make me think of using this engine which can be found here - <https://www.unrealengine.com/marketplace/en-US/product/pixel-2d-complete-2d-engine-for-unreal>

But at the price of \$99.99 I could buy Game maker Studio 2 which is specialised in this area as a whole engine.

What are the advantages of Unreal Engine?

It is free to use

Unreal is free and does not require a cost, although anything extra such as plugins from the marketplace will cost money and can add up quickly.

It is Powerful.

Unreal Engine is a very powerful engine and is capable of 2D/3D.

Blueprints

Unreal Engine has visual scripting built into the engine called "Blueprints" which work just the same as code but for people who are not programmers a lot of tutorials and videos also use blueprinting. UE4's visual scripting system, Blueprint, is the main reason the engine is a great tool for prototyping and iteration. In Blueprint, you drag-and-drop nodes and add wires to connect them and add logic. This node-

based interface allows designers who don't have a technical background to explore their ideas without needing to know how to code.

Unreal Engine allows for quick prototyping and iteration

Right off the bat, Unreal lets you choose a template and your straight into the engine with something working and you're able to prototype and iteration from the get-go.

Unreal Engine's Marketplace

Unreal Engine has a large and flexibly marketplace where you can build upon code and make games using templates and piecing them together to create something of our own. It's a fast and constructive way to make something fast, especially in a game jam situation.

What are the disadvantages of Unreal Engine?

Not Ideal for smaller games

Unlike Gamemaker, Unreal is not as quick to make smaller projects, as it takes more time due to the power of the engine and a steeper learning curve with a lot more bells and whistles. UE4 comes with the background of being an engine for large AAA titles supporting tons of features for all kinds of advanced systems.

Engine is more dedicated to 3D

Unreal engine is more focused on 3D than 2D, unlike Gamemaker which is primarily built for 2D.

Unity

Although I am most comfortable and familiar with this engine I feel it falls short for 2d pixel arcade games even though it can be done, the engine is built with powerful 3d in mind, while there are engines out there which are dedicated especially for 2d. As discussed on the lesson's live brief that "Dann Rees" built a 2d retro game using the engine, but the engine needs to be rolled back a few versions for the conversions to work. Unreal does have a pixel 2D engine port which did make me think of using this engine which can be found here

What are the advantages of Unity?

Unity is fast and agile

Unity's ability to get things done very quickly is another valuable strength -- it allows for very fast iteration and can be extremely useful when you brainstorm for a new game concept. because it's a fast engine that you can make a game in quickly, without even coding or with very simple coding.

Unity visual scripting

Unity like Unreal has visual scripting called "Bolt" which is their version of C# of visual scripting. They also have a simplified version for easy implementation for simple mechanics called Playmaker but the downside it costs money from the Unity store.

Here is a quick table that shows the pros and cons of both programming languages, a link for information found here – <https://vionixstudio.com/2020/05/01/bolt-vs-playmaker-which-is-the-best-in-unity/>

Bolt	playmaker
Can do everything that can be done using C#.	Limited to pre-defined actions, Complex action requires coding
It is C# in visual form	It is a finite state machine
Has a steep learning curve	Easy to learn and implement
Best for complex game mechanics	Best for simple game mechanics
Suitable for large and complex codes	Not suitable for large behavior requirement
Best for people planning to switch to scripting at later stage	Best for beginners and people looking for pre-defined functions

Portability

Unity is also fast at porting, with your game being essentially ready to go on all the different platforms in one click. It's worth repeating here that Unity is available for 25 different platforms, which makes for a very valuable feature it is also free.

Unity has a large and varied asset store

well-stocked Asset Store for further tools, very well documented and supported with a great dev community all helping figure things out. Around 1.5 million developers visit the Asset Store each month to browse the 56,000 packages available -- that represents over one million individuals' assets available for creators to use.

What are the disadvantages of Unity?

Steep learning curve

From what I researched in articles and online forums; Unity seems to have a steep learning curve compared to Gamemaker as I only have two weeks to learn a new engine but also develop a game this might not be a good option compared to Gamemaker.

Code

Unlike Unreal Engine, where a lot of online tutorials use blueprints the majority of the tutorials on Unity would be focused more on C# code rather than using "Bolt" or "Playmaker" visuals scripting system. From a comparison from code between GML vs C#, C# has a steeper learning curve than GML.

Game maker studio 2

Why gamemaker? From researched gathered through forums and google searches game maker studio 2 seems to have the correct balance of accessibility and powerful. GameMaker's primary strength lies in making 2D games, and direct control of how sprites are rendered and plenty of tutorials and videos that heavily focus on games that use pixel art.

What are the advantages of Gamemaker?

GameMaker is arguably the best engine for 2D games

According to a Butterscotch Shenanigans CEO and co-founder Seth Coster. The co-creator of Crashlands and Level head, quoted "GameMaker is the best tool around for making 2D games of any kind -- that's what it was built for" there is a lot of resources arguably saying that Gamemaker is the best engine for 2D game development subjective but from anecdotal resourced data gather it seems to be very much the favourite. The source can be found here - <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-gamemaker-the-right-game-engine-for-you>

GameMaker is fast and accessible.

GameMaker is known to be very user friendly for beginners, for being incredibly powerful and fast for development, you can have a simple up game running quickly, while also iterating and prototyping with ease.

GameMaker is simple

GameMaker is also a simple engine to learn, using a plug-and-play system for making games with almost no programming using the drag and drop system.

GMS2's 2D toolset, takes care of the most common aspects you'll have to deal with -- objects, sprites, sounds, because it's not juggling both 2D and 3D tasks, it's lighter-weight with less bloat and a more focused interface.

GameMaker has its own language.

Although Gamemaker features a drag and drop system it has its own programme language also built into the engine, which is easy to learn according to many sources found in researching.

"GameMaker Language (GML) is the code language used by the engine, it is not as robust or object-oriented as common languages like C# or C++, "C#/C++ is more robust, but also 'heavier'. GML is fast and also forgiving.

GameMaker has a large and strong community

Although Unreal and Unity also have a massive community from my research from looking at their forum and third party forums unity has a very strong community which has links to discord channels with Dev youtuber's on the page helping people with problems. I feel from personal experience and research it has a stronger/tighter community-based system.

What are the disadvantages of Gamemaker?

GameMaker is not free

Unlike Unreal and Unity, Gamemaker you must purchase a licence to use its full functions in the engine, it does have a trial period but that has limitations. It also cost quite a lot of money although they do have sales quite often plus a monthly subscription, you can view prices here -

<https://www.yoyogames.com/get>

GameMaker's ecosystem is small

GameMaker's status as an engine for beginners is its limited asset store -- especially when compares to Unity or Unreal Engine 4, GameMaker's asset ecosystem is smaller than that of Unity, and Unreal Engine

GameMaker's export to other Platforms

With Gamemaker, they have different publishers' packages each with separate prices where you have to pay extra on top of purchasing the engine which can cost a lot when you have all that free with the other two engines.

GameMaker has its own language.

Even though this was listed as a pro it also has a downside, Gamemaker can provide bad habits to programmers due to its flexible/forgiving language. Although it is powerful it does not compete with the power of C#/C++.

