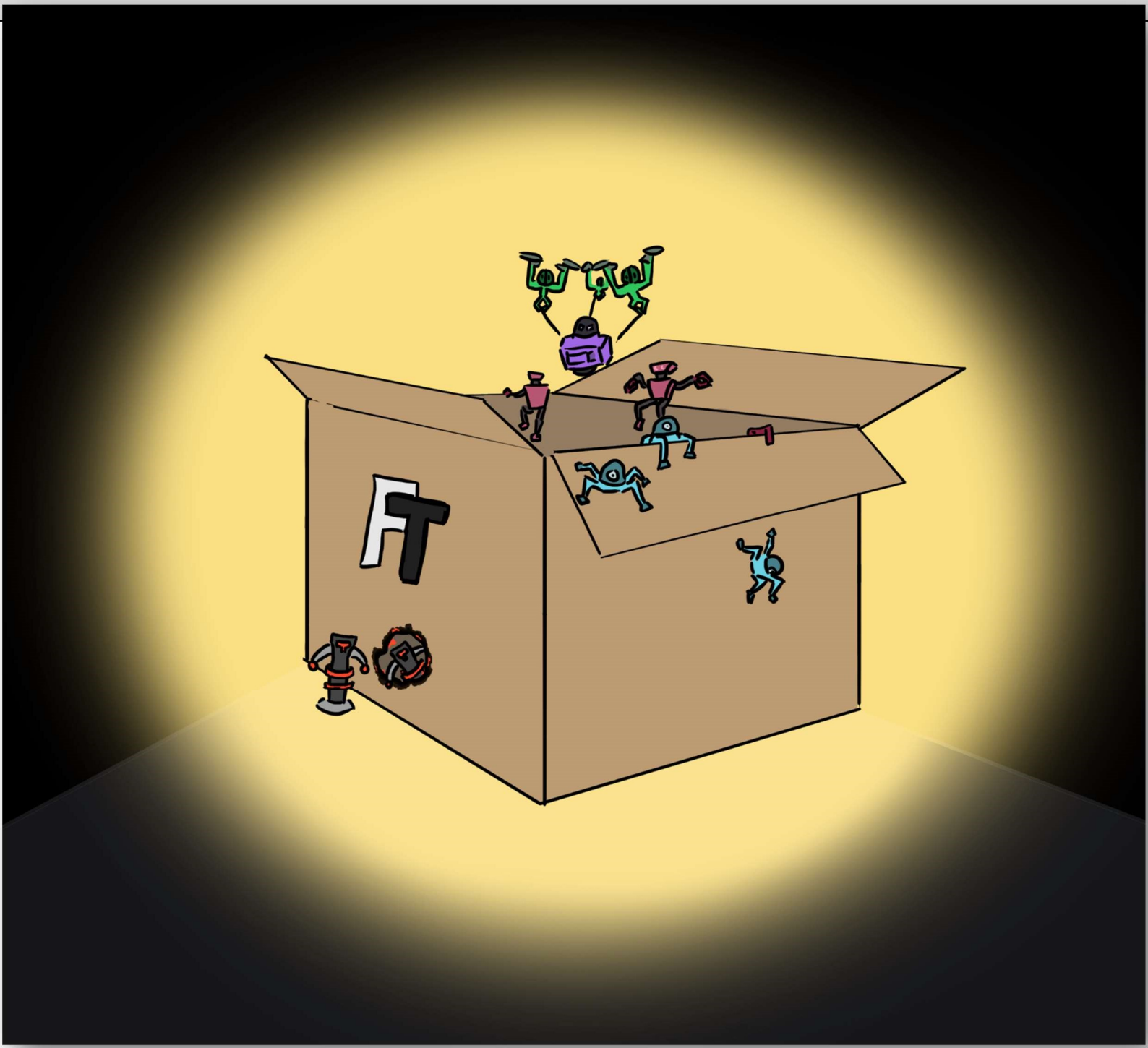
Nano Bots

GAME DESIGN DOCUMENT



Contents

Game Overview

Nano bots is a puzzle/real-time strategy game about commanding small robots to perform actions in order to complete goals. It will be developed for Augmented Reality (AR) on mobile devices. The game will consist of short physics-based challenges that the player will be able complete at their own pace. Speed will be incentivised with a bonus to score, in order to encourage multiple playthroughs of levels as players attempt to beat their own scores and others.

Setting

The game is set in the distant future, in which small robots or Nano bots are used ubiquitously for day-to-day tasks. Tasks that humans find tedious or time consuming are given to Nano bots. Thus, exists a market of competing companies that produce said Nano bots, each model with its own area of specialism and unique abilities. The game centres around one particular leading brand of Nano bot, named *Friendly Tech*.

Design Ethos

The visual design of Nano bots will be centred around minimalist low-polygon models and simple pastel colours. The general aesthetic of the user interface and environment will take inspiration from futuristic media, bringing the game in line with its setting.

Platform

Nano bots will be designed and deployed for mobile devices. This will initially consist of Android devices but could be later expanded to Apple products as well. An AR software development kit (SDK) will be used as a basis for all AR functionalities, with all levels and scenes being placed in accordance with the environment. The game will be available on the Google Play store and will utilise the Google Play Games SDK for features such as tracking scores and leaderboards.

Concept Research

This section will cover research into similar existing examples of video games in the same genre. The goal being to ascertain key mechanics and features as well gaining inspiration for design elements.

Pikmin 2

Figure 1: Pikmin 2 (2004)

Pikmin 2 (2004) is a puzzle strategy game in which the player is tasked with managing groups of alien creatures, the player can command these creatures to complete tasks such as collecting and carrying items or breaking down obstacles. The amount of characters under the player's control changes throughout levels as the player progresses, improving the ability to complete tasks as more and more characters are accumulated by the player. The game is centred around the concept of rebuilding a rocket ship that the main character crash landed on the planet with. There is a time limit in the form of a day-night cycle, the player must spend the day completing various tasks making sure to gather characters before it gets dark. Pikmin 2 (2004) uses a simple art style with flat colours used to indicate characters and objects. Characters consist of simple models which are consistent and identifiable, with differences in colour to indicate their abilities. This simplistic style makes actions clear and obvious to the user. This design scheme would apply well to the Nano Bots concept, as it will facilitate user experience as well as reduce the visual complexity, improving the game's performance.

Overlord

Figure 2: Overlord (2007)

Overlord (2007) is an action role-playing game set in a fantasy world where the player's role is to command minions of varying types and skillset. Skin colour of the minions is used to denote what they are capable of doing, for example, "Greens" are able to use stealth and are immune to poison whereas "Blues" can swim underwater and heal other minions. Each minion type benefits off each other, creating an element of strategy in building a balanced army of units. This inclusion of a system where variety in unit type is encouraged provides opportunity for players to experiment with the mechanics attached to said units. Resulting in a multitude of possible solutions to challenges and therefore a reason to go back and retry levels with a different approach. Such variety would be well suited to Nano bots.

Surgeon Simulator

Figure 3: Surgeon Simulator (2013)

Surgeon Simulator (2013) is a physics-based game in which the player is required to perform surgery in a limited amount of time, without the patient losing too much blood. Controls are simple but difficult to master with objects reacting to collisions with the characters hand. A multitude of tools are provided at the onset of each level, providing a variety of different ways to complete each surgery. As Nano bots will also be using physics, Surgeon Simulator (2013) provides a good example for a system in which tasks are made challenging by introducing a method of input that forces the player to approach said task in a novel way.

Features/Mechanics

This section will cover all the mechanics and features that will be included in the final game.

Control Methods

The user will be able to interface with the game via touch screen.

- The game will use Augmented Reality, so camera movement will be based on how the user moves the device.
- The game view will be combined with the device's camera.

Player Actions

These are actions that will be performed by the player:

- Task Assignment e.g. move to x,y location.
- Select character to assign task.

AI Actions

These are actions that will be performed by the AI:

- Unique jobs based on character type:
    - Strong characters will be able to lift objects.
    - Light characters will be able to climb.
    - Heavy characters will be able to act as counter-weights.
- Task management.
    - Delegate tasks to characters based on their abilities.
    - Prioritise tasks on a first-in-first-out basis.
    - Perform actions in appropriate order e.g. move to position before attempting to pick up object.
- Interaction with physics.
    - Possible actions.
        - Push.
        - Pull.
        - Use item.
        - Lift.

