Team Grey Sprint Documentation

Repository **6**

Sprint 0 Overview: 02 Nov 2021 - 14 Nov 2021 **6**

Sprint 0 Meeting Minutes **6**
- Team Meeting 1: 02 Nov 2021 6
- Customer Meeting 1: 03 Nov 2021 7
- Team Meeting 2: 09 Nov 2021 7
- Customer Meeting 2: 10 Nov 2021 9

Sprint 0 Requirements List **11**
- Explanation of Changes to Requirements List 11
- Initial List Version 0 11

Sprint 0 Backlog **12**
- Complete Backlog Tasks 12

Sprint 0 Exception Handling **12**

Sprint 0: Storyboard Development **13**
- RTS Style Game Storyboard 13
- Choose Your Own Adventure Storyboard 16
- Adventure Platformer Storyboard 18

Sprint 0: Team Skills Document **22**

Sprint 0 Retrospective **23**

Sprint 1 Overview: 14 Nov 2021 - 21 Nov 2021 **24**

Sprint 1 Meeting Minutes **24**
- Team Meeting 3: 14 Nov 2021 24
- Team Meeting 4: 16 Nov 2021 25
- Customer Meeting 3: 17 Nov 2021 26
- Team Meeting 5: 17 Nov 2021 27
- Team Meeting 6: 19 Nov 2021 28
- Team Meeting 7: 21 Nov 2021 29

Sprint 1 Requirements List **29**
- Explanation of Changes to Requirements List 29
- Updated List Version 1 30

Sprint 1 Backlog **30**
- Complete Backlog Tasks 30

New Backlog Tasks 31

Sprint 1 Exception Handling 31

Sprint 1 Product Documents 32
User Stories 32
Development Tasks 36
Requirements Use Cases 37
Design Use Cases 38
Test Cases 39
CRC Cards 43
User Interface Design Elements 47
Design Element 1.1: Intro Screen Design Version 1 47
Design Element 1.2: Game Demo Shown in Customer Meeting 3 49

Sprint 1 Game Progress 50
Game mechanics progress 50

Sprint 1 Retrospective 50

Sprint 2 Overview: 21 Nov 2021 - 28 Nov 2021 51

Sprint 2 Meeting Minutes 51
Team Meeting 8: 22 Nov 2021 51
Customer Meeting 4: 24 Nov 2021 52
Team Meeting 9: 24 Nov 2021 53
Team Meeting 10: 26 Nov 2021 53
Team Meeting 11: 29 Nov 2021 55

Sprint 2 Requirements List 55
Explanation of Changes to Requirements List 55
Updated List Version 2 56

Sprint 2 Backlog 57
Complete Backlog Tasks 57
New Backlog Tasks 57

Sprint 2 Exception Handling 58

Sprint 2 Product Documents 59
User Stories 59
Revised User Stories - Final Set 64
Requirements Use Cases 64
Design Use Cases 65
Tests 66
CRC Cards 71

User Interface Design Elements 73
Design Element 2.1: 3D Model for Alien Enemy 73
Design Element 2.2: Intro Cutscene Video for Game Backstory 74
Design Element 2.3: Healer Character 76
Design Element 2.4: Buildings 77
Design Element 2.5: Level Select Screen Design (Version 1) 79

Sprint 2 Game Progress 80
Game mechanics progress 80

Sprint 2 Retrospective 81

Sprint 3 Overview: 28 Nov 2021 - 05 Dec 2021 82

Sprint 3 Meeting Minutes 82
Team Meeting 12: 01 Dec 2021 82
Team Meeting 13: 03 Dec 2021 84
Team Meeting 14: 07 Dec 2021 84

Sprint 3 Requirements List 85
Explanation of Changes to Requirements List 85
Updated List Version 3 85

Sprint 3 Backlog 86
Complete Backlog Tasks 86
New Backlog Tasks 86

Sprint 3 Exception Handling 87

Sprint 3 Product Documents 88
User Stories 88
Revised User Stories - Final Set 92
Requirements Use Cases 93
Design Use Cases 94
Tests 95
CRC Cards 98
User Interface Design Elements 100
Design Element 3.1: Intro Screen Design Version 2 100
Design Element 3.2: Level Select Screen Design Version 2 101
Design Element 3.3: Intro Cutscene Video for Game Backstory Version 2 102
Design Element 3.4: Level Design 103
Level 1: Sketch and Final 103
Level 2: Sketch and Final 104
Level 3: Sketch and Final 105
Level 4: Sketch and Final 106

Level 5: Sketch and Final 107
Level 6: Sketch and Final 108
Design Element 3.5: Cube Character Models 109
Archers 109
Healers 109
Knights 109
Warriors 110
Design Element 3.6: Table with Object Sats 111
Design Element 3.7: Lose Screen 112
Design Element 3.8: Win Screen Version 1 113

Sprint 3 Game Progress 113
Game Mechanics 114

Sprint 3 Retrospective 114

Sprint 4 Overview: 05 Dec 2021 - 12 Dec 2021 115

Sprint 4 Meeting Minutes 115
Customer Meeting 5: 08 Dec 2021 115
Team Meeting 15: 10 Dec 2021 115
Team Meeting 16: 12 Dec 2021 116

Sprint 4 Requirements List 116
Explanation of Changes to Requirements List 117
Updated List Version 4 117

Sprint 4 Backlog 118
Complete Backlog Tasks 118

Sprint 4 Exception Handling 118

Sprint 4 Product Documents 118
User Stories 119
Development Tasks 120
Requirements Use Cases 120
Design Use Cases 120
Tests 121
User Interface Design Elements 123
Design Element 4.1: Win/Lose Screens Version 2 123
Design Element 4.2: Character Intro Scenes 124
DE4.2.1: Level 1 Player Introduction Screen 124
DE4.2.2: Level 2 Player Introduction Screen 124
DE4.2.3: Level 3 Player Introduction Screen 125
DE4.2.4: Level 4 Player Introduction Screen 125

DE4.2.5: Level 5 Player Introduction Screen 126
DE4.2.6: Level 6 Player Introduction Screen 126

Sprint 4 Game Progress 126
Game mechanics progress 127

Sprint 4 Retrospective 127

Sprint 5 Overview: 12 Dec 2021 - 19 Dec 2021 128

Sprint 5 Meeting Minutes 128
Team Meeting 17: 13 Dec 2021 128
Customer Meeting 6: 15 Dec 2021 129
Team Meeting 18: 19 Dec 2021 129

Sprint 5 Requirements List 130
Final List Version 5 131

Sprint 5 Backlog 131
Complete Backlog Tasks 132

Sprint 5 Exception Handling 132

Sprint 5 Product Documents 132
User Stories 133
Development Tasks 135
Requirements Use Cases 135
Design Use Cases 136
Tests 136
User Interface Design Elements 138
Design Element 5.1: Level # Overlay 138
Design Element 5.2: Instruction Screen 139
Design Element 5.3: Pause Menu 140
Design Element 5.4: Controls Menu 141
Design Element 5.5: Level Select Screen Version 3 142
Design Element 5.6: Intro Cutscene Video for Game Backstory Version 3 142