Classic retro video games

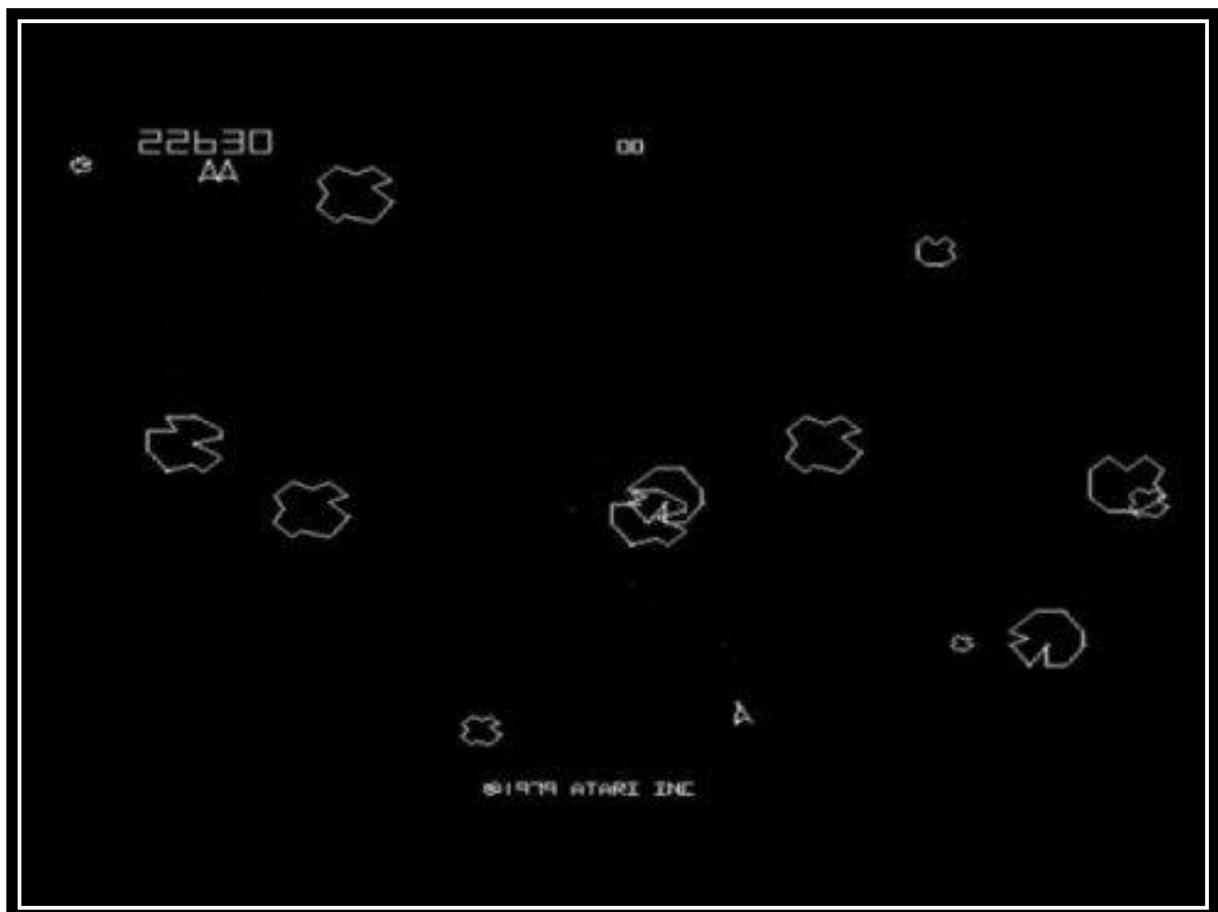
Overview

I will be researching through different sources of retro video games and look into the designs, sounds and artwork and delving deeper into a selection of different mechanics and what can be taken away and influenced into our project.

Asteroids

The overall premises of Asteroids are that the player controls a single spaceship in an asteroid field which is periodically traversed by flying saucers. The object of the game is to shoot and destroy the asteroids and saucers, while not colliding with either, or being hit by the saucers' counter-fire. The game becomes harder as the number of asteroids increases

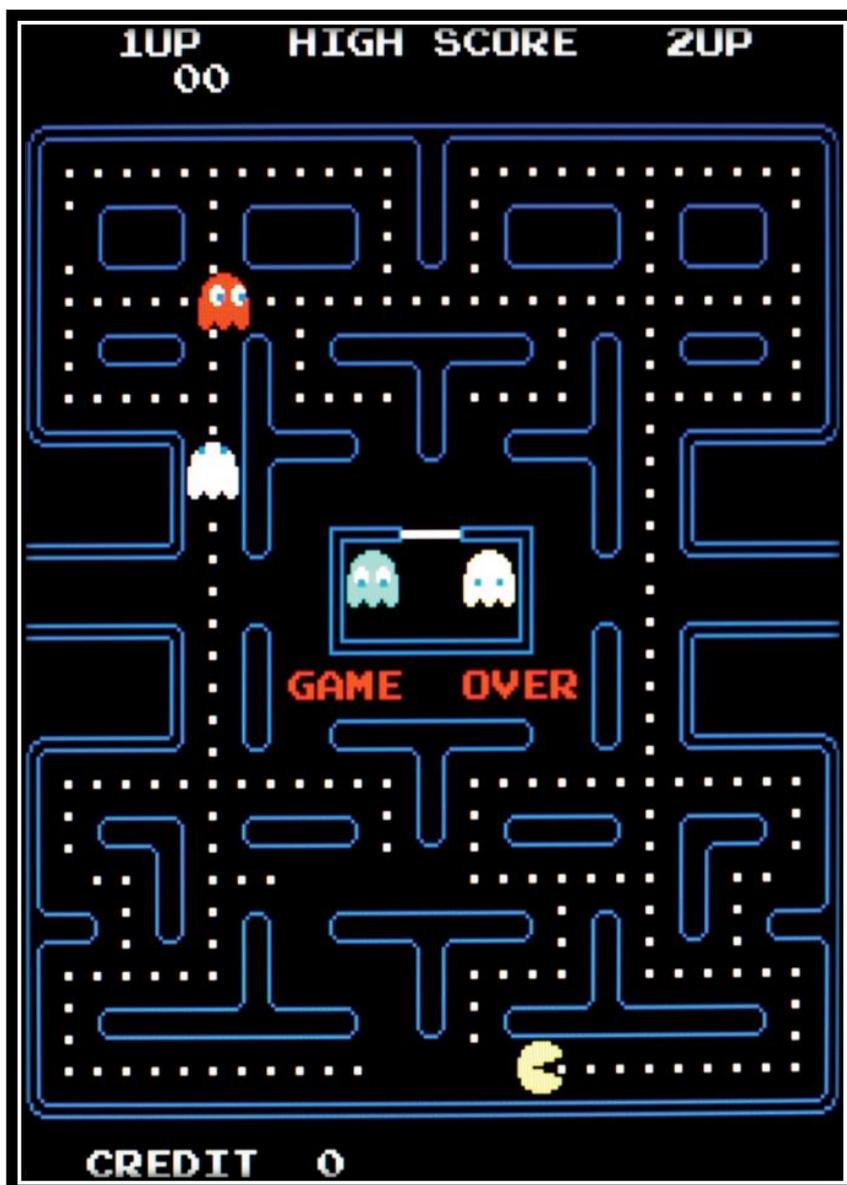
Asteroids use a lot of the same underlining mechanics the client wants in a video game. This is what Dann Rees built using Asteroids as the influences as we discussed with the group, we wanted to take a different method and use a completely different concept completely but taking elements of what asteroids uses and research deeper.



Pacman

Pac-Man is a maze chase video game; the player controls the eponymous character through an enclosed maze. The objective of the game is to eat all the dots placed in the maze while avoiding four coloured ghosts. the player advances to the next level. If Pac-Man makes a collision with a ghost, the player will lose a life; the game ends when all lives are lost. Pacman also has powerups where the player can eat the ghost which increases the scoring value, the game increases in difficulty as the player progresses further by surviving longer.

Pacman like asteroids also covers the basis of mechanics the brief wanted but is a completely different game, the mechanics differ slightly where Pacman is about surviving by running away, while asteroids are about attacking the enemy to survive. Both have the same premises of surviving which increases the score and progressively gets harder.



Dizzy

Dizzy is primarily known as a platform adventure game which includes many running and jumping to avoid enemies and hazards and relies heavily on objectives and inventory-based puzzles to interact with the environment and non-player characters.

Although Dizzy is not really what the brief mechanics are after, Dizzy has something important like other video games such as “Mario” and “Sonic the Hedgehog” which is a strong protagonist character, as the brief mentions having a security analyst as a playable character using elements which these games portray would help set the bar.

Through the first part of developing our earlier stages was inspired by that of dizzy, Mario and other platform games but as discussed the mechanics did not fit well with the brief so we looking into running games using a blend of platformer and keeping the initial idea but relevant with the scoring system.



Game Breakdown Analysis

Overview

I will also be breaking down a selection of games which was influenced in the final decision and the thought process that went into making our game.

In my research, I'll be looking into retro-style games in the style of pixel art that use elements related to the brief, such as –

- High scoring system
- Protecting the character/object
- Powerups
- Enemies/threats

I'll be looking into games that use these mechanics and delve into deep analysis from different genres to take elements from many different sources to come up with something interesting which is relevant to what the client is looking for. All the games research all have limited simple controls and retro-style music which will be implemented into the creation of our game.

Art style 16bit/32bit Colour

16-bit colour

With 16-bit colour, also called High colour, computers and monitors can display as many as 65,536 colours, which is adequate for most uses. However, graphic-intensive video games and higher resolution video can benefit from and take advantage of the higher colour depths.

32-bit colour

Like 24-bit colour, 32-bit colour supports 16,777,215 colours but has an alpha channel it can create more convincing gradients, shadows, and transparencies. With the alpha channel, 32-bit colour supports 4,294,967,296 colour combinations.