AI movement will be handled using Unity NavMesh, actions and character decisions will need to follow a hierarchical system.

AI Classes

AI code will be designed using classes, to facilitate the sharing of methods between different character types. Below is a piece of pseudocode that will form the basis of character classes.

```
//Nano bot base class
var mass, strength, speed = Base float properties
var climb = Boolean indicating if character can climb
var agent = Navigation agent attached to character

Start(){
      If(climb == true){
            Set agent type to climbable
      }
}

public method for movement(x,y location){
      agent target = x,y location;
}

Public method for lifting(targetObject){
      If(close enough to targetObject)
      {
            If(targetObject.mass < total strength of characters lifting)
            {
                  lift targetObject;
            }
      } else{
            Move toward targetObject;
      }
```

Figure 4: Base class for nano-bot behaviour

This code will then be inherited and built upon depending on the skills or abilities a character type may have. The agent type will be set in runtime to ensure that climbing agents are able to move across off-mesh links, where non-climbing agents will not.

AI State Machine

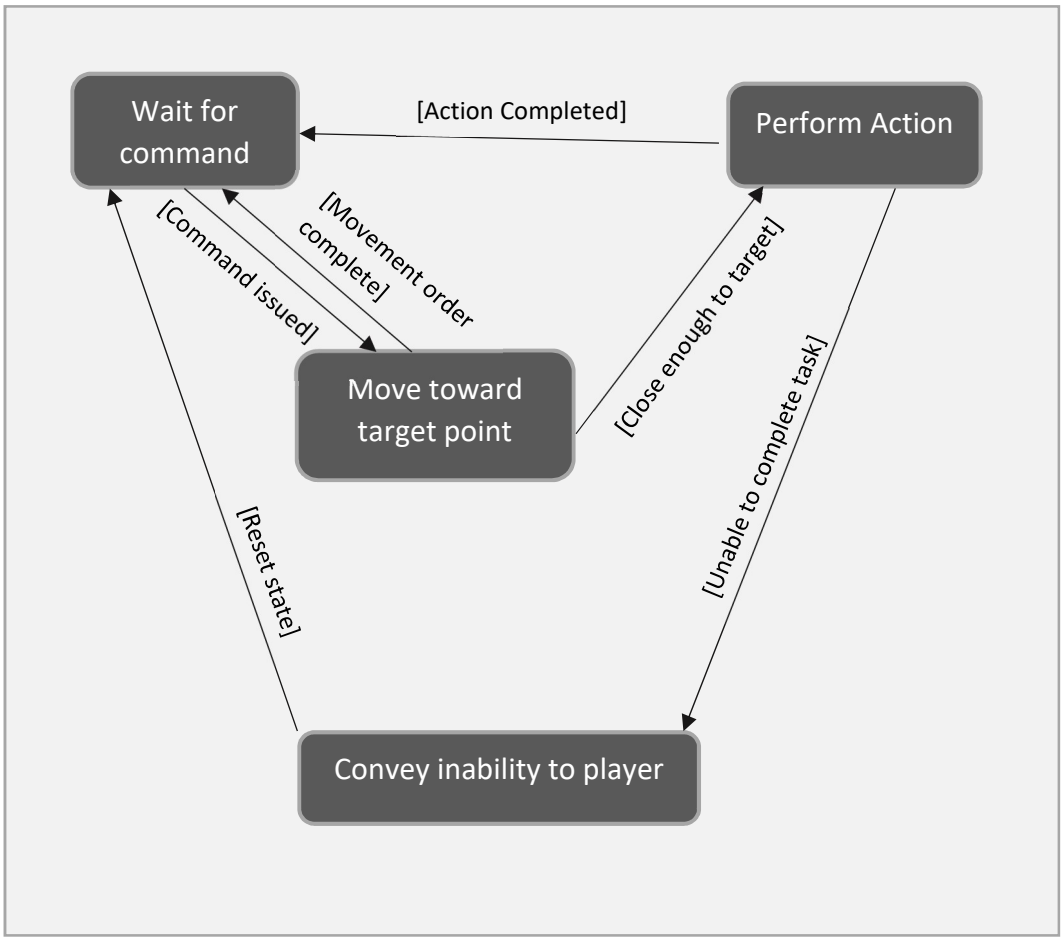


Figure 5: State machine chart

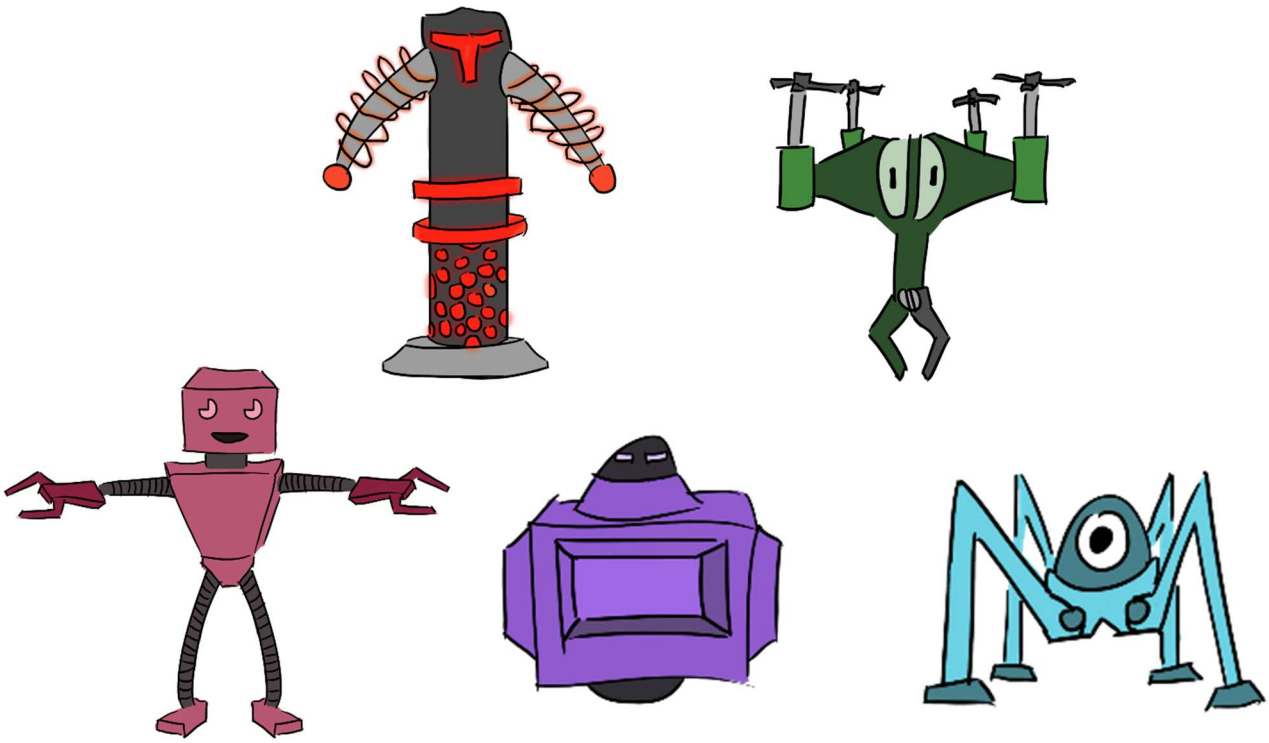
All artificial intelligence in Nano bots will follow a core state machine which will dictate what actions the character will perform at any given time. Any input from the player will either be a movement order or action order, the latter of which requiring the former. This is done to ensure that the character/s are close enough to their target object before performing an action. This state machine will exist as a layer of logic on top of the Navmesh AI in Unity. Therefore any movement order will be parsed through to the destination property of the corresponding Navmesh agent.

Scoring

The player will be scored on how quickly they complete a challenge, providing a reason for players to try to improve their strategy.

- High score board.
- Score presented at end of level.
- Timer on screen.