Sprint 5 Game Progress 142
Game mechanics 143

Sprint 5 Retrospective 143

Repository

`git@github.com:wc641/Team-Grey-SE-CW2.git`

Sprint 0 Overview: 02 Nov 2021 - 14 Nov 2021

- Planning sprint: idea generation and project planning based on initial customer brief and Customer Meetings 1 and 2
- Brainstorm and develop game ideas to present to customer
- Create and refine storyboards for top game ideas based on customer feedback from Customer Meeting 1 (see Product Documents for Storyboard Development)
- Initial project planning and preparation

Sprint 0 Meeting Minutes

Team Meeting 1: 02 Nov 2021

Minutes: 13:00 - 13:30 via Microsoft Teams

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals

- Team member introductions
- Discussion of individual strengths and coding experience
- Discussion of project details and initial customer brief/ requirements
- Brainstorm game ideas and game development engines/languages

Action items

- Each team member to come up with 2 game ideas before Customer Meeting 1
- Each team member to think of questions to clarify customer's requirements at Customer Meeting 1

Customer Meeting 1: 03 Nov 2021

Minutes: 10:45 – 11:00 @ Chancellor's Building 5.13

Participants
- William Chambers
- Ashton Mitchell
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik
- Customer

Discussion topics
- Manage the expectation of implementing a "bot"
    - customer says a bot is not essential or relevant to all game ideas
- Presentation of all game ideas:
    - dungeon crawler
    - prisoner escape
    - quiz-based game
    - choose your own adventure
        - customer's favorite idea because of the potential for strategy by requiring player to create and update mental models of the game path
        - needs heavy plot and engaging story

Action items
- Decide on maximum of 3 game ideas to develop further
- Create storyboards for top game ideas to present to customer at next Customer Meeting

Team Meeting 2: 09 Nov 2021

Minutes: 15:00 - 17:30 @ Virgil Building

Participants
- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta via Microsoft Teams

Goals

- Get group set up with Confluence, Jira, and GitLab
- Develop 3 game ideas further and create storyboards to present at Customer Meeting 2

Discussion topics

Topic	Notes
Game Ideas	● Choose Your Own Adventure: ○ Issues with reliance on art/visuals, story could get too complex ○ Discussed how to manage complexity by modelling after Undertale (i.e. using pixel art and keeping story possibilities from branching too far out) ● RTS Game: ○ new idea brought forward ○ strategize to build a team of heroes that will spawn with the aim to destroy a base ○ emphasis on strategy to satisfy customer requirements ● Platformer: ○ new idea brought forward ○ focus on creating a good story to engage customer ○ cross with adventure-style game
Storyboards	● Discuss what to include in our storyboards, including text descriptions ● Draft storyboards in sub-teams: Choose Your Own Adventure (Vlad & Ani), RTS Game (Will & Ashton), Platformer (George & Dinil) ● Create final version storyboards to present to customer
Project Tools	● Decision on which tools to use ● Setup and familiarization with Jira, Confluence, and GitLab

Action items

- William Chambers to send out links for Team Jira and Confluence
- Each team member to join Team Jira, Confluence, and familiarize self with GitLab
- Ashton Mitchell to record and upload meeting notes on Confluence
- Each sub-team to develop their assigned game idea into a storyboard for presentation to the customer
- Practice game idea presentations for upcoming Customer Meeting 2

Decisions

- Narrowed down best game ideas to 3 distinct styles: Choose Your Own Adventure, RTS, and Platformer
- Chose project management, documentation, and collaboration tools

Customer Meeting 2: 10 Nov 2021

Minutes: 11:30 – 11:45 @ Chancellor's Building 5.13

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta

Goals

- Present storyboards for 3 game ideas of distinct gameplays (choose your own adventure, RTS, platformer)
- Ask customer to select his favorite game idea
- Get clarification on duration of gameplay and any further questions

Discussion Topics

Game Idea	Presenter	Notes
Choose Your Own Adventure	Ani	● Went through the story behind our game: An engineer is a lone survivor on a spaceship that's been invaded by aliens. ● Explained what the player's mission would be: Choose options that lead to him either defeating/evading the aliens or hiding/following their orders. Every choice could either progress their game by giving them some kind of advantage or set them back ● The player would win if they managed to get back to earth.
RTS	Will	● Went through the story behind our game: TLDR; Dragons attack town and heroes are sent to destroy the dragons ● Explained the mechanism behind the game: Real time strategy - Choosing a squad of heroes that are capable of breaking down the dragon's defenses and destroying the dragon egg, balancing hero abilities with inventory/slot space. ● Player wins if they can make it through the dragon's lair (each level) and destroy the dragon king
Adventure Platformer	George	● Presented the idea of a 2D adventure platformer. ● The player controls a software engineer who falls down a waterfall and must traverse through a dungeon in order to gain the "agile" ability which will enable them to escape back up the waterfall to the office before they are missed by their boss. ● Showed the customer a storyboard which demonstrated some level ideas and described an outline of the story.

Action items

- Ani Udom to set up collaborative document for everyone to input their individual skills, experience, and preferences for project roles and tasks
- Ashton Mitchell to upload team and customer meeting notes to Confluence
- George Griffiths to upload audio recording to Customer Meeting 2

- Each sub-team to upload their game idea storyboards (from brainstorming notes to final document presented to customer)
- Each game idea presenter (William Chambers, Ani Udom, George Griffiths) to include notes within the Customer Meeting 2 page on their presentation to the customer
- Everyone to research and get familiar with potential game engines before next Team Meeting

Decisions

- Customer preferred RTS game idea
- Team decided to pursue RTS game idea
- Team decided to use space theme for RTS game

Sprint 0 Requirements List

Explanation of Changes to Requirements List

- Based on the initial brief provided by the customer, R1.1 - R1.6 were created.
- Based on Customer Meeting 1, R1.4 was removed. The customer clarified that R1.4 is only necessary for certain game designs and has said that the proposed game design does not require the implementation of a “bot”.
- Based on Customer Meeting 1 and 2, the customer emphasized the desire for a mentally engaging game with a high level of strategy leading to the creation of R1.7.
- Based on Customer Meeting 2, R1.1 was removed. The customer clarified that R1.1 was not a strict theme to follow and authorized creative freedom in the theme/story of the game.

Initial List Version 0

As each requirement is met, the corresponding test or design element will be listed in parentheses at the end of the requirement.