As you increase the support for more colours, more memory is required. However, almost all computers today include video cards with enough memory to support 32-bit colours at most resolutions. Older computer and video cards may only be able to support up to 16-bit colour.

Analysis

Most users cannot tell much of a difference between 16-bit and 32-bit. However, if you are using a program that has gradients, shadows, transparency, or other visual effects that require a wide range of colours you may notice a difference. While a 16-bit processor can simulate 32-bit arithmetic using double-precision operands, 32-bit processors are much more efficient.

This researched was important to us and the art team to find out if the colours we are using or art style must be changed or clarified in any way but the overall analysis in simple terms suggests the user would not be able to tell the difference and its mostly about the amount of memory pro

Game World

Thirdspace are wanting a game set in either a space theme or office/business theme. Exploring games that have these game worlds will be important for different reasons by the type of game and mechanics used that fit in the world.

Xeodrifter

Xeodrifters is a modern take on pixel style arcade games, the world setting is using a space-themed world. Xeodrifter is a Metroidvania video game, where the player controls an astronaut whose spaceship is damaged by an asteroid and must-visit four nearby planets to repair it. The game allows the player to explore the planets, each representing a different environment. The planets can be visited in any order, but exploration depends on power-ups to gain access to a new area.



Gameplay

The gameplay of Xeodrifiers involves the mechanics of shooting and enemy combat. The enemies are alien type creatures such as bugs and sci-fi organic lifeforms. These lifeforms are a great inspiration for enemy character design

Enemies

The enemies are organic and are based on bugs/plant lifeforms each with different mechanics such as flying or shooting.



Controls

Xeodrifiers use more controls than the old limited control set retro games used that is due to all the added mechanics used then it was back then. I think limiting the control system and making it simple is a string approach to the brief.

Controls

PS4/Vita	3DS/Wii U	Action
		Move character/cursor
		Pause, access menu screen
		Jump
		Fire weapon in direction facing
		Plane shift (power-up required)
		Dock to planet, perform action
		Phaze (power-up required)

Xeodrifiers uses a lot of different power-ups which is relevant to the brief but as you can see this adds a lot to the control system, I think having researched into this, having powerups that you collect and automatically activating on collision would keep the balance of simple controls and more complex mechanics used in powerups.

Power-Ups			
Name	Description	How to Perform	Found
 Submarine	Allows you to dive underwater. You can shoot in four directions while underwater.	  (in water)	Planet 1 (defeat Boss #1)
 Plane Shift	Allows you to shift in and out of the background where appropriate.	 	Planet 2 (defeat Boss #2)
 Run	Allows you to run across pools of lava unscathed.	Forward +  	Planet 1 (defeat Boss #3)
 Rocket	Allows you to fly straight upward in the air.	Up +  	Planet 3 (defeat Boss #4)
 Solar Flare	Allows you to charge a powerful shot to clear dark blocks found in certain areas.	Hold   to charge (if activated on menu)	Planet 2 (defeat Boss #5)
 Phaze	Allows you to warp a short distance, including through walls and blocks.	 	Planet 4 (defeat Boss #6)

The World

Xeodrifiers art style is also set using limited colours throughout the world and characters. It uses 3 layered images of a foreground, middle ground and background.



Sound

Xeodrifiers uses retro-style music and sound fx using a style known as “Chiptunes” is a style of synthesized electronic music made using the programmable sound generator sound chips or synthesizers in vintage arcade machines, computers and video game consoles. 8-bit music is what the original retro arcade-style games used. Rather than being created by traditional instruments being recorded and imported as loops into the game, they were synthesised using the computer chips themselves.

Examples of Chiptune - <https://www.youtube.com/watch?v=kJgdw1Lhzg4>

8-bit - <https://www.youtube.com/watch?v=oMgQJecVToY&list=PLzjkiYUjXuevVG0fTOX4GCTzbU0ooHQ-O>

Xeodrifiers music - <https://www.youtube.com/watch?v=YeOwUvNCF9c>

This is a perfect example of what type of style we will be using in the development stage using music and sound fx of Chiptunes/b-bit style of music.

Bit.Trip Runner

Bit.Trip Runner is a 2D runner platformer in which the player must input certain commands on the controller, including jumping, sliding and kicking, to avoid or destroy obstacles and enemies. If the player is hit by any object, he is immediately warped back to the beginning of the level and starts running again. During levels, the player obtains power-ups that upgrade the score multiplier which increases the players score throughout the level.

Later through the development, this was suggested by the tutor through a discussion in blackboard in the initial stages of earlier devolvement. I purchased the game on steam so I could playtest it myself, as watching on YouTube you don't have the feel of the controls and missing that palatable experience. This was heavily inspired upon creation using elements which I play tested back and forth using this game to set speed and movement controls.

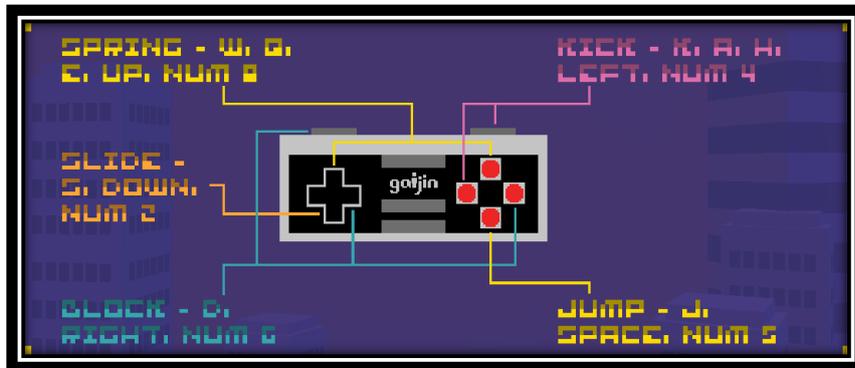


Gameplay

Bit.Trip Runner is an endless- running game where the player auto-runs and you must jump, duck or kick to avoid or destroy enemies. The game has over 50 levels and progressively gets harder as you progress., There are powerups which help you progress further and add to your score. The overall idea to get the highest score is to collect all goal bars in the level which unlocks a bonus level but the gold bars are placed in such a way that it is risky and the player dies which takes you straight back to the start of the level.

Controls

The controls are very simple using 5 control commands in the whole game, Having this simplicity is something which I think should be implemented in devolvement sticking to the old school style of retro games.



Powerups

During levels, the player obtains power-ups that upgrade the score multiplier, starting at 'Hyper' and progressing through 'Mega', 'Super', 'Ultra', and finally 'Extra'. As these multipliers are gained, the background music becomes more advanced accordingly, with new melodies being added and modern instrumentation layered on top.

Having items such as powerups to increase/or help increase score would be a cool idea to take from this research and add to the devolvement stages.

The World

Bit.trip running although is not set in a space type world or office space would not be relevant but setting up the scene in Gamedev using parallax backgrounds that Bit.trip running has and the dimension layout of the level design is elements to take into account in the devolvement process. The game has multiple levels through the development stages we decided to stick to a single level (more detail will be explained in the development section).

Sound

The game's soundtrack combines 8-bit music with contemporary genres such as techno and electronica. The independent chiptune band Anamanaguchi contributed some songs to the soundtrack.

The soundtrack here - <https://www.youtube.com/watch?v=BQ9Z5jxTVI&list=PL362A5DFE79840DDB>

The sound fx also play a huge part in the powerups as its not only an endless running but is a rhythmic game by changing the sound score by pickups.

The overall idea of the sounds used in this game was a huge influence through our development as it uses retro but adds a modern twist to the overall sound design.

DEVOLVEMENT

Overview

In this document section, I will be discussing the devolvement stages and the process of how the game was created. I will be breaking sections down and writing a deep analysis of each section and explaining my thought process through design and technical implementation.

Early stages of Devolvement

Abstract

I used this time to learn the engine while doing my research, the early research we were looking primary at developing a platformer. From previously mentioned in my research analysis I came to a split decision of whether to use drag and drop from the start or jump into the deep end and use game maker language which I'll refer to as GML. Gathering research from forums and other sources, it was apparent that the majority recommended using GML as most tutorials and communities use this.

I started from a tutorial series for a few days while actively researching as I have had limited experience using this engine since Gamemaker 1 back over 3 years ago in my "Game art and design" course. The tutorial helped me understand the UI and the basic foundations of GML from a very basic level. I will break down the engine into detail and explain the stages of devolvement using this engine.