Designs

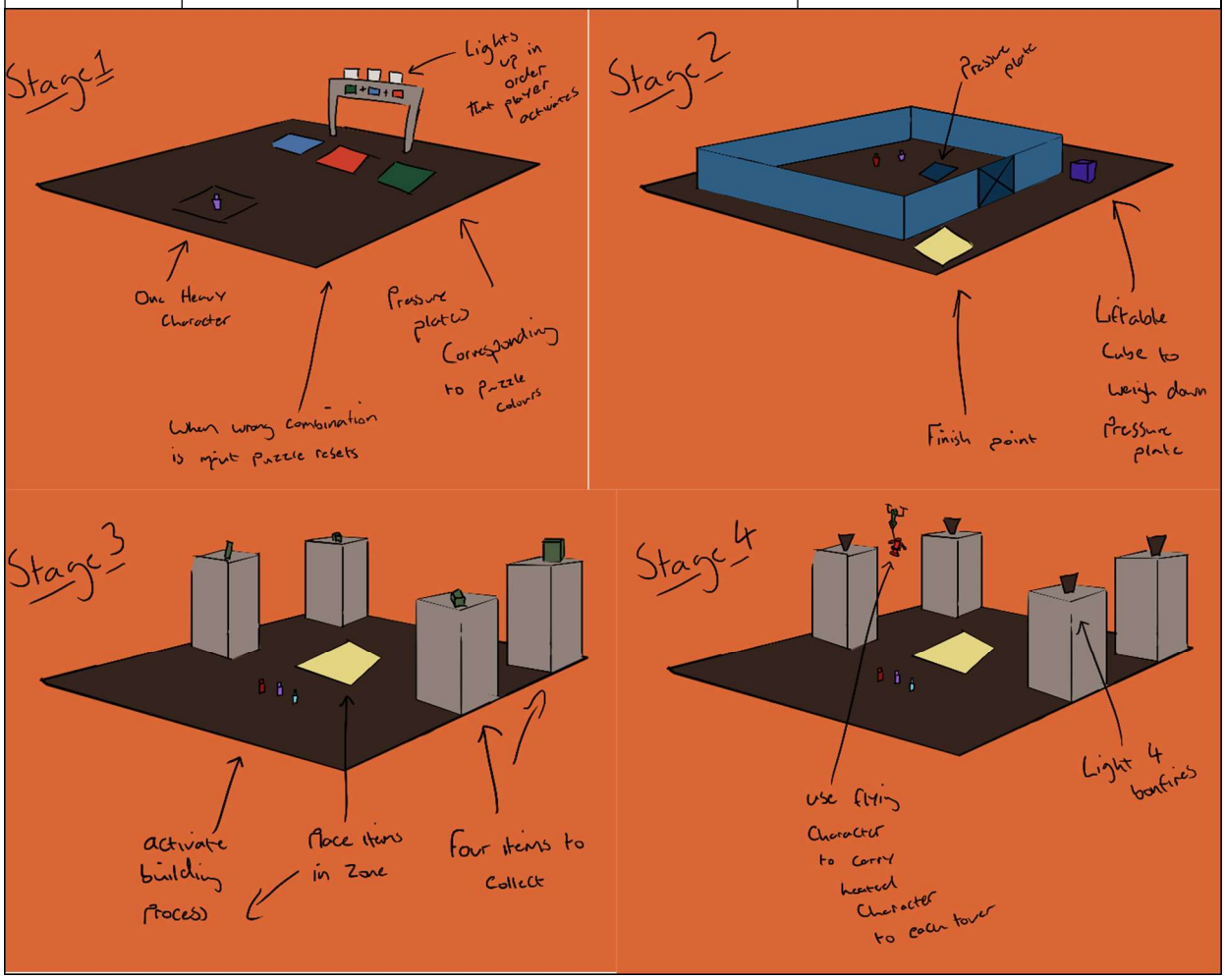


Gameplay

This section will cover level and interaction designs.

Levels

Level	Stages/Steps	Assets needed
Tutorial	1. Use heavy character to activate pressure plates in correct order- unlock strong character. 2. Release strong character from room by activating pressure plate with heavy character. Strong character can then carry block to weigh down pressure plate so both characters can exit- unlock light character. 3. Light character is used to bring down items from the top of four towers. The items need to be placed in the centre of room to create a flying character and hot character. 4. Use flying character to lift hot character up to light fires at the top of each tower. Tutorial is finished.	1. Walls 2. Towers 3. Animated door 4. Pressure plates 5. Building components



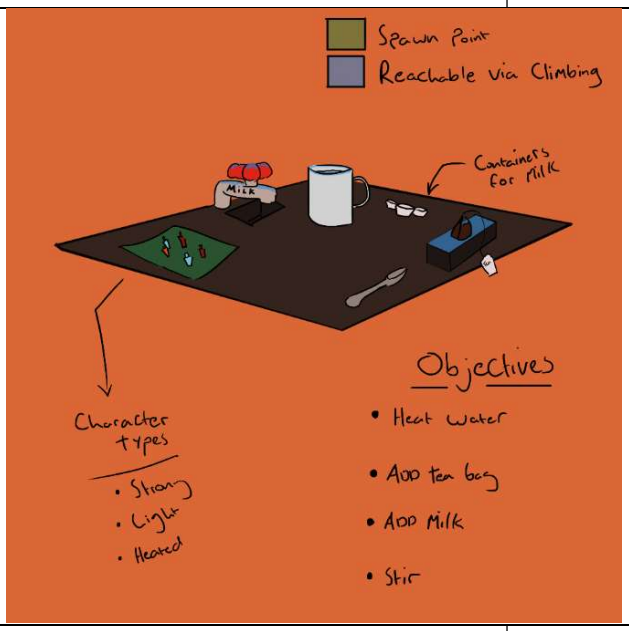
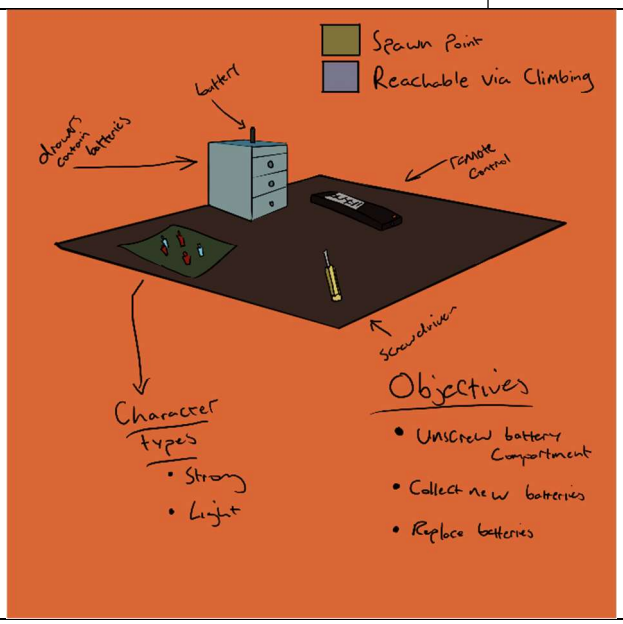
Level 1: Make a cup of tea	1. Boil kettle 2. Get a tea bag in the mug 3. Add water 4. Add milk 5. Stir	1. Model of mug 2. Model of tea bag 3. Water and milk (fluid dynamics) 4. Model of spoon 5. Surrounding geometry to facilitate certain steps
		
Level 2: Replace batteries	1. Flip remote 2. Find screwdriver 3. Unscrew battery compartment 4. Remove old batteries 5. Place new batteries in compartment 6. Screw cover back on	1. Model of television remote controller 2. Model of screwdriver 3. Model of battery cover 4. Model of Battery 5. Surrounding geometry to facilitate certain steps
		

Table 1: Game Levels

Characters

This section will cover the design process for the characters in the game. Beginning with basic shape experimentation, then moving through development of design and character specification.

Inspiration

Figure 6: Character Inspiration

Above is a collection of images that inspire the Nano bot character designs. Overall the shapes and colours appear simple and elementary, with limbs and anatomy often times relating back to humans or animals. The simple colours and distinct shapes of characters in the movie Robots (2005) differentiate them from one another, accentuating their personality providing instantaneous visual cues.

(Android, n.d.) (Half-Life 2, 2004) (SJRC, n.d.) (Sanctum 2, 2013) (Robots, 2005) (GuidoVrola, n.d.)