Key: ~~Strikethrough~~ indicates removed requirement

R1.1: ~~Dungeon themed game~~
R1.2: Game that will run on a laptop
R1.3: Game that will run on a phone
R1.4: ~~A “bot” that can play the game on it’s own or against other bots~~
R1.5: Ability to understand how well the player is doing in the game
R1.6: Game that has increasing difficulty through progression
R1.7: Gameplay needs to prioritize strategy

Sprint 0 Backlog

B0.1: Brainstorm game ideas
B0.2: Develop storyboards for the 3 best game ideas
B0.3: Set up Team boards on Jira and Confluence

Complete Backlog Tasks

B0.1: Brainstorm game ideas
B0.2: Develop storyboards for the 3 best game ideas (See Storyboard Development)
B0.3: Set up Team boards on Jira and Confluence

Sprint 0 Exception Handling

- After analysis of Customer Meeting 1, the team thought we would be pursuing a game in the style of a choose your own adventure, however, after presenting storyboards at Customer Meeting 2, the customer changed preference to the RTS style game
- See Customer Meeting 1 and 2 Notes for analysis of the customer's input on the presented game ideas
- See the Storyboard Development for a comparison between the original choose your own adventure style game and the RTS style which became the finalized idea to pursue

Sprint 0: Storyboard Development

RTS Style Game Storyboard

Game Description:

Title: Duel of Dragons
Story: Dragons have long ruled over the Townspeople. After another fiery, deadly attack from the Dragons, the Townspeople rise up and fight back to eradicate the Dragons from their land. Four groups of heroes emerge, Knights, Archers, Strong Men, and Wizards. They enter the Dragon's lair, first fighting to destroy the Dragon eggs. As they venture deeper and deeper into the lair, the heroes are faced with the final boss, the Dragon King.

Characters: "Heroes"

- Knight:
    - attack and health middle range, dependent on attack style from close up
    - takes up least "inventory" space
    - acquired character from start of game
- Archer:
    - attack and health middle range, dependent on attack style from distance
    - takes up one more "inventory" space than Knight
    - acquired character from start of game
- Wizard:
    - high attack, low health
    - takes up more "inventory" space than Archer or Knight
    - unlock character as game progresses
- Strong Man:
    - high attack, high health
    - takes up most "inventory" space
    - unlock character as game progresses

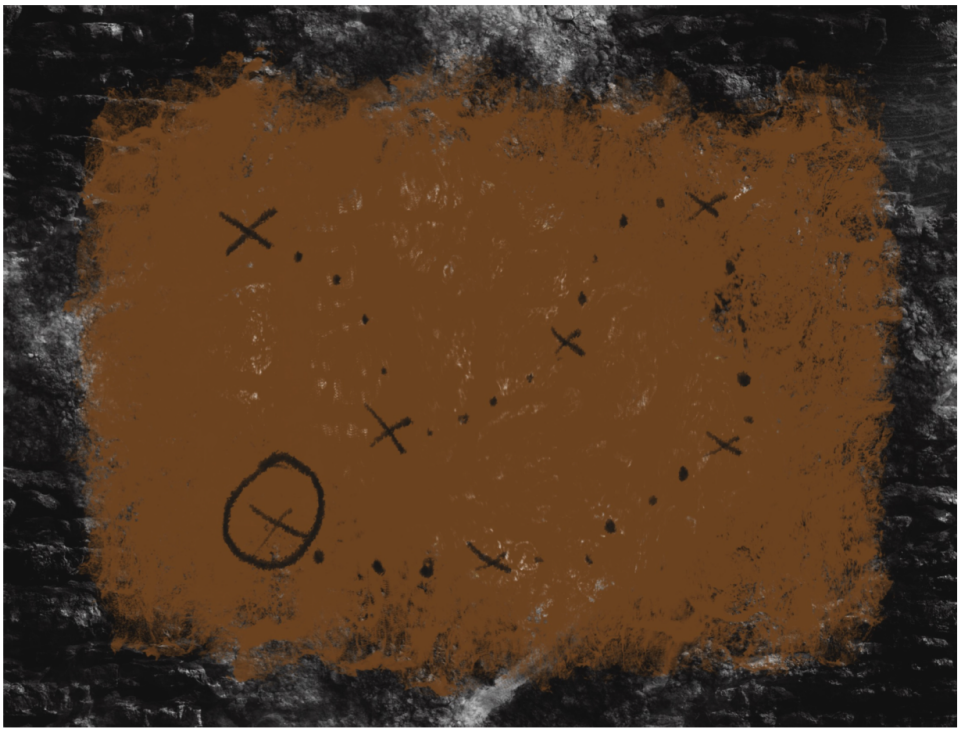
Real-Time Strategy Gameplay:

- Player must balance quantity with quality of chosen heroes, as each hero has different strengths and weaknesses (i.e. health, attack, different amount of "inventory")
- Increasing difficulty in levels
- Percentage of damage inflicted on the dragon's defenses must reach 100% for the heroes to destroy the egg or dragon at the center of each level
- Progress through levels but if the player fails a level, 1 out of 3 overall lives is lost and they can re-try the current level. If the player fails 3 levels, then they are forced to start back at Level 1.

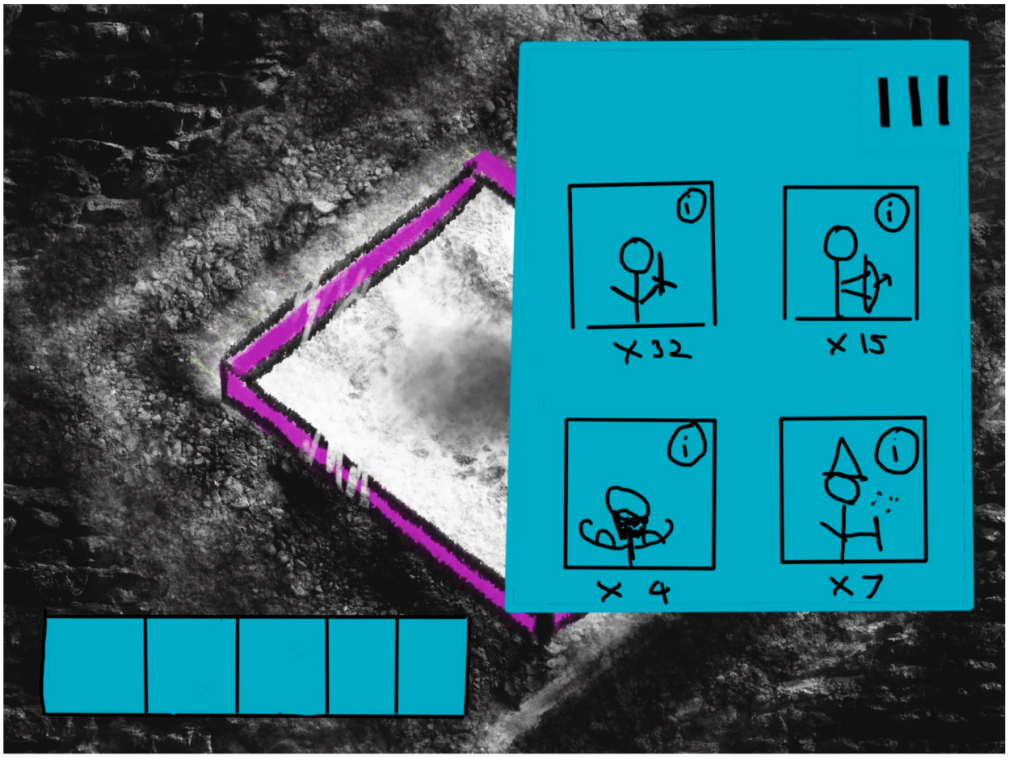
Game Design:

Intro: dragon destroying town

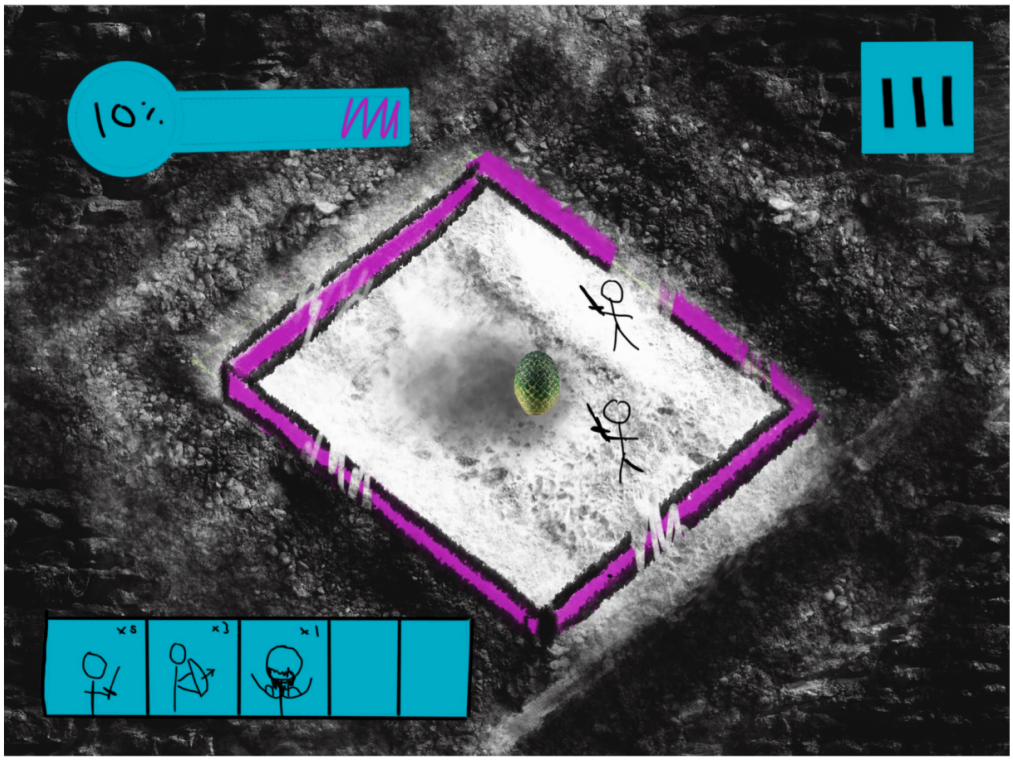
map of egg locations in dungeon



Hero selection screen



battle



Choose Your Own Adventure Storyboard

Intro:
You're the lead engineer of a research vessel sent to Mars 10 years ago.
You're finally on your way back to earth, excited to reunite with your loving wife and children.
Mostly, you cannot wait for a chilled pint at your favourite pub just around the corner.
new page
It has been a month since you've departed Mars. The ship has been very faulty ever since and you have been deeply overwhelmed with work.

Screen 1

You've just woken up from a long sleep after making some fixes to the fuelling matrix. Your normal routine after waking up is going for a hot mug of coffee. But something is not feeling right today....

Go to the kitchen

Go to the common room to catch up with your crewmates

Start the day with taking a quick shower

Selected option

Screen 2

After you take the shower, you decide to shave, but discover that you don't have a razor.

Go ask your friend down the corridor for a razor

Selected option

Don't bother

If this option is selected, go back to *Screen 2*, but only 2 option remain, excluding the one picked before

Screen 3

Your friend is not in, so you go inside and grab the razor yourself, invading your friend's personal space (you've known them for long enough).
On your way back to your room, you hear a scream and a loud thud from the common room.

Ignore it, go shave that beard off

If this option is selected, go back to *Screen 2*, but only 2 option remain, excluding the one picked before

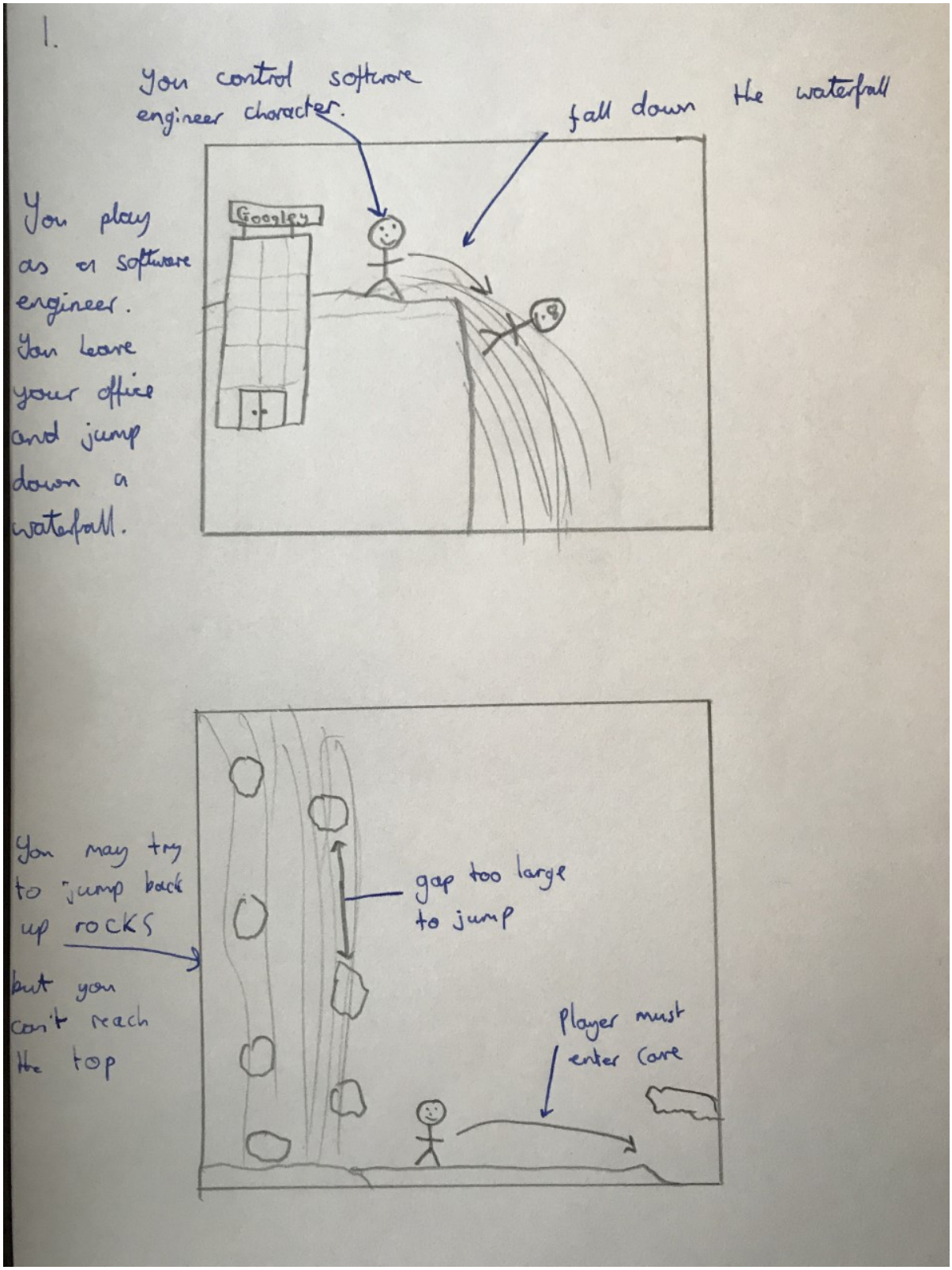
Check it out, make sure everything's okay