Link for the tutorial can be found here - <https://www.youtube.com/watch?v=izNXbMdu348&t=62s>

Introduction of Gamemaker

Like most engines, gamemaker uses a bunch of different resources that combine to make a fully functioning game. Gamemaker primary functions are spires (The art in the game). Objects (objects are a special resource that we use to control aspects of a game and to do specific things) and rooms (The level/levels).

As I have not used ganemaker since over 3 years the first thing I noticed is gamemaker studio 2 has a different interface prior its predecessor, in theory, I needed to learn the whole engine again not just because of the time away from using it but also the whole interface.

I will break down in sections the resources gamemaker provides and break down my understand in the journey and how I learned from scratch to developing a game in a short space of time.

Sprites

What is a sprite? A sprite is an image and artwork in your game. You can set the size of the spires within the engine and edit sprites with a great deal of customisation. Gamemaker 2 has a built-in image editor where you can paint and draw but generally, you import sprites into the engine from other artists working in groups especially.

Spites animations are created by using frames very much like a paper flipbook where you create an animation but a set of drawings and flipping through the pages quickly, gamemaker works very similarly to this.

Objects

An object is something that is interactable in the game by using an instance, an instance is something which happens by the code either GML/drag and drops. Think of an object to that of unreal Actors, Pawns, Characters and Controllers but all rolled into one.

Rooms

Rooms are essentially your level/game world, rooms are the viewing port in which the player can see what is happening on the screen this includes not just levels but splash screens, menus which are divided into different layers.

Logic

The logic gamemaker uses are its programming language, which is created in two different formats, the programming language is known as "Game maker language" or scripted visually referred to as "Drag and drop". The language works by creating an object which is made by using events and actions, these events and actions are created using the login within the engine by either GML/drag and drop.

Prototyping

Using the building blocks in the engine that was previously discussed using sprites, objects, rooms and logic I created a prototype in the early stages of devolvement using a tutorial series to familiarise myself using the fundamentals learnt. I created a platformer with basic movement mechanics and a room with a collision as the initial ideas was a platformer so I thought it would be a great way to pick up the fundamentals and something that could be applied in some way using the tutorial.



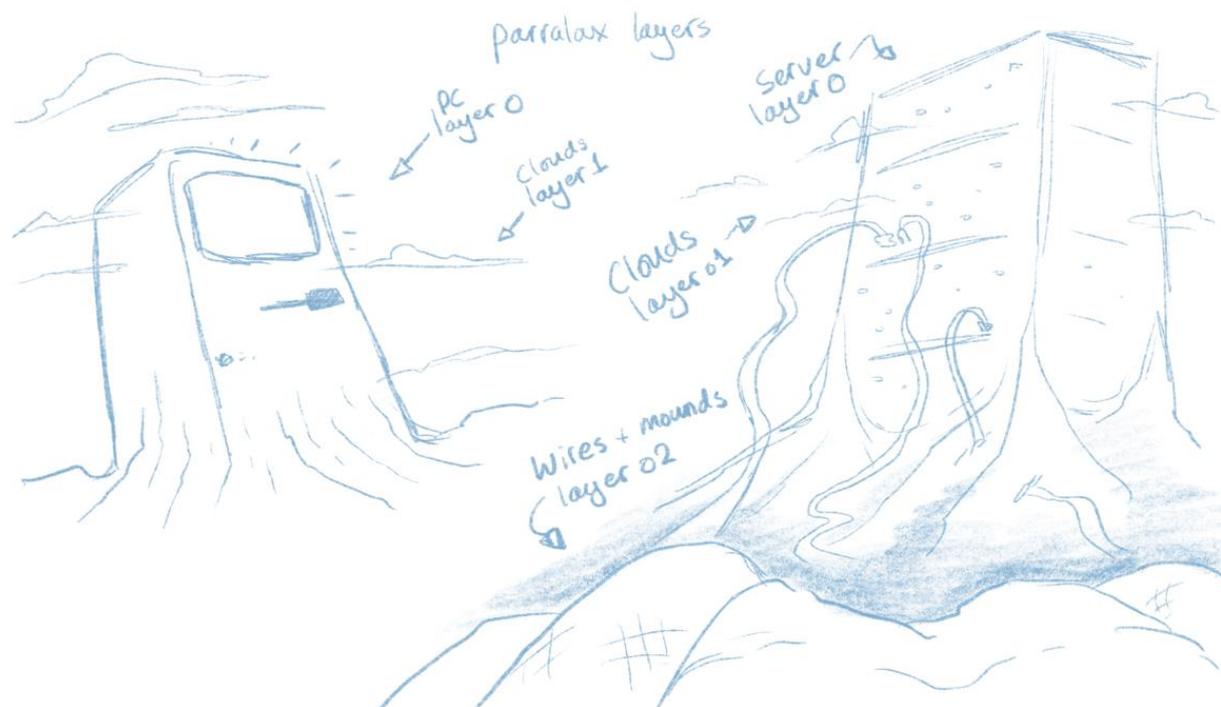
Early Character Design

Sam drew up some early character designs for a platformer, a security analyst who attacks enemies with a power sword which absorbs powerups and slices the enemy bugs. The scoring system would be how many bugs killed in each level. The character would go through the level of absorbing powerups like shields and weapon powerups.



Early Level Design

The early ideas of the level design were to have multi-floor levels where the protagonist would summon from an elevator each level and at the end of the level would take the elevator to the next floor. We were going to have a type of planet where it has huge giant servers and wires like the planet were organic but with office machinery.



Evaluating

After gathering more research and idea generation within the group, we decided that a platformer with multiple levels would not be a great idea as the scoring system would be limited to a set value and we would need multiple levels for this to work. We then came up with the idea of a single level that was endless that way it makes sense with the scoring system and allows the level to progress in difficulty. It seemed this approach had the correct balance of difficulty and relevance to the brief.

Development of Cyberman

Analysis

We gathered research and delved further as a group and thought having an endless runner would be a great idea as it fits the brief with the scoring system, the further the player runs the higher the score, not only that but it was easier for us to develop having only a single level to focus on. We gather the research for game influences which is stated in the “research” area of this documentation, looking at games such as “Bit.Trip Runner” and “Canabalt”. I downloaded both games so I could playtest them and get design ideas and influences from both titles. Once we had an idea we could develop further as a group and work on different roles of production.

Break down of team roles within production –

- Cole Underwood – I designed the overall premises of the game, how the game functions, implementation, working in the engine, programming, game design aspects.
- Tira Pace – Tira worked on the powerup art designs and some enemy creatures which I then animated in the engine and implemented into the game.
- Cavan Pimm – Cavan worked on the art of the overall level and I worked with Cavan how to design the level and implemented the design within the engine.
- Sam Alexander-Ford – Sam worked on the main character design and creatures and the menus/splash screens I worked with Sam and implemented his designs.

I will break down each part of the design and each part of our roles throughout the documentation and do a step by step process of the design of cyberman.

The beginning phase of Cyberman

The first process of design I researched into running games and took concepts and ideas to come up with the design of how the game will work, art styles, sound etc. I created a one-page GDD of the ideas generated in a document.

Copy of GDD

Researching development

I researched into the development of how I was going to build an endless runner as my programmer is not a strong skill-set I needed tutorials and programming examples to get it up running and working. Thankful I found a tutorial series on YouTube on "How to make an endless runner" this was great as I could build the overall fundamentals using the tutorial as a template to build from.

[Cyber]Man

One-page design document.

Game Identity / Mantra:

A Stylized action platformer about a security analyst with an old-style computer monitor for a head, avoiding bug and being hacked.

Design Pillars:

Fun, Action-packed Bug- Mayhem

Genre/Story/Mechanics Summary:

The game is an endless runner where the player must avoid the bugs and collect powerups to help them get further in the level. The idea is for the player to get the highest score possibly by dodging enemies by rolling, jumping over enemies.

Features:

Infinite Runner
Scoring System
Powerups

Controls:

- Keyboard "UP" to jump "Down" to roll

Art Style:

Pixelated retro art.

Music/Sound:

Retro music and sound effects.

Tutorial found here - <https://www.youtube.com/watch?v=UB7QKBtKtp0>

After getting so far, I had issues with the programming code as a lot changed as this was an earlier version So I needed to read through online forums and use problem-solving to figure out what the issues were and how to fix them. I managed to get to a certain point in the videos but got completely stuck and at a dead-end, this was very frustrating as I felt I hit a wall and no way around to fix it. I then decided to think outside the box and look into other tutorials or some other way to build the concept.

After searching I resorted to looking at gamemakers marketplace and to find something of relevance. I found a template to the endless runner I could use which worked! This was great as I could now focus on implemented the artwork and progressing the game further into development.