Basic Shapes

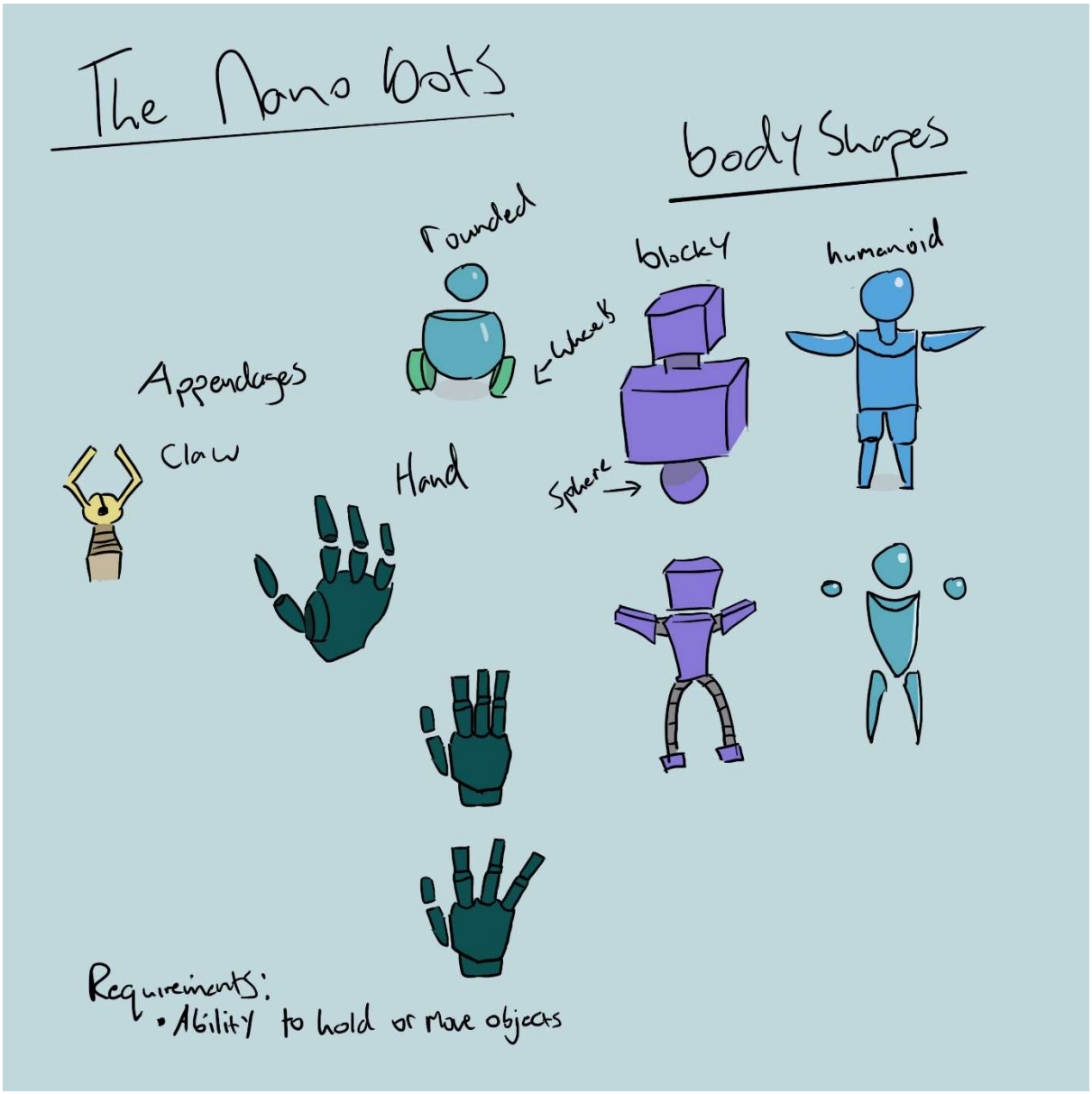


Figure 7: Basic shape experimentation

Initial shape designs for Nano bot characters. Characters would need to appear practical for their actions to match reality. Additionally, body shapes would need to facilitate animation for movements to appear natural. Limbs would either need to be disjointed or appear bendable for walk cycles or actions such as lifting. Wheels could be used instead of legs to differentiate a character type and claws could be added to infer that a character is able to lift heavy objects.

Development

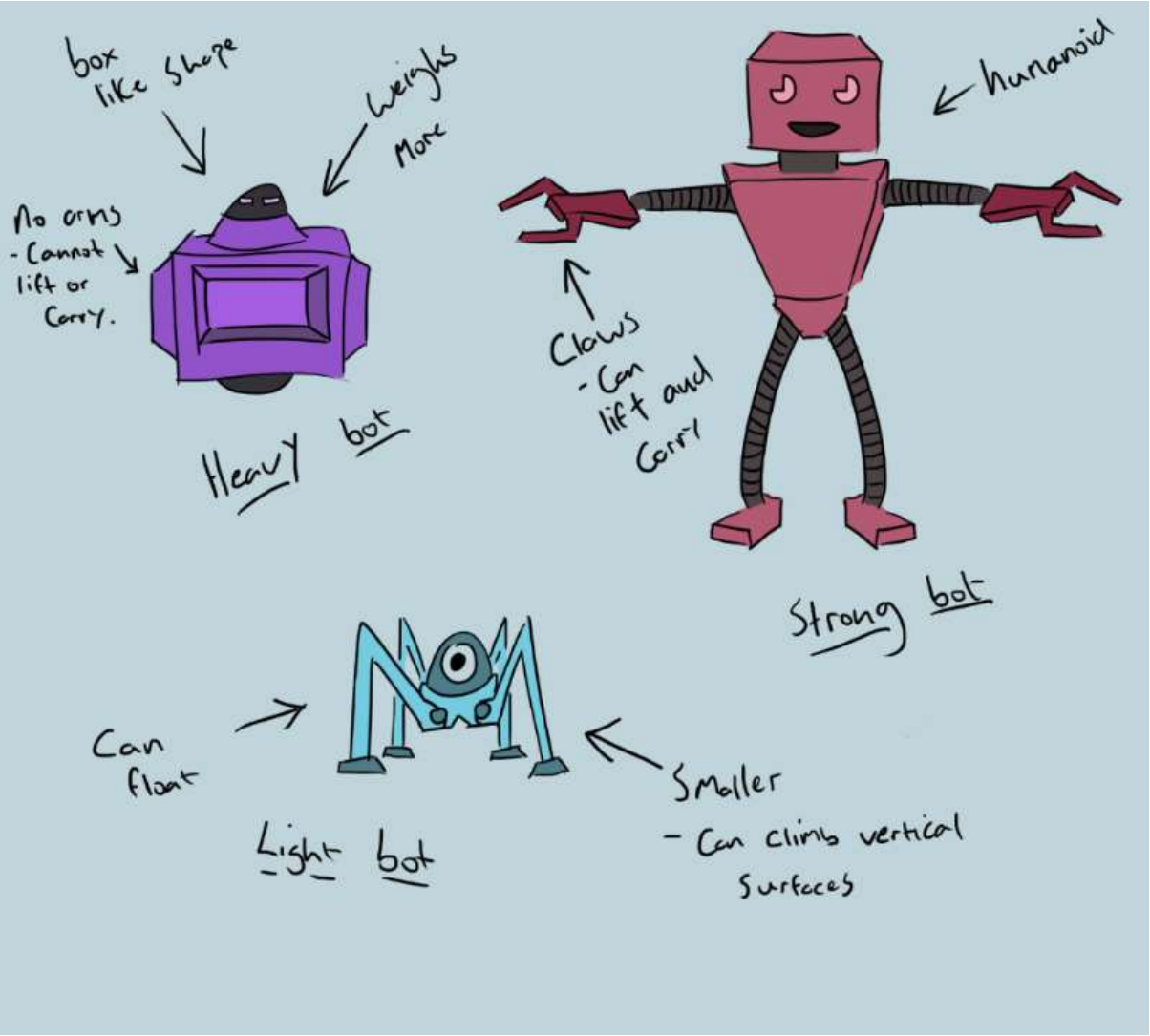


Figure 8: Character Designs

Above are three final designs for character types that will be included in the game. The goal was to differentiate characters in shape, form and colour scheme

to make distinguishing between them easier.