Screen 4

Soundtrack: techno, synthwave

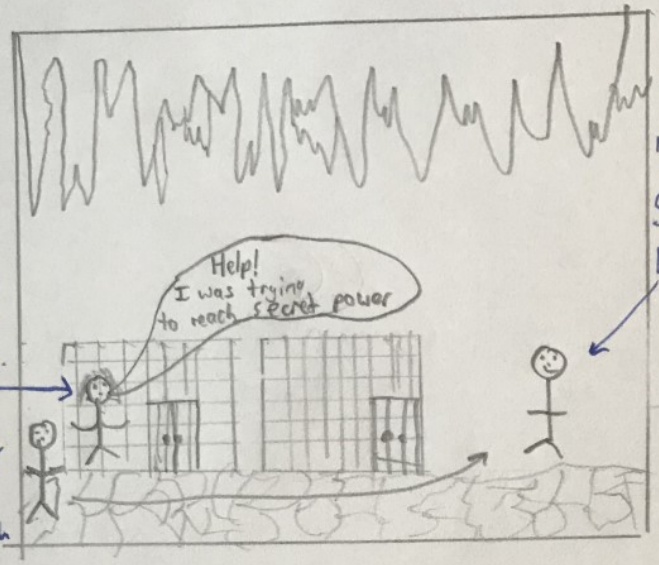
For escape route: add a progress bar that shows the likelihood of the MC escaping

Adventure Platformer Storyboard

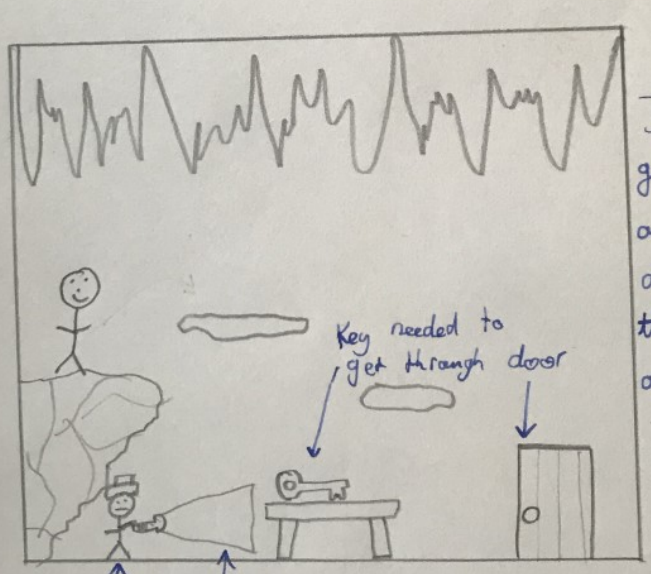


2.

Find yourself in a dungeon with a man resembling 'Julian Assange' locked in a cell. He talks of a secret power ahead he was trying to reach



Walk to the right to continue. Side scrolling level.

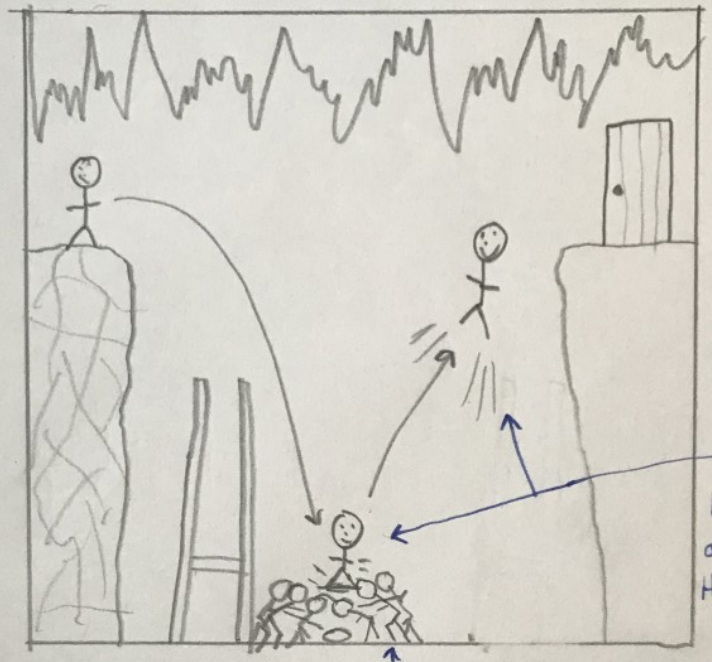


Jump down to get key from table and open the door at the end of the room while avoiding the guard.

a bot which is a dungeon guard who walks side to side

bot has a field of vision from his torch which you must avoid walking in

3.