Link of template - <https://marketplace.yoyogames.com/assets/9223/endless-runner-engine>

Designs

These are the designs and process of the start of the project based on the new design of an endless runner. With the team now having documentation and ideas talked about also on discord we had a plan of what we were building. Sam drew up sketches of the idea of the character and enemies that will be featured in the game.

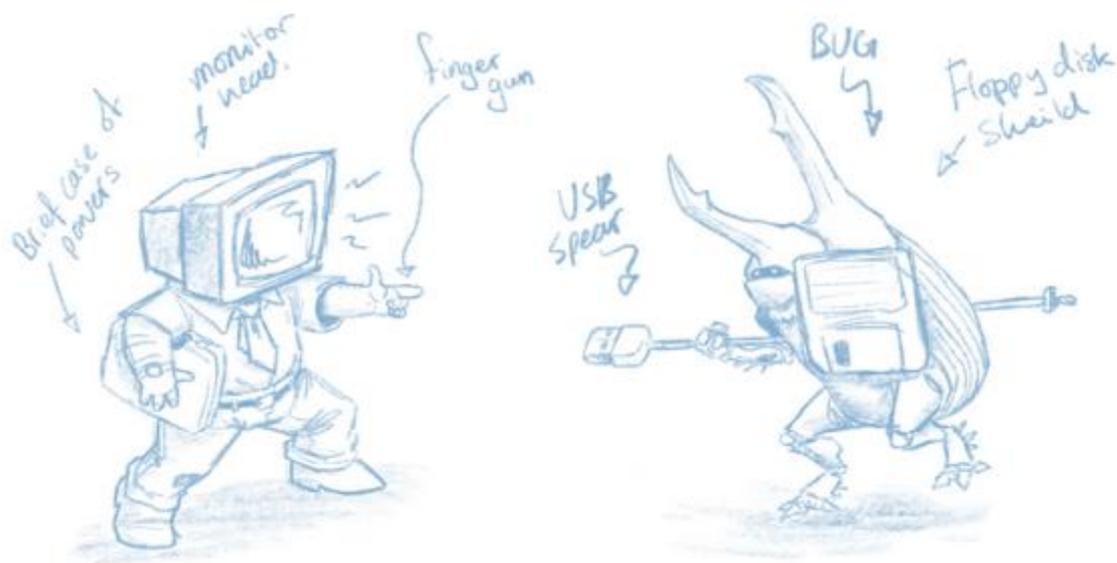
Note: Through the design process the team will keep in mind that we will be referring to the design brief in the appendix index

- Appendix A
- Appendix B
- Appendix C
- Appendix D
- Appendix E

Character Design

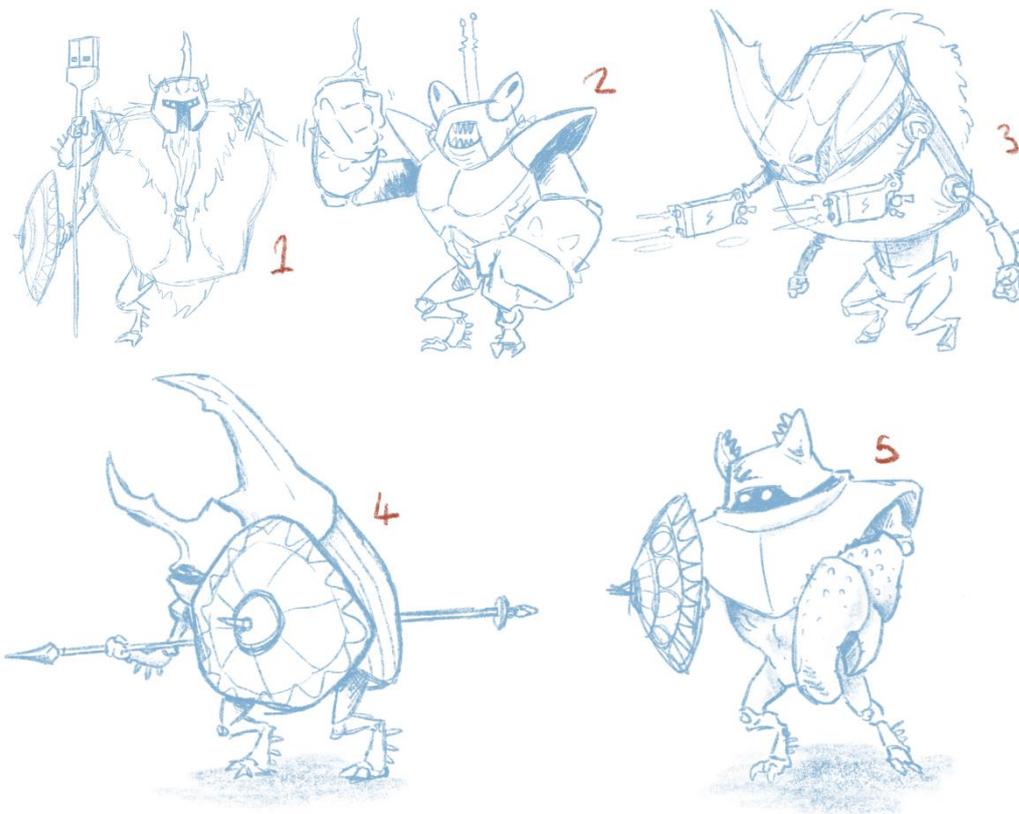
Sam drew up ideas for the main protagonist of using a monitor for a head where the player can essentially get hack and you must avoid the bugs and collecting powerups would stop the bugs hacking you. Sam was thinking a run and gunner like a game called “Cuphead” but with my limitations on coding, I thought it was best to avoid that and working on keeping a tight game by keeping it as simplistic as possible by the resources gathered. The character enemy I and the group liked and having a USB stick as a spear we thought was really fitting.

The character Sam suggested was based on Mario having the small stumpy frame as he suggested “It would be easier to animate” which I completely agree with as longer limbs would more likely be tricky.



Enemy Design

Sam designed enemies for the game using the briefs designs outliner, these enemies are bug related software hacks such as worms and trojans. He posted the images on discord where we picked the ones that stood out and to take into a pixel format.



Implementation

Now I had a template to work off, I could work on implementing designs further and importation assets and sounds. The first thing I needed to do was important the character and get that working in the engine and workout with the group how we are getting the animations to work.

Character animations

The first thing I needed to do was important the character and get that working in the engine and workout with the group how we are getting the animations to work. From my previous prototype, I used a reference sprite and used image strips by separate frames and imported them into the engine where I cut a single strip into frames.

Here are some early examples Sam created -

<https://youtu.be/OHU1nUylZS8>

I worked with Sam on the character animations as he was struggling in areas and suggested to use character animation templates for example -



That way he has a form of an animation cycle and could paint his character over the cycle for the animations to work. Sam got it working using the method I suggested, and I then needed to have each animation frame by frame and implement into the engine.

Here are the frame strips Sam created for Cyberhead, from the template we had a un, jump, duck and I wanted to add a death animation. Once the strips where created I split them into the engine and play tested them by setting the correct speed in which the frames animate.

Character Run cycle



Character Rolling Animation Cycle



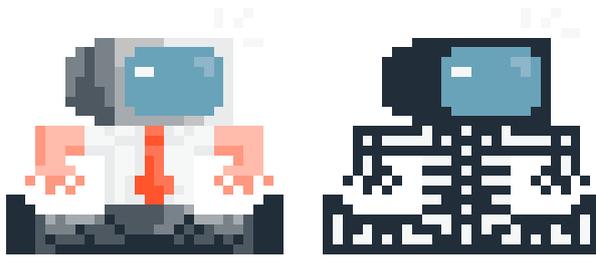
Character Jumping Animation

With the jumping animation, only one frame was needed.