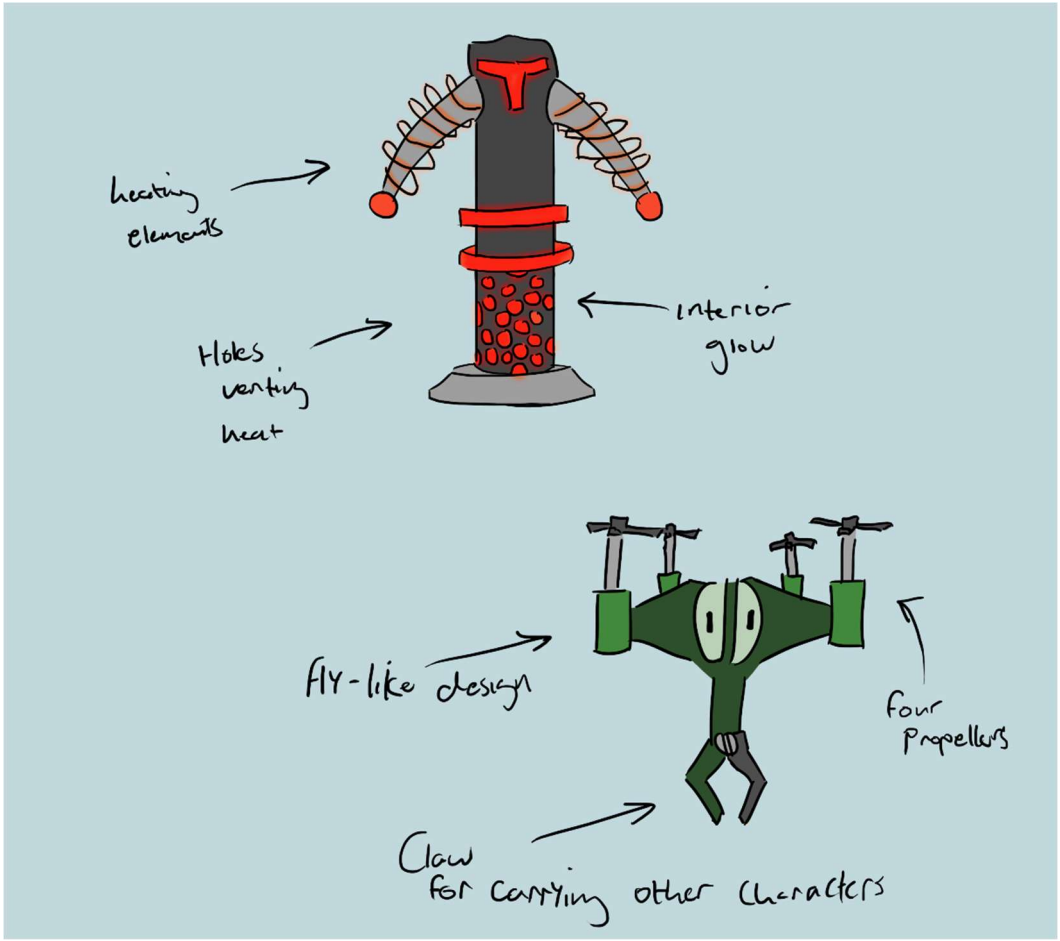


Figure 9: Additional Character Designs

A couple new character types were designed as a result of the prototype seminar feedback. The goal was to expand game mechanics beyond Newtonian physics to create more choice and variance in gameplay. One such character has been designed around the ability to create heat, inspiration was taken from various heating elements used in household devices such as toaster ovens. The other character designed was based on the ability to fly and lift other Nano bots to higher places. This took inspiration from flies and remote control drone devices.

Character Profiles

This section will be going over each character's appearance, abilities and stats. Each character type will have a simple distinct colour scheme with two main colour swatches. Audio for each character will be handled using melodies from a digital synth, differentiated by musical key. Character names will be abbreviated with a consistent suffix based on the company name *Friendly Tech*.

B.I.G.F.T

Name in full:

Boundless Indestructible Great-bot by Friendly Tech

Description:

The main purpose of this type of bot is to act as a weight. Movement is achieved by a rotating ball at its base, allowing it to traverse difficult terrain while retaining balance. It's lack of appendages results in its inability to carry objects.

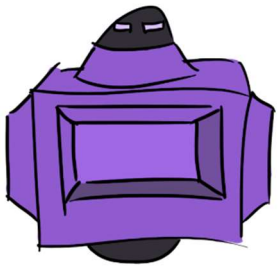


Figure 10: B.I.G.F.T Nano-Bot

Colour Scheme:

Grey and Purple

Figure 11: B.I.G.F.T Colour scheme

Stats

Lifting Strength	Mass	Movement speed	Climb
0	15	5	✗

Table 2: B.I.G.F.T Stats

Animations:

- Movement animation- ball rolling. When coming to a stop its chassis rolls forward slightly due to inertia.

Audio:

Vocal Key: G

Movement: Rolling sound

L.I.F.T

Name in full:

Loading and Interdisciplinary bot by Friendly Tech

Description:

This type of bot is the core of any successful Nano bot team. Their strength allows them to lift objects much larger than themselves. However, they cannot climb or move as fast as their lighter counterparts.

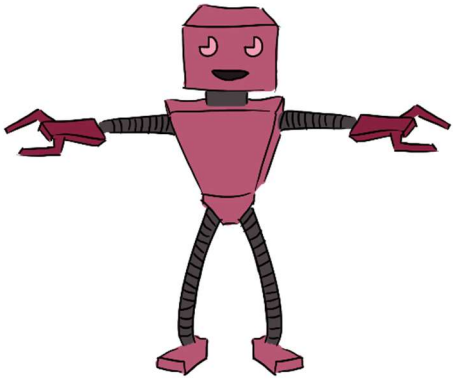


Figure 12: L.I.F.T Nano-Bot

Colour Scheme:

Maroon and Pink

Figure 13: L.I.F.T Colour scheme

Stats:

Lifting Strength	Mass	Movement speed	Climb
15	5	10	✗

Table 3: L.I.F.T Stats

Animations:

- Walk animation- bipedal.
- Lift and push animations.

Audio:

Vocal Key: C

Movement: Tapping sound for each footstep

S.W.F.T

Name in full:

Small and light Weight bot by Friendly Tech

Description:

This type of bot can traverse surfaces at high speed, with their light chassis allowing them to float and climb vertical surfaces to reach hard to get places. They lack in strength; therefore, they are unable to carry heavy objects.

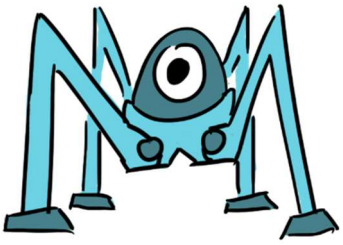


Figure 14: S.W.F.T Nano-bot

Colour Scheme:

Light Blue and Grey.

Figure 15: S.W.F.T Colour scheme

Stats

Lifting Strength	Mass	Movement speed	Climb
5	1	15	✓

Table 4: S.W.F.T Stats

Animations:

- Climb
- Move- spider like leg movement

Audio

Vocal Key: A

Movement: Fast tapping sound

F.L.Y.F.T

Name in full:

Free-flying bot by Friendly Tech

Figure 16: F.L.Y.F.T Nano-bot

Description:

This type of bot is able to travel at varying altitudes using its four propellers. Its claw module allows it to carry light to medium weight objects with relative ease.

Colour Scheme:

Green and Grey.

Figure 17: F.L.Y.F.T Colour scheme

Stats

Lifting Strength	Mass	Movement speed	Climb
5	1	10	✓

Table 5: F.L.Y.F.T Stats

Animations:

- Grab, when carrying item.
- Move- rotating propellers.

Audio

Vocal Key: F

Movement: Buzzing sound

P.Y.F.T

Name in full:

Pyrotechnic bot by Friendly Tech

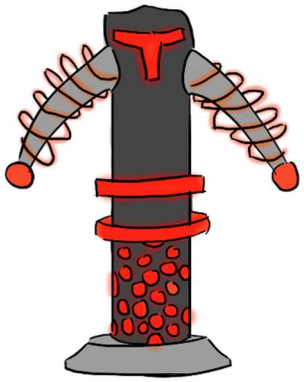


Figure 18: P.Y.F.T Nano-bot

Description:

This type of bot is specialised in producing heat. What it lacks in lifting ability it makes up for in its ability to spontaneously combust, making it useful for boiling water, lighting fires or simply warming a room up.

Colour Scheme:

Red and Grey.

Figure 19: P.Y.F.T Colour scheme

Stats

Lifting Strength	Mass	Movement speed	Climb
0	5	8	X

Table 6: P.Y.F.T Stats

Animations:

- Move- bouncing
- Heating ability

Audio

Vocal Key: B

Movement: Thud with each bounce

User Interface

This section will cover the implementation of UI, exploring how the user will interact with the game.