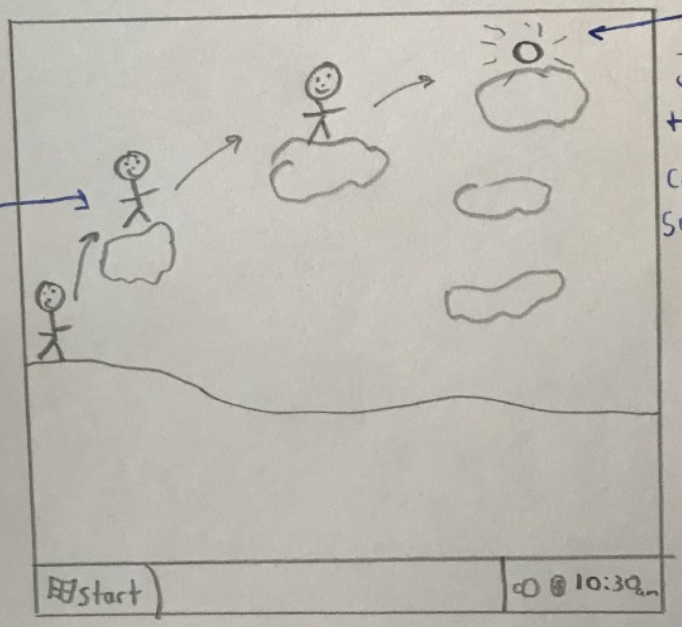
Plyou must time your jump to land on the scrum and bounce to the door platform.

a scrum of rugby players (bot) that moves side to side

"agile super power"

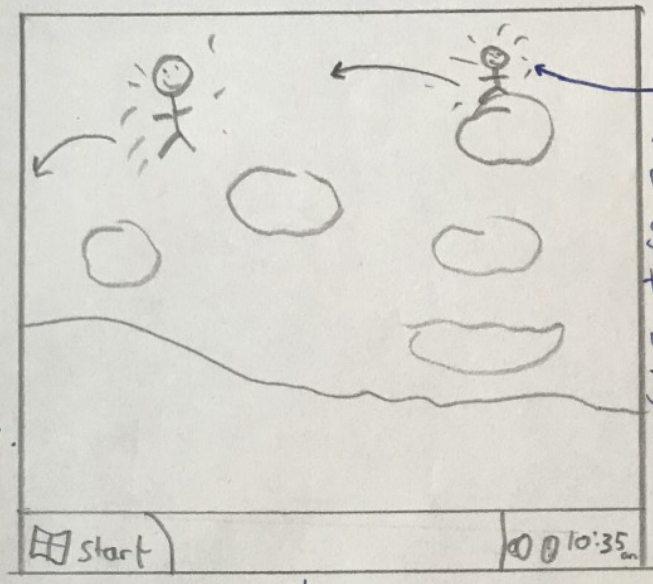
Jump across the clouds to collect the super power "agile"

Skillful platforming over clouds to reach "agile" power.



4.

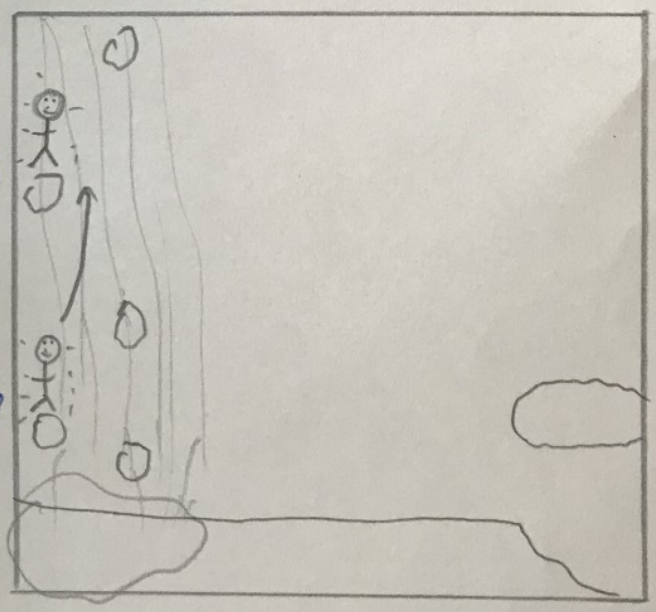
Begin to back track your way through the Levels with the "agile" power.



The agile super power causes you to glow and have the ability to walk faster and jump higher.

with agile ability easily/more easily backtrack through stages

Using the agile power jump back up waterfall to return to work in the office.



You can now jump high enough to escape back up the waterfall.

Sprint 0: Team Skills Document

- Created to give insight into each team member's previous experience and interests for the assigning of roles and tasks in the project
- See image below of Team Skills Document created on Team Confluence page:

Team Skills

AU VG WC +5

Created by Ani Udom
Last updated: Nov 14, 2021 by Vinay Kumar Gupta • 1 min read

Member	Undergrad Degree	Skills/Experience (coding, design, etc...)	Role you would like to play (don't need to have experience)
Ani	Economics	Databases (SQL)	graphic/ui design (don't have experience but happy to try), project management?
Ashton	Psychology	Python, Java (beginner), C (beginner)	art, user experience and design, documentation
George	Biology	I have little coding experience but I have lots of experience in writing and recording music, some experience in 2d and 3d animation and a little experience using unreal engine 4.	Write some code especially to do with implementing a bot. Compose and record music. Write story and help with animations.
Will	Audio Engineering	Two years in industry as audio software developer; C++, Scripting (bash, batch), small amount of python.	Software engineer, potentially sound design and or music.

Dinil	Mathematics	C++ made a raytracer once. Python (Beginner). JavaScript(Beginner)	Writing (No experience). Music. Level/Game Design(No experience), Willing to try Art.
Vinay	Textile Engineering	JAVA - (Intermediate) Python, C, C++ (beginner)	Coding and srum master role.
Yash	Electronics and Telecommunication	Python(intermediate), SQL (intermediate), Java (Beginner)	Coding, Project Management
Vlad	Mechanical Engineering	C# (intermediate), Python (intermediate), C (beginner), Unity (experienced beginner), design - more from mech eng perspective (CAD, etc.), Adobe Photoshop/Premiere Pro	In order of preference and ability: 1. Coding (from game mechanics to implementing graphics and UI) 2. Character/level mechanics design 3. Character/level/UI graphics design Skills in Photoshop and Premiere Pro as well

Sprint 0 Retrospective

- Generated wide range of game ideas encompassing distinct gameplay styles (see Team Meeting 1 and 2 notes)
- Presented five ideas to customer and based on feedback narrowed down to one leading idea, but also produced two new ideas to fit customer's updated requirements (see Customer Meeting 1 notes and Sprint 0 Requirements List)
- Created storyboards for each of the three final game ideas (see Storyboard Development)
- Developed good team working dynamic through group brainstorming and idea development

Sprint 1 Overview: 14 Nov 2021 - 21 Nov 2021

At the start of this sprint we set out to accomplish the following:

- Develop story for game
- Create intro cut-scene and start screen
- Flesh out game environment
- Create demo for first level to show customer

Sprint 1 Meeting Minutes

Team Meeting 3: 14 Nov 2021

Minutes: 17:30 - 19:15 via Microsoft Teams

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals

- Sprint 0 review
- Sprint 1 planning

Discussion Topics

- Review of progress made during Sprint 0
    - Discussion of improvements to project management in sprints going forward
- Unity
    - group consensus to use Unity for game development
    - helpful Youtube links to be shared to group via MS Teams
        - Ani Udom found a unity tutorial
        - Vladislav Shishkin found a unity tutorial
        - Dinil Wickramasinghe found a pathfinding AI video
- Discussion of Team Skills Document and assigning of roles based on individual strengths
- Set up of Jira page for project management
    - Creation of backlog
    - Creation of Sprint 1 including pulling tasks to complete from backlog and assigning to team members

Action items

- Post helpful video links to MS Teams
- Refine Jira page
- Arrange next Team Meeting before the Customer Meeting 3 on November 17, 2021

Team Meeting 4: 16 Nov 2021

Minutes: 15:15 - 16:15 @ Computer Lab 2.56

Participants

- Ashton Mitchell
- Dinil Wickramasinghe
- William Chambers
- Yash Malik
- Ani Udom
- Vladislav Shishkin via Teams
- George Griffiths via Teams
- Vinay Kumar Gupta via Teams

Goals

- What each team member has accomplished so far in Sprint 1
- Team preparation of what to present and discuss with customer at Customer Meeting 3

Discussion Topics

- Preparation for what to present to customer at Customer Meeting 3
    - Present thorough run through of game:

1. Intro screen (option to start game or view instructions)
2. Introduction scene
    a. backstory and development of player's motivation to play game
3. Map outlining levels needed to get through to win game
4. Choose level (all levels locked except first level)
    a. player able to view level and then use slide-out panel to select army based on the level design
    b. different screens after success or failure of each level (e.g. "Mission Accomplished" or "Enemy Capture, Retry?")
5. Gameplay strategy:
    a. Limit on army size based on overall "strength" - different characters take up different amount of "strength" in army
    b. Set parameters:
        i. each level will have certain combinations of characters that will result in success and other combinations that will result in failure

        ii. each defence/obstacle will have specific features which are best attacked by certain characters, player must take this into account when building army
    c. Increasing strategy
        i. first three levels easier to get player familiar with gameplay
        ii. less army combinations that will result in success as levels progress
        iii. number of levels depending on what can be implemented/kept interesting - starting point to achieve 8-10 levels
        iv. every 2 levels progressed, the player unlocks a new character and new game level design

Action Items

- Make intro screen
- Other team members to add to description of game run through
- Think of names for the game
- Develop backstory for the game to present. Build off ideas from Ani Udom and George Griffiths
- Ani Udom to collect space themed art/pictures for use in game
- William Chambers to host brief tutorial of Git for team after Customer Meeting 3
- Meet after Customer Meeting 3 to divide into pairs and assign each pair coding tasks

Customer Meeting 3: 17 Nov 2021

Minutes: 11:30 – 11:45 @ Chancellor's Building 5.13

Participants

- William Chambers
- Ashton Mitchell
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik
- Customer

Discussion topics

- Run through game concept and design as outlined in Team Meeting 4
- Show customer a demo of character and enemy movements (see Product Documents for images)

Action items

- Make an intro screen and one working level
- Ensure our documentation from Sprint 0 is in order and submittable

Team Meeting 5: 17 Nov 2021

Minutes: 12:30 - 13:45 @ Computer Lab 2.56

Participants

- Ani Udom
- William Chambers
- George Griffiths
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Vladislav Shishkin
- Yash Malik
- Ashton Mitchell Via Microsoft Teams

Goals

- Agree on Game design
- Git Demo
- Unity Demo
- Decide features to present at next customer meeting

Discussion Topics

- Defining how the game will work (so everyone is in the same page)
- Do we want to have enemies if so are there different types?
- Ways to define the total number of characters that can be on the board
- Camera angles for game

Decisions

- Base in the centre and drop zone around
- Characters destroy structures or fight enemy
- Some enemies stationary (initial can make some move in later iterations)
- Characters are on map and you decide which ones you want

Action Items

- Create Start scene (with play button) and map scene: Dinil Wickramasinghe and Yash Malik
- Write game story and Create intro scene: George Griffiths
- Work on game mechanics: Vladislav Shishkin
- Work on enemy for the Knight characters: Vinay Kumar Gupta
- Work on Camera Mechanics: William Chambers

Team Meeting 6: 19 Nov 2021

Minutes: 14:00 - 15:20 @ Computer Lab 2.56

Participants

- Ashton Mitchell via Microsoft Teams
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin via Microsoft Teams
- Dinil Wickramasinghe via Microsoft Teams
- Yash Malik via Microsoft Teams
- Vinay Kumar Gupta via Microsoft Teams

Goals

- Get team on track to meet first deliverable deadline
- Address product blockers:
    - game mechanics
    - team divisions

Discussion Topics

- Blocker: division in team over complexity of game mechanics and feasibility to build and implement additional features
    - open discussion of each team member's feelings about current direction including talking through doubts
    - team divided over game direction
- Progress made so far on game may be out of scope of team's coding abilities within the timeline
- Blocker: game mechanics
    - Solution: idea proposed to develop game mechanics and game UI in parallel:
        - Game UI: strip back game idea to viable deliverable (from start screen to end screen) by replacing game mechanics with "battle commenced" cut scene
        - Game mechanics: traditional RTS whereby the player is able to direct their spawned characters to an enemy to attack
            - Have a couple of team member's dedicated to testing and debugging game mechanics
    - This way the team will have a complete product to deliver with "placeholders" for the game mechanics to be input once they are able to be developed
- Run through of game mechanics achieved so far: up to characters spawning, being directed to an active enemy, and once in the field of view of an active enemy, combat commences
- Need to develop and implement static enemies (i.e. torrets) that can fire projectiles

Decisions

- After open and productive discussion, the team agreed on dividing efforts to produce game UI in parallel to game mechanics in order to have more complete “deliverable” by next Customer Meeting
- Team voted to pursue current RTS style game mechanics instead of switching to game mechanics modelled after Clash of Clans. Vote was split 4-4 but came to a consensus to progress with the RTS style that had already been started.
- Discussed character design and decided to use pre-fab characters
- Deliverable to be completed by end of Sprint 1:
    - Start menu with options and music with change to intro backstory video and change to level select screen with 1 playable level (Vladislav Shishkin, George Griffiths, Dinil Wickramasinghe, Vinay Kumar Gupta, Yash Malik, and William Chambers)
    - Submittable documentation completed by Ani Udom and Ashton Mitchell
- Sprint 1 retrospective to be held on Sunday after 5pm via Teams

Team Meeting 7: 21 Nov 2021

Minutes: 18:00 - 18:40 via Microsoft Teams

Participants
- Ani Udom
- Ashton Mitchell
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Vladislav Shishkin
- William Chambers
- Yash Malik

Goals
- Discuss how sprint 1 went and how we can improve for sprint 2 (see Sprint 1 Retrospective)

Sprint 1 Requirements List

Explanation of Changes to Requirements List

- In Team Meetings 6 and 7, evaluations of what could be achieved within the timeline constraints were discussed. This led to R1.3 being removed.
- R1.7 was updated to reflect the specific strategic gameplay (RTS) the team decided to implement in Team Meeting 7.
- R2.1-R2.9 were pitched to the customer in Customer Meeting 3 and confirmed as new requirements.