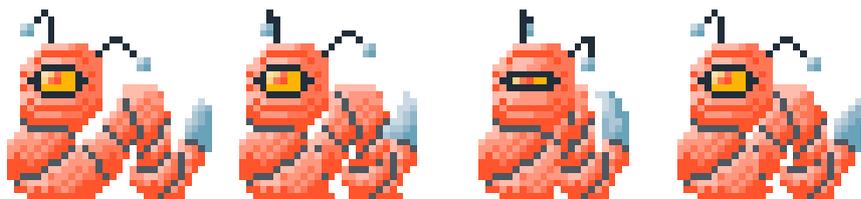
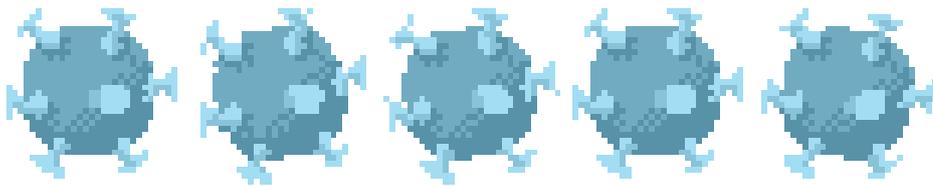
Character Death

The characters death was split into two frames that loop to get an electric shock image (Note this did not work in presentation due to error with the animation I needed to fix)



Enemy characters

The enemy characters were examples set to the brief using Bugs, Worms and Trojans as characters. I will put animation strips below to show the examples implemented into the engine.



Powerup/Power down Animations

Tira created on the powerups, as there was only two implemented we worked on a power-up that slowed the player down making it easier for the player and a debuff called "Phishing" as in Phishing attacks we added a fish animation to add some quirkiness.

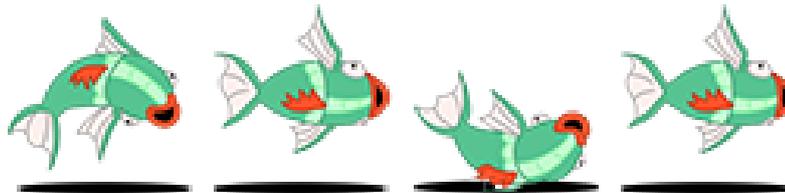
Azure Sentinel

This powerup slows down the enemy making it easier to travel further to accumulate points.



Phish

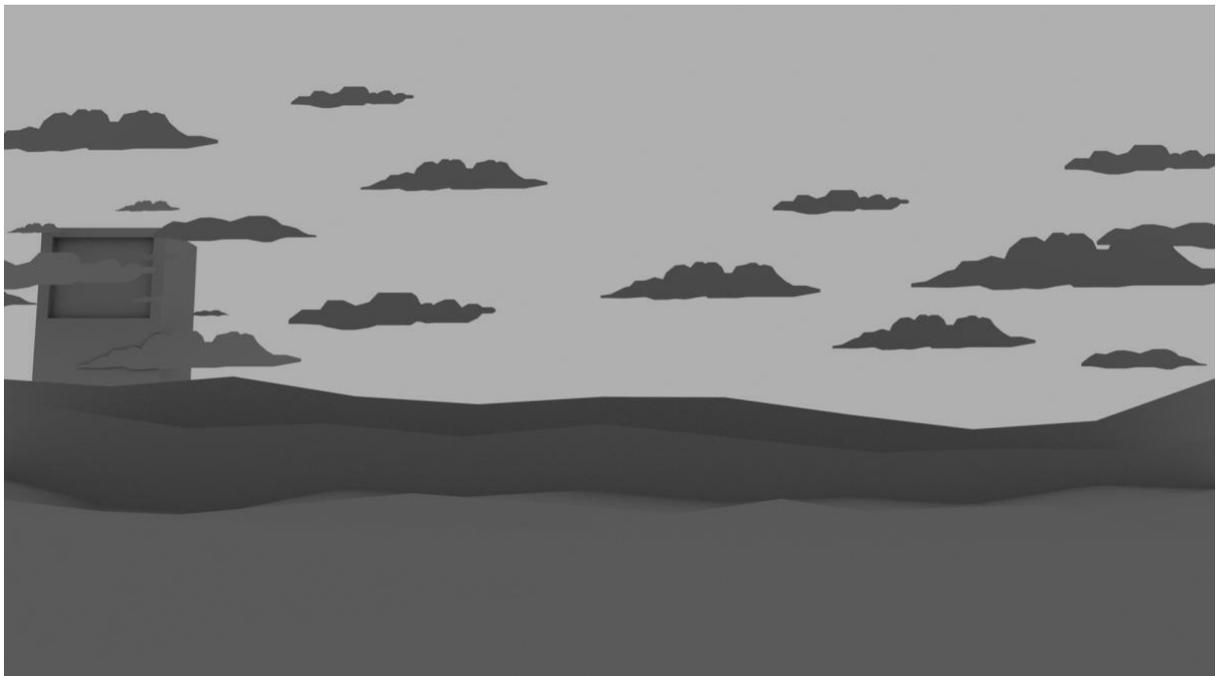
This inverts the player's controls, making up down and down up.



Level Design

Cavan worked on the art for level for the game I then implemented the level and made it work inside the engine. The level is a single image in the background and block flooring in the foreground that is tiled. The image in the background keeps scroll and the blocks. The background image is a single that is mirrored, if not mirrored there will be overlap and there will be a split in the image. The background is split into two separate layers to create a parallax background, this effect adds depth to the background images and makes them feel interactive.

This is the background used as you can see this had lots of problems such as the image not being mirrored which cause a split in the image scrolling, also the clouds where static which did not give the overall effect any depth in the level.



Level Background

We created the level into separate layers to create a parallax effect from previous feedback from our tutors which made sense to add depth to the level, so I worked with Cavan to implement that effect. I suggested splitting the details into layers and then implementing the layers at different scrolling speeds. With the level art, we kept to the brief see

- Colour theme – Appendix A
- Logo design on the buildings – Appendix B
- Typography – Appendix C
- Brackets – Appendix D

Background 1



Background 2



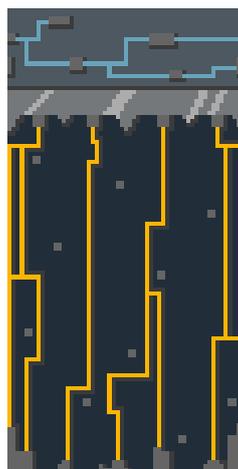
This is the image when combined from background 1 and background 2, in-game they both move separately at different speeds which creates a parallax effect.



Cavan created clouds which I added to another layer adding more dimension to the level. The first cloud needed transparency as the enemy could be hidden which could kill the player as they could not see behind the cloud.



The flooring was created using a single block that was layered in the engines room editor. The floor scrolls but also spawns from the start which gives it a continues loop. Cavan used circuit boards for the floor design.



We added another dimension to the background but after playing and feedback from the tutor that it was distracting, and we decided to remove it from the final version.



This is an image with the layer added when looking you can see the colours blend too much with the floor due to the limited colour palette the artist could work with. We agreed it looked better without and removed it from the final version of the game.



Screen Designs

Title Screen

To create the title screen, I thought looking in the marketplace of gamemaker would be a quick solution than making it from scratch. I found a free version which is noted in the description

Important Note: You may need to set up the audio groups (music and sound_effects) and assign the sounds in those folders to those audio groups to avoid an error.

This would not work unless I fixed the issues addressed, I implemented the template and fix the audio groups and managed to get everything up and running.

This Asset Contains:

- Aspect Ratio Setter
- Logo Screen
- Main Menu
- Options Menu
- Pause Menu
- Sound Control
- INI Save System

Link to the source - <https://marketplace.yoyogames.com/assets/6280/game-shell>



Title Screen for Cyberman

This was the first screen created by Sam which looked great but once implemented in the engine the text was barely visible due to the background colour and did not work well as a menu colour even when I change the colour text. Being as it was a title we spoke to Lewis Cobern (ThirdSpace Marketing Manager) and he said to feel free with detail and colour scheme with the titles menus which gave Sam flexibility. Title screen 2 we used black which worked far better and easier on the eyes.