Character interaction

How the user can interact with the mobile device:

- Tap
- Tap and hold
- Drag
- Pinch
- Double tap

The user will need to be able to command multiple characters in quick succession, actions will be carried out by A.I.

As there is limited screen space on a hand-held mobile device UI will need to be kept to a minimum, presenting only the necessary information. The screen is used to display the game content as well as handling touch controls, so all necessary buttons will need to be displayed on the sides of the screen, where they are accessible while holding the device and out of the way of the content.

The decision was made to limit control of characters to groups, therefore removing the ability to select individual characters. This will force the user to control characters on a group level, reducing the amount of UI needed, and preventing the level of control from being overwhelming.

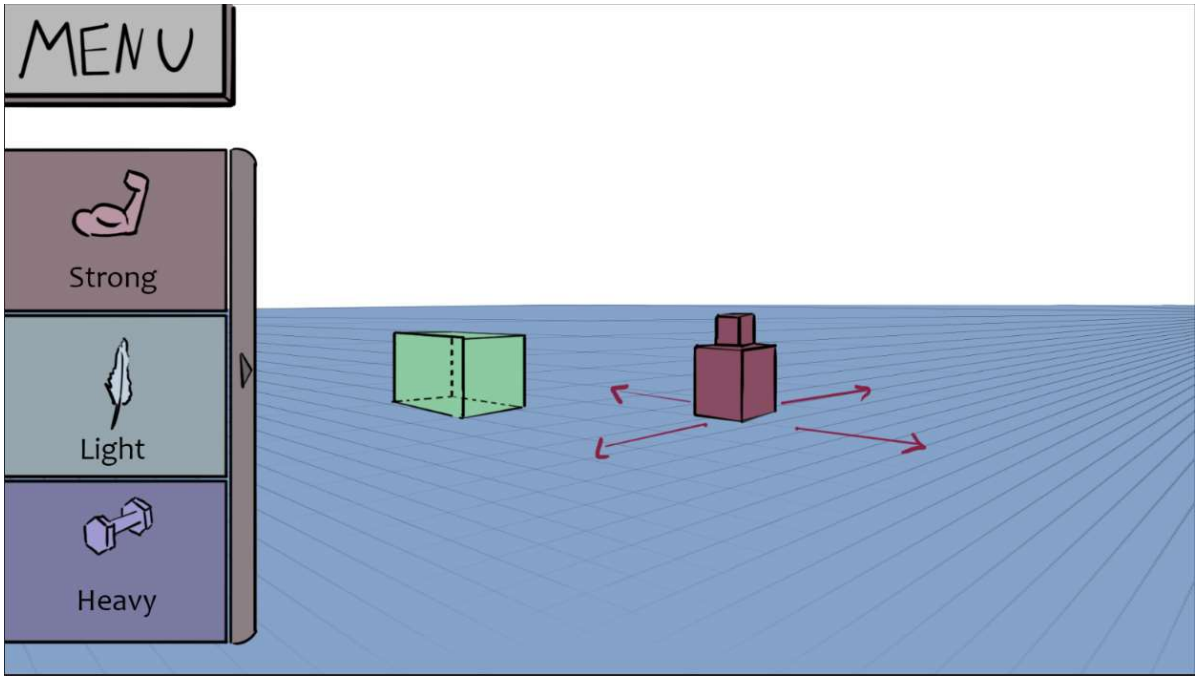


Figure 20: In-game UI

Menu Hierarchy

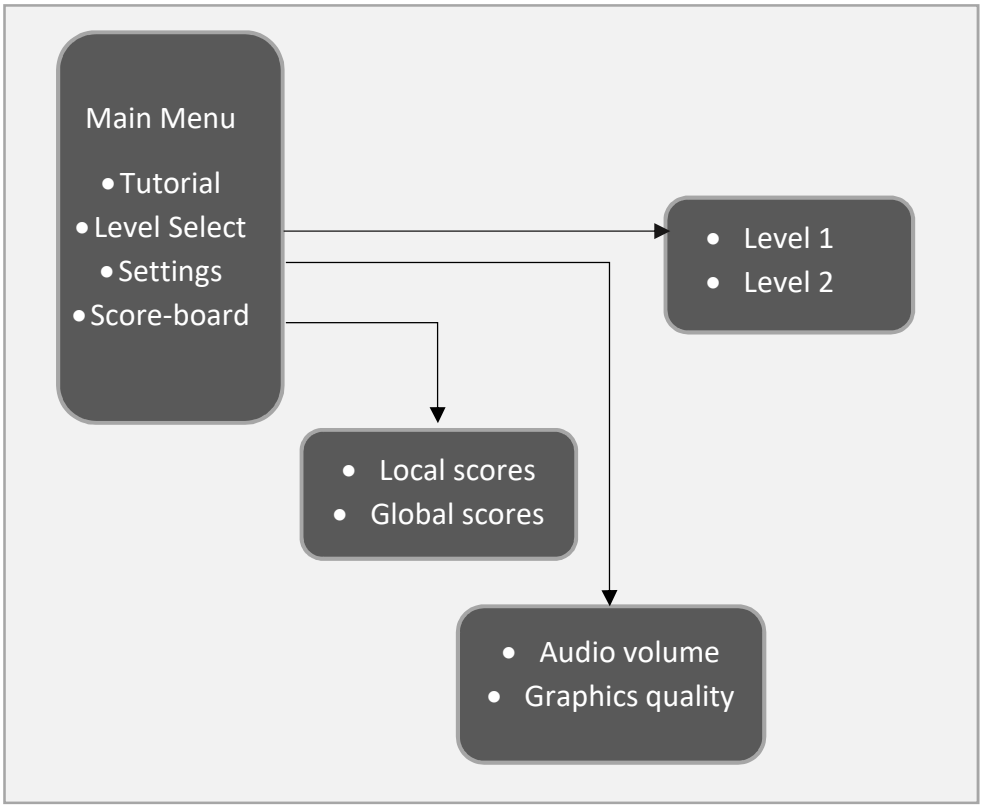


Figure 21: Main menu hierarchy

When the game loads up the player will be presented with the option to either play a tutorial level or enter the level selection menu. Alternatively, the settings menu can be opened, which will allow the player to change the audio volume and/or graphical quality of the game based on preferences. The score-board is also available from this screen allowing players to compare their level completion times with friends or globally through the Google Play API.

In-game Menu Hierarchy

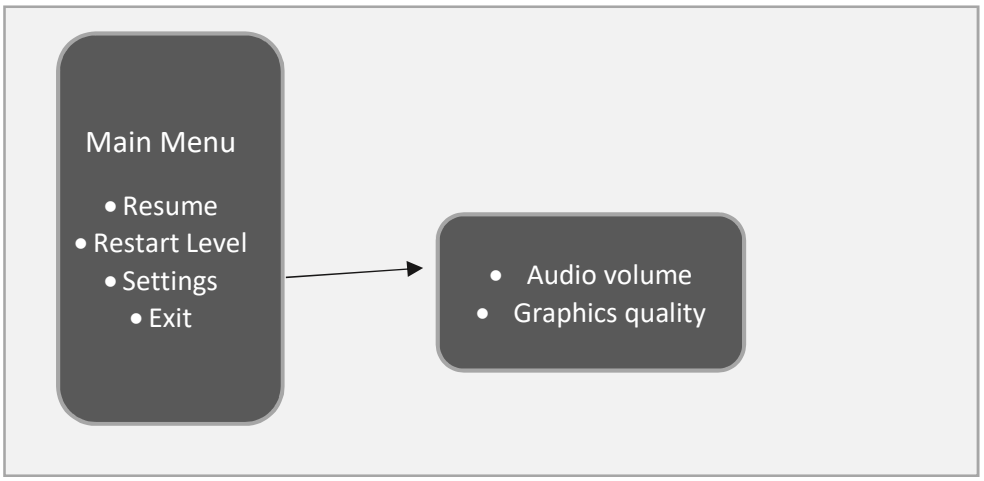


Figure 22: In-game menu hierarchy

A different menu will be available while a level is being played. This will allow the player to restart the level if they find themselves stuck or if they are attempting a high score and make a mistake.

Sound Design

This section will cover how sound will be used in the finished product.

Sound effects

Some sound effects will be used to provide the player with feedback for their actions.