Title Screen 2



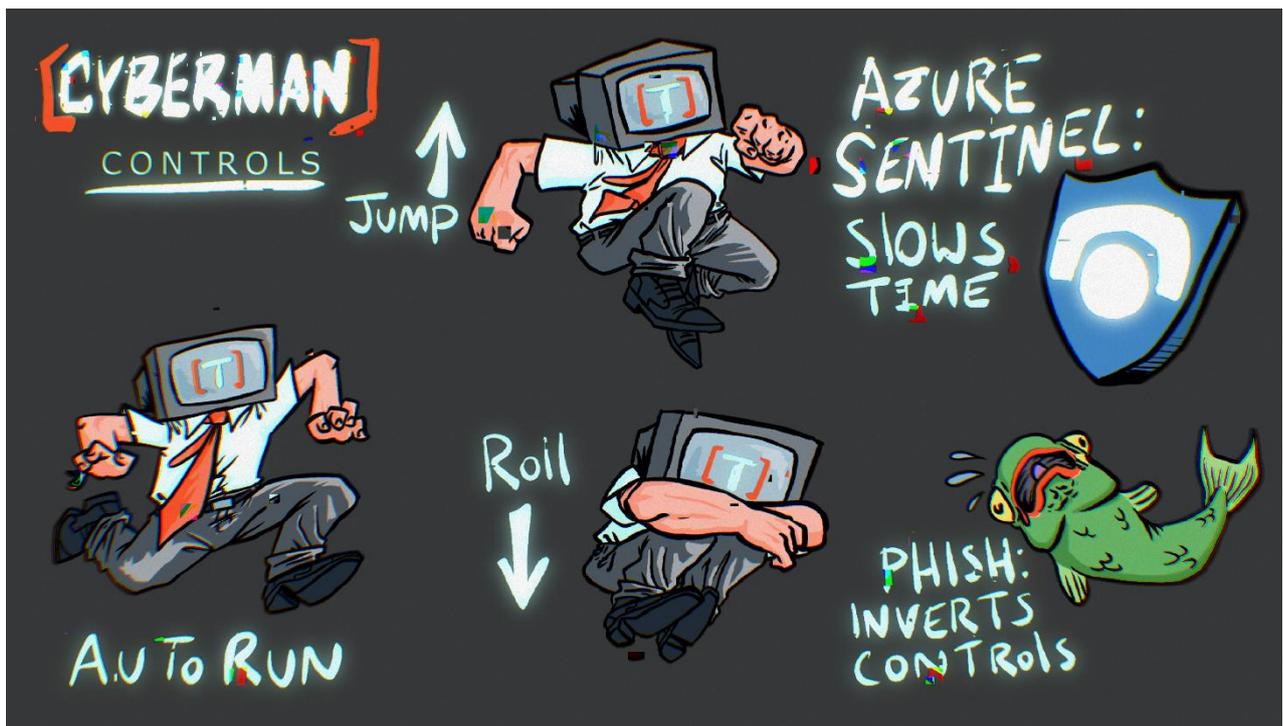
Splash Screens for Cyberman

I implemented two splash screens one as a welcome screen to introduce the player with nice imagery and the second a tutorial screen to get the player familiar to the buttons as a fast introduction. Sam drew up the art and designs for the pages note- The images were still blue to match the first title screen we would change this to black sometime in development.

Welcome Screen



Tutorial Screen



Sound Design

Audacity

Audacity is a free and open-source digital audio editor and recording application software known as a DAW (Digital Audio workstation). I used Audacity as it is great for creating sound effects in video games as the programme comes with a bunch of inbuilt features and I don't have to go to 3rd party software plug-ins to get the acquired effect.

Soundtrack

To soundtrack the game was used by a website that you can download royalty-free music, I looked for the style of using retro 8bit but more modified to adding the quirky elements used with the style of cyberman. The track I downloaded was called "Beep Beep"

Link to the source - <https://www.premiumbeat.com/royalty-free-music-genre/8bit?page=3>

I made edits in the track by adding recording my voice in Audacity with me saying "Thirdspace" in certain sections of the mix. I edited my voice in Audacity.

To create the desired effect in Audacity I recorded my voice on a standard microphone and used noise removal to get rid of any background noise/white noise to get clear audio. I amplified the soundwaves increasing the noise values, I then duplicated the voice and added echo and reverb effects to the second track. I changed the pitches between both tracks and duplicated the track another time and changed the tempo on the new track.

Sound Effects

I created the sound effects the same way I created them in the soundtrack. The sound effects I created were collecting the powerups with recorded voice audio "Phishing" when the Phish debuff is collected and "Powerup" when powerup is collected.

Cyberman

With everything implemented into gamemaker when completed a finished product for Thirdspace.

Brief description of Cyberman –

Cyberman is a running game about a man half monitor known as “Cyberman”! The game is about running as far as you can with getting hacked by bugs. These are not your standard bugs you get to see actual BUGS! Running in the cyber-world of third space you must survive and get the biggest score you can! Collect powerups along your adventure and stop the bugs hacking your cyberspace!

Features –

- Running fun without a gun!
- Scoring system
- Bugs and more BUGS!!
- Fast-pace action with a man for a monitor for a head.

Gameplay video link - <https://youtu.be/F9C0UyPObZQ>

APPENDIX A

Colour Pallet

Our gaming world, characters, items everything visual will be using these colour pallets provided by the brief that the client provided.

Primary palette

Thirdspace – “Colour plays an important role in defining our brand identity, as well as reflecting our cultural history and heritage. Therefore, consistency in the way we apply our colour palette is crucial for maintaining brand recognition.”



Secondary palette

Thirdspace – “At times we understand that a broader range of colours is needed, for example, to support process or flow diagrams and infographics. The following colours are acceptable to use in these situations, although where possible we would always advise staying true to our primary palette as much as possible. This secondary palette should be used sparingly”

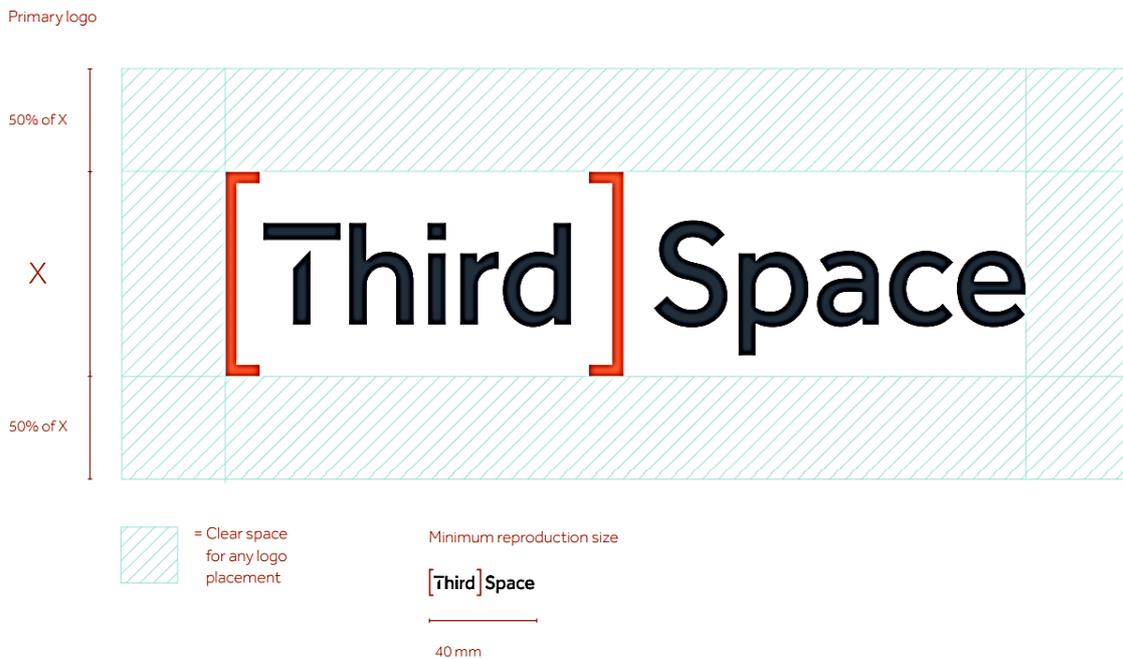


APPENDIX B

Thirdspace Logo

Primary Our logo is our most important asset and should never be distorted, manipulated or embellished. It can be used in three main formats as specified on the following pages: When written, the company should be known as 'ThirdSpace' rather than 'Third Space', 'thirdspace', 'Thirdspace' or 'Third space', or abbreviating to TS.

Our logo is part of a system that visually represents ThirdSpace as a company, and uses the brackets as a symbol that people can recognise and identify with. It is important to apply the logo with the two following principles always in mind: Iconic – We apply the ThirdSpace logo clearly and proudly on all communications. Consistent – We maintain the quality of our logo by only using the provided lock-ups and following the guidance on how to position them, maintaining legibility at all times



Our logo Lock-up & Variants

We have three main colour variations of our logo, allowing us to be flexible across a range of applications.

When selecting the correct logo for the application, please take care to ensure it will have maximum legibility

Primary logo – on White

[Third] Space

Primary logo – on Orange

[Third] Space

Primary logo – on Slate

[Third] Space

Icon & Avatar

In both square and circular app or social icons, the 'T' can appear as a singular mark. Spacing varies slightly between square and circular applications of the symbol. In a square application, the symbol may be metrically spaced to the centre of the box, with reasonable clear space applied. In a circular crop, the 'T' must be scaled down further to avoid the edges of the brackets being cropped too close to the edge of the circle. Suggested scale and proportions are shown in examples to the right. There are three colour options available.