Potential user actions that should have sounds:

- Select
- Pause
- Un-pause
- Issue Command
- Button press in menu

Actions that are indirectly related to user input that should have sound:

- Character pick-up
- Character drop
- Character moving
    - Different types of characters may make different sounds when moving
- Victory sound (when a challenge is completed)
- Personal best or record sound

All characters will use melodies as audio cues for different responses to commands. These melodies will be transposed to different keys depending on the character type, emulating different tones of voice.

Assets

Asset	Production
Levels	
Tutorial Level Geometry	3D modelled using 3Ds Max
Level 1 Geometry	
Level 2 Geometry	
UI Elements	
Main Menu	Icons and graphics created in Adobe Photoshop. Text and layout in Unity.
Character Selection Menu (Pre-game)	
In-Game Character Menu	
Settings Menu	
Mission Completion Screen	
Score Table	
Level Selection	
Characters	
Strong	3D modelled using 3Ds Max. Animated within Unity
Light	
Heavy	
Hot	
Props	
Tea Spoon	3D modelled using 3Ds Max
Tea Bag	
Mug	
Liquids (water etc.)	
Remote Controller	
Battery	
Battery Cover	
Screwdriver	
Audio	
Select	Produced using Audacity, LMMS and BFXR. Alternatively sourced.
Pause	
Un-pause	
Issue Command	
Button press in menu	

Character pick-up	
Character drop	
Character moving	
Victory sound	
Personal best/record sound	

Table 7: Project assets table

Roadmap

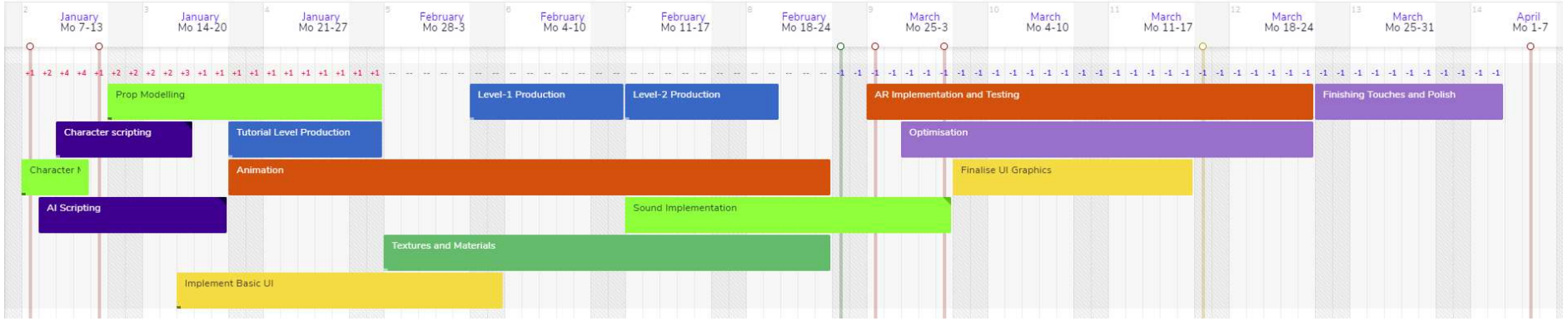


Figure 23: Project roadmap overview

The above Gantt chart covers the production plan for the game, between the 7th of January 2019 and the 7th of April 2019.

January 7-13	January 14-20	January 21-27	January 28 – February 3	February 4-10	February 11-17	February 18-24
Hackathon 1	All AI and character scripting will be finished with a working prototype using some modelled props.	Functioning tutorial level with some animations added.	Half of the animations will be finished as well as textures and materials.	Functioning first level will be completed.	A mostly functioning second level as well as some sound will be implemented.	All levels and 3D assets, including animation and materials will be completed.
February 25 – March 3	**March 4-10**	**March 11-17**	**March 18-24**	**March 25-31**	**April 1-7**	
Hackathon 2	UI graphics, optimisation and AR implementation will continue to be worked on.	UI graphics will be finalised.	Testing and optimisation process will finish, and product will be mostly complete.	Minor adjustments will be made.	Final Deadline	

Table 8: Project weekly plan

Hackathon I

Figure 24: First hackathon plan

During the first hackathon the main focus will be to implement the core functionality for characters in the game. This will include setting up Unity Navmesh and creating a base class as well as derivatives which will handle the logic involved with each character type. To achieve this, characters will be modelled in parallel to this process, this will allow for mechanics to be tested and visualised as they are worked on. The scripting will then continue into the following week, as work on prop models begins.

Hackathon 2

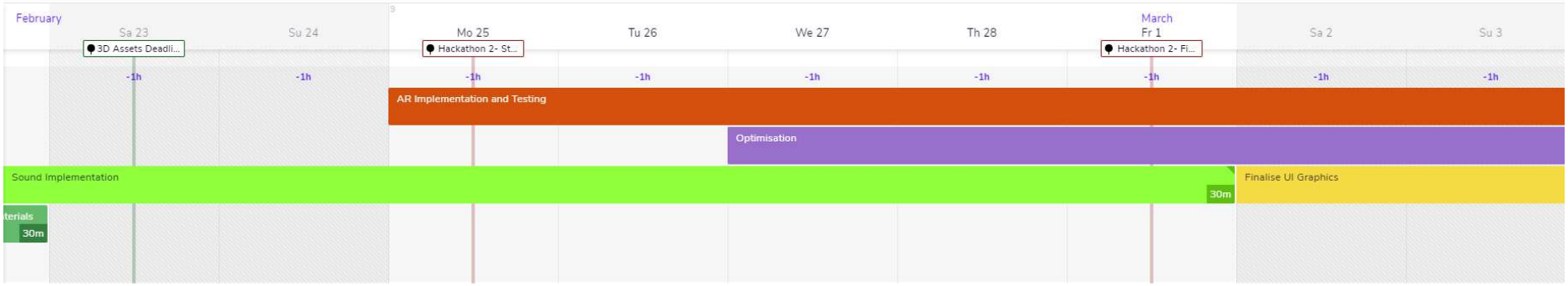


Figure 25: Second hackathon plan

The focus of the second hackathon will be to implement the game in AR and test it on mobile devices. The optimisation process for code and assets will then take place, with the goal of having the game run at 30 frames per second or more on select Android devices. The 3D asset deadline will occur prior to the second hackathon, this will ensure that performance test results are indicative of the final produc

Seminar Feedback Review

This section will cover feedback received during the prototype seminar, and improvements that will be made as a result of this.

Issues

A couple issues were brought up during the seminar. The first of which is that the navigation mesh agents (NavMesh Agents), would jitter on the spot when they reached their destination, this is because each of the characters destinations are set to the same vector position. This can be solved by increasing the stopping distance and/or radius properties, allowing the artificial intelligence (AI) to settle for not the exact destination vector, but a position far enough away to avoid agents from competing for a position.

Another concern was how the number of each type of unit will be handled. The player could be given the option to choose which character types to bring and how many, adding an element of meta strategy to the game. However, this could also cause frustration for players that do not understand the mechanics attached to each character type, with levels becoming impossible to complete without certain character types selected. A suggested solution to this was that this would become an unlockable feature, requiring players to successfully complete levels with a predetermined set of characters- before they are given the freedom to experiment with different combinations. The replaying of levels can then be incentivised by a score bonus for using less characters, creating more challenge.

Additions

The addition of new character types was brought up during the seminar, this is something that will be taken on board in the project as it will expand on the gameplay, creating more mechanics and providing more opportunities for levels outside of mass and gravity based physics interactions. For example, a character type that produces heat, allowing water to be boiled or to light fires.

The concept of a tutorial level was also brought up, due to the need for players to understand how each character can be used. This will need to be implemented as without an introduction to mechanics, players will have no understanding of what tools they have at their disposal, which could result in frustration. This tutorial level will introduce each character type one at a time, with small puzzles as the level progresses. By the end of the level the player will have at least one of each character type, and a final challenge will incorporate all mechanics. This should act to guide the player through the core mechanics by allowing them to experiment with them in isolation, rather than requiring them to read instructions.

Amendments

As a result of the prototype seminar, the following amendments will be made to the game:

1. NavMesh Agents will be adjusted to prevent jittering.
2. Additional character types will be designed and implemented.
3. Players will be able to unlock the ability to choose which characters to have in levels that they have already completed. With a score point bonus for using less.
4. A tutorial level will be designed and implemented, covering each character type and how their mechanics can be used.

GDD Review

During the GDD seminar a few pieces of feedback came up. Some sections of the document required clarification. Two main additions were mentioned. The first being the addition of pseudo code for the core mechanics. This would provide an understanding for how the logic in the game will be handled. Another addition that came up was the inclusion of legal issues, this would need to be done to explain what might come up with the use of assets that are created by other sources.

Justification

In this section of the document, decisions about the software that will be used in production will be justified. Legal issues and revenue generation will also be explored.

AR Platform

As the game will be running on mobile devices, using the devices camera to place the game environment on a surface, an AR platform would need to be used to be able to bring the digital and real environments together in real-time. A popular free AR platform called Vuforia (PTC, 2018) could be used to achieve this, however it requires the use of a marker to give the system a reference point for rendering a scene. This would cause issues as users would need to print out or otherwise be in possession of an item that the software can recognise through the camera. Another AR platform that could be used is called 8th Wall (2018), which does not require a visual marker and instead can be used to place scenes on flat surfaces such as the ground or desk. 8th Wall is also free to use and does not have a watermark. Resultantly it has been decided that 8th Wall will be used to develop the game's AR functionality.

Game Engine

To develop the game an engine will be required to handle base physics and rendering, acting as a platform for development. Unity (2018) is game engine that allows for fast prototyping with the use of prefabs and a component system, providing the ability to create modular objects. Unity if free to use, only costing money when revenue exceeds $100,000 USD. Unity supports AR, most notably 8th wall has an SDK specifically for Unity. It is able to build to various platforms including Android and IOS and the game can be tested within the software itself without the need to build. Unreal (Epic Games, 2018) is another engine that can be used to develop games for free. 8th wall does not have and SDK available for use on Unreal. Unreal is not as associated with mobile development, however it is able to produce games with high quality photo-realistic graphics. This is not necessarily beneficial for the project, as the game is designed for mobile, requiring real-time tracking through a camera. Therefore, if the game also had realistic graphics it would risk having system requirements that most mobile phones do not meet. Due to this, the game will be developed in Unity.

3D Software

There are various pieces of software available for 3D modelling and animating. Three that have been considered for the project are Autodesk Maya (Autodesk, 2018), Autodesk 3Ds Max (Autodesk, 2018) and Blender (Blender Foundation, 2018). Blender is a free open source 3D modelling software that allows for animation, sculpting and rendering. However, it has a plethora of features that give it a steep learning curve. Autodesk 3Ds Max can be used to do many of the same tasks as Blender, it also has a system of modifiers allowing for complex algorithmic changes to geometry saving time for the user. The software costs £1,740 per year or £216 monthly to be used commercially. Autodesk Maya also has many of the features of Blender and 3Ds Max, often being used for animation for film and games. Maya also has the same pricing as 3Ds Max. The Autodesk products have the additional benefit of being able to work together, with files often being cross compatible. It has been decided that 3Ds Max will be used to create the models for the game.

Sound Software

All sound effects and audio will either be sourced or produced in LMMS (Linux MultiMedia Studio, n.d.), Audacity (Audacity Team, n.d.) and BFXR (increpare, n.d.). LMMS is an open source DAW (Digital Audio Workstation) which allows for sequencing midi notes that can be played by digital synths or instrument samples. This will be used to create and transpose all character voices for the game, producing a consistent yet unique sound between characters. Audacity is also an open source piece of software, which can be used to record and/or modify sound files. This will be used to increase amplitude or adjust pitch of sound effects, or to record microphone input for sounds created physically. Finally, BFXR is a web-based application that can be used to create short sound effects. This would be useful for sounds used as feedback such as when a button is pressed in a menu or when a challenge is completed.

Legal Issues

Some consideration needs to be made about the legality of the game. Though all assets are planned to be created, some audio may be sourced. In that case, copyright laws will need to be considered to ensure that the audio is used legally. The priority would be to find audio that is licensed under creative commons, while abiding by usage restrictions based on the particular licence. Otherwise audio could also be purchased. The final game itself will be automatically protected under copyright laws, however registering trade mark protection could be considered later to protect the logos and branding. Applications for this take 4 months (gov.uk, n.d.).

Revenue Generation

The game will be initially sold on the Google Play store for Android. This will mean that it must abide by the requirements that Google Play puts forth. These include but are not limited to file size, user privacy and using minimal permissions. The file size limit is 4.1GB in total, which is broken down into 100MB for the main APK (Android Package) file and a maximum of 4GB in APK expansion files (Google, 2018). The game will be sold at £2 per unit, allowing for revenue without the need for using advertisements, which would hinder the experience and/or take up too much screen space. This price has been chosen as it is low enough for people who are interested in AR on the mobile platform, while not too expensive as to where users won't purchase it out of being unsure if they will get their money's worth.

References

8th Wall, 2018. *8th Wall | Augmented Reality.* [Online]
Available at: https://8thwall.com/
[Accessed 10 December 2018].

Android, n.d. *Android Mini Robot Revolution - Dead Zebra, Inc Shop.* [Online]
Available at: https://shop.deadzebra.com/android-mini-robot-revolution/
[Accessed 5 December 2018].

Audacity Team, n.d. *Audacity ® | Free, open source, cross-platform audio software for multi-track recording and editing..* [Online]
Available at: https://www.audacityteam.org/
[Accessed 10 December 2018].

Autodesk, 2018. *3ds Max | 3D Modelling, Animation & Rendering Software | Autodesk.* [Online]
Available at: https://www.autodesk.co.uk/products/3ds-max/overview
[Accessed 10 December 2018].

Autodesk, 2018. *Maya | Computer Animation & Modelling Software | Autodesk.* [Online]
Available at: https://www.autodesk.co.uk/products/maya/overview
[Accessed 10 December 2018].

Blender Foundation, 2018. *blender.org - Home of the Blender project - Free and Open 3D Creation Software.* [Online]
Available at: https://www.blender.org/
[Accessed 10 December 2018].

Epic Games, 2018. *What is Unreal Engine 4.* [Online]
Available at: https://www.unrealengine.com/en-US/what-is-unreal-engine-4
[Accessed 10 December 2018].

Google, 2018. *Core app quality | Android Developers.* [Online]
Available at: https://developer.android.com/docs/quality-guidelines/core-app-quality
[Accessed 10 December 2018].

gov.uk, n.d. *Intellectual property and your work.* [Online]
Available at: https://www.gov.uk/intellectual-property-an-overview/protect-your-intellectual-property
[Accessed 10 December 2018].

GuidoVrola, n.d. *Nanobot original concept 3D model.* [Art] (www.cgtrader.com).

Half-Life 2. 2004. [Game] United States of America: Valve Corporation.

increpare, n.d. *Bfxr. Make sound effects for your games..* [Online]
Available at: https://www.bfxr.net/
[Accessed 10 December 2018].

Larabie, R., n.d. *Neuropolitical Font . 1001 Fonts.* [Online]
Available at: https://www.1001fonts.com/neuropolitical-font.html
[Accessed 26 November 2018].

Linux MultiMedia Studio, n.d. *LMMS | Home.* [Online]
Available at: https://lmms.io/
[Accessed 10 December 2018].

Overlord. 2007. [Game] Directed by Lennart Sas. Netherlands: Triumph Studios.

Pikmin 2. 2004. [Game] Directed by Shigefumi Hino, Masamichi Abe. Japan: Nintendo EAD.

PTC, 2018. *Vuforia | Augmented Reality for the Industrial Enterprise.* [Online]
Available at: https://www.vuforia.com/
[Accessed 10 December 2018].

Robots. 2005. [Film] Directed by Chris Wedge. United States of America: Blue Sky Studios; 20th Century Fox Animation.

Sanctum 2. 2013. [Game] Sweden: Coffee Stain Studios.

SJRC, n.d. *SJRC S70W RC Drone WiFi FPV / Double GPS Module - $117.13 Free Shipping|GearBest.com.* [Online]
Available at: https://www.gearbest.com/rc-quadcopters/pp_1499656.html
[Accessed 5 December 2018].

Surgeon Simulator. 2013. [Game] Directed by Tom Jackson, Jack Good, Luke Williams, James Broadley. England: Bossa Studios.

Unity, 2018. *Unity.* [Online]
Available at: https://unity3d.com/
[Accessed 10 December 2018].