Square crop – White



Square crop – Orange



Square crop – Navy



Circular crop – White



Circular crop – Orange



Circular crop – Navy



Misuse

The treatments shown on this page are not allowed, the ThirdSpace logo mark is our most important visual asset. To ensure consistency of our brand please follow the guidelines given and use the relevant master digital artworks.



DO NOT change the kerning or spacing of the word mark.



DO NOT change the typeface of the word mark.



DO NOT change the position of the brackets in relation to the symbol.



DO NOT add a drop shadow to the logo.



DO NOT add an outer glow to the logo, in any colour way.



DO NOT change the colour of the 'brackets' or the word mark from the primary or secondary logotypes.



DO NOT change the 'brackets' mark to a filled in shape, rather than an outline.



DO NOT distort the vertical or horizontal aspect ratio of the logo.



DO NOT change the line weight or curves of the 'brackets'.

APPENDIX C

Typography

Headline

Thirdspace headline font is Effra Medium. It's bold and modern, providing a feeling of clarity, expertise and maximum legibility. Typography plays an important role in communicating an overall tone and quality of the brand. Careful use of typography reinforces our personality and ensures clarity and harmony in all communications. We have selected Effra for the logo font and artwork design, and Segoe UI for all other documents. Effra Light may be used as subheaders, secondary titles and introductory paragraphs. Wherever possible, the text should be left-aligned. Type should be set in sentence case.

Headline

Effra Medium

abcdefghijklmnopqrstuvxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890[!@£\$%^&*()_+::"'\|}]

Sub header

Effra light

abcdefghijklmnopqrstuvxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890[!@£\$%^&*()_+::"'\|}]

Body

The body font is Segoe UI Regular. Selected for its legibility at smaller sizes, particularly in digital formats. Segoe UI Bold may be used to add emphasis. We never add type effects such as drop shadows or outline type. Segoe UI should be used for all text and headings in documents such as PowerPoint and Word documents, that are not for marketing purposes.

Please note: Use bold upper and lower case for emphasis rather than all capitals.

Body emphasis

Segoe UI Regular

abcdefghijklmnopqrstuvwxy
ABCDEFGHIJKLMNPOQRSTUVWXYZ
1234567890[!@£\$%^&*()_+;:"'\}]

Body

Segoe UI Bold

abcdefghijklmnopqrstuvwxy
ABCDEFGHIJKLMNPOQRSTUVWXYZ
1234567890[!@£\$%^&*()_+;:"'\}]

Styling

Thirdspace use a range of typographic styles in our brand communications. Examples can be seen on the right-hand side of the page. A few key rules of thumb are:

- Point size should be 10–12 points in printed documents, 15-25 pixels on the web.
- Line spacing should be 120– 145% of the point size.
- Type should be left-aligned for maximum legibility.
- Use centred text sparingly.
- Use bold for subtitles or keywords and use italic as little as possible.
- All caps are fine for less than one line of text.

Highlighter boxes

The explosion of cloud technology is driving a move away from traditional on-prem systems to cloud and hybrid **IDaaS solutions.**

The global IAM market is expected to grow to USD **\$20.87 billion** by the end of year 2022 with a compound annual growth rate of **14.8%** ^[1]

Bullet points

- Identity life cycles
- Intuitive access request systems
- Segregation of duty management
- and most importantly – auditing!

Introductory paragraph

The job for CISOs and IT managers isn't getting any easier. The global demand for Identity and Access Management (IAM) is outstripping enterprise organisation budget spending and the IAM skills gap in the market continues to grow.

The global IAM market is expected to grow to USD \$20.87 billion by the end of year 2022, with a compound annual growth rate of 14.8%.^[1] Meanwhile, average enterprise organisation IT budgets, despite growing, remain tight, with data showing enterprise firms on average are spending only 8% of overall IT/Security budgets on IAM.^[2]

Citations

1. Technavio (June 6 2018). "Global Identity and Access Management Market 2018-2022". Orbis Research. Available at: <http://www.orbisresearch.com/reports/index/global-identity-and-access-management-market-2018-2022> [accessed 20.07.18].

2. Cser A, Maxim M et al (January 9 2018). "Top Trends Shaping IAM in 2018". Forrester Report.

Section header & key questions

Where should IT leaders focus time and resources?

Strong data governance practices must start with having a complete understanding of all identities in your organisation. This in turn allows you to manage access to sensitive data based on user privileges, roles and access rights.

Headline & sub-header

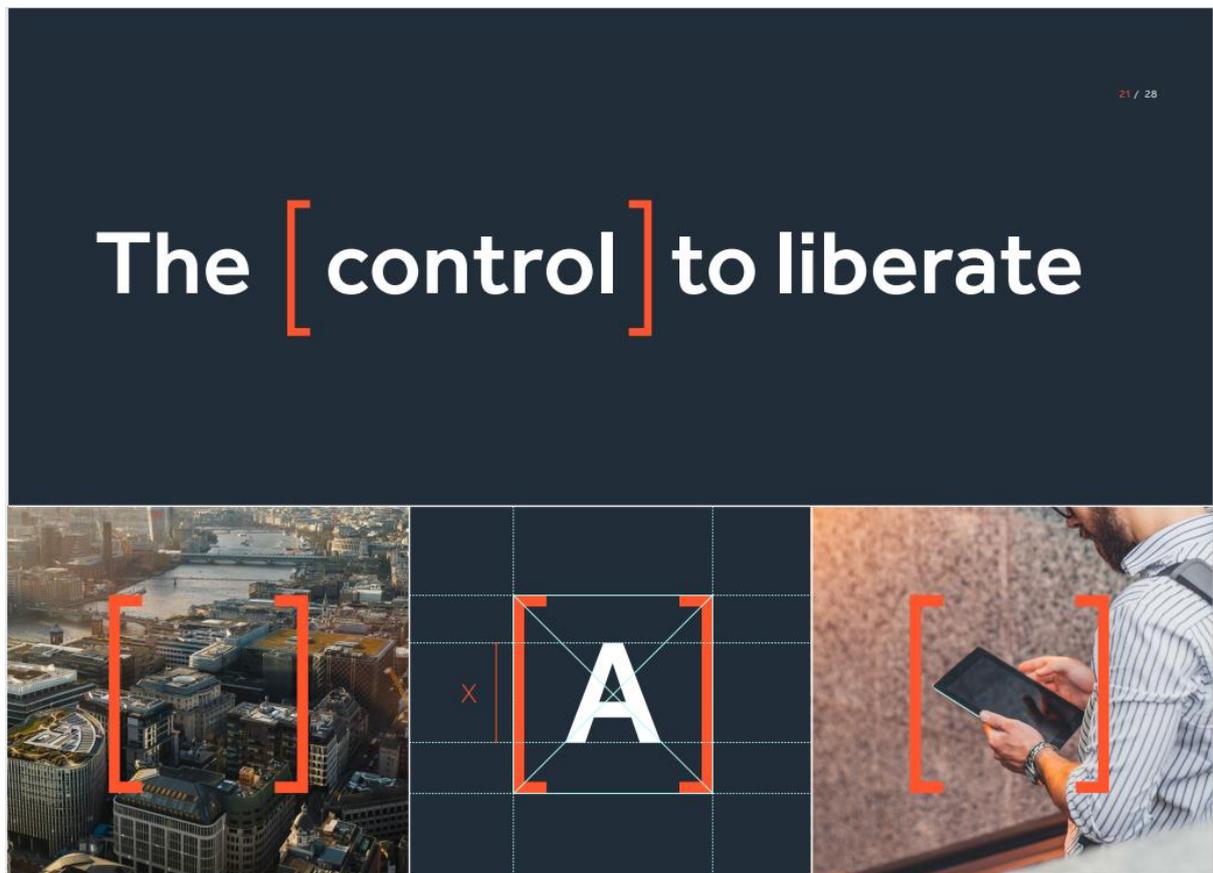
What are the biggest trends in Identity and Access Management?

How can CISOs evolve strategy to overcome the challenges of tomorrow?

APPENDIX D

Brackets

Used to highlight We use the brackets from our logo as a highlighter for key imagery and in messaging. This is to highlight the 'ThirdSpace' – whether that be a person, device, word or location. When used within typography, the brackets should be set to double the X-height of the headline font (as illustrated on the right-hand side). This creates a visual harmony between the two elements to mirror the composition of the main logo.



APPENDIX E

Type as image

IMAGERY Wordplay can be used to highlight singular keywords and phrases within a sentence or short statement.