Updated List Version 1

As each requirement is met, the corresponding test or design element will be listed in parentheses at the end of the requirement.

Key: ~~Strikethrough~~ indicates removed requirement

R1.1: ~~Dungeon themed game~~
R1.2: Game that will run on a laptop
R1.3: ~~Game that will run on a phone~~
R1.4: ~~A "bot" that can play the game on it's own or against other bots~~
R1.5: Ability to understand how well the player is doing in the game (see DE1.2)
R1.6: Game that has increasing difficulty through progression
R1.7: Gameplay modelled after real-time strategy (RTS) (see RTS Style Game Storyboard)
R2.1: Character selection feature as part of gameplay strategy
R2.2: Intro screen with title, "START", and "OPTIONS" buttons (see DE1.1)
R2.3: Cut scene to establish game backstory
R2.4: Level selection functionality
R2.5: 8-10 levels
R2.6: Camera zoom functionality
R2.7: Enemy characters for player to battle against (see DE1.2)
R2.8: Fully animated character models
R2.9: Game music throughout to enhance story and theme

Sprint 1 Backlog

B1.1: Create unity repo (US1-1)
B1.2: Test path-finding AI (US1-4)
B1.3: Write game backstory (US1-3)
B1.4: Unity Demo (US1-2)
B1.5: Create animated character (US1-8, 1-9, 1-10)
B1.6: Intro cut-scene
B1.7: Demonstrate git/ gitlab (US1-2)
B1.8: Create ground (US1-6)
B1.9: Design intro screen (US1-7)

Complete Backlog Tasks

B1.1: Create unity repo
B1.3: Write game backstory
B1.10: Intro cut-scene (DE2.2)
B1.9: Design intro screen (DE1.1)
B1.4: Unity Demo
B1.5: Create animated character

B1.2: Test path-finding AI
B1.7: Demonstrate git/ gitlab

New Backlog Tasks

B1.10: Intro cut-scene

Sprint 1 Exception Handling

- Halfway through sprint 1 the team realised there was some confusion surrounding how the game mechanics were to function. To ensure everyone was on the same page, a meeting was held to address the game mechanics (discussion outlined in Team Meeting 6).
    - This issue pointed out flaws in our communication and enabled the team to identify this as an area to work on moving forward. Once the team was able to come to a decision the project was able to carry on as planned.

Sprint 1 Product Documents

User Stories

- First set of user stories based on requirements determined during first customer and team meetings (see Sprint 1 Requirements List)

User Story 1-1: Create Unity Repo

AS A Developer *I WANT TO* Create a Unity Repo *SO THAT* All developers in the team can work on the code at the same time as well as have access to all versions of the code.	*ACCEPTANCE CRITERION:* The repo works and all members of the team have access to it

User Story 1-2: Demonstrate GitLab and Unity

AS A Developer *I WANT TO* Have a demonstration of Gitlab and Unity *SO THAT* I understand how to use them efficiently and start working on the code.	*Acceptance Criterion:* All developers have a basic understanding of how to use them and can begin writing code

User Story 1-3.1: Write Game Backstory

AS A Developer *I WANT TO* Create a backstory for the game *SO THAT* The graphics team can begin designing the game and the coders can begin creating it.	*Acceptance Criterion:* All relevant team members understand the narrative of the game

User Story 1-3.2: Game Backstory

AS A Customer/Player *I WANT TO* Have a game with an interesting backstory *SO THAT* The users are engaged and want to keep playing the game	*Acceptance Criterion:* Story is approved by customer

User Story 1-4: Test Pathfinding AI

AS A Developer *I WANT TO* Test how to create a pathfinding AI *SO THAT* The team can decide if the game design is feasible	*Acceptance Criterion:* Able to create a working pathfinding AI on at least one character in the game

User Story 1-5: Create One-story Wall

AS A Developer *I WANT TO* Create a one-story wall *SO THAT* We can begin to test how characters will interact with buildings in the game	*Acceptance Criterion:* A simple wall is created that other objects in the game can interact with

User Story 1-6.1: Create Ground

AS A Developer *I WANT TO* Create ground *SO THAT* The foundations of the game will start to form, and other game objects will have a space to be placed	*Acceptance Criterion:* A plane is created that other objects in the game will move and be placed on

User Story 1-6.2: Ground

AS A Player *I WANT TO* Have ground in the game *SO THAT* There is a plane in which I can move my fighters to attack enemies	*Acceptance Criterion:* A plane is created that other objects in the game will move and be placed on

User Story 1-7: Design an Intro Screen

AS A Player *I WANT TO* Have an intro screen *SO THAT* I can get a feel for the game and have a way to start the game	*Acceptance Criterion:* Aesthetic intro screen that shows the theme of the game *Comments:* A working intro screen with functional buttons will be developed in Sprint 2

User Story 1-8: Create Knight Character

AS A Player *I WANT TO* Have a Knight character *SO THAT* I can fight enemies and damage some buildings	*Acceptance Criterion:* A character that is able to fight enemy characters and help destroy enemy buildings

User Story 1-9: Create Warrior Character

AS A Player *I WANT TO* Have a Warrior character *SO THAT* I can destroy enemy buildings	*Acceptance criterion* A character that is good at destroying enemy buildings *Comments:* Can attack enemies as well

User Story 1-10: Create Archer Character

AS A Player *I WANT TO* Have an Archer character *SO THAT* I can fight enemies and attack buildings from a long range	*Acceptance Criterion:* A player that shoots some sort of projectile and causes harm to enemy characters and buildings from a long range

Development Tasks

T1.1 for US1-7: Design Intro Screen
The screen should:

- Have a start button
- Have an exit button
- A button where players can change settings
- Be visually appealing, fonts should be a good size and easily read

T1.2 for US1-8: Knight character

- Create a base character that can attack other game objects and be attacked
- This should be the first character created

T1.3 for US1-9: Warrior Character

- This character can be built off T1.2
- The character should have a greater attack level and health than the Knight character

T1.4 for US1-10: Archer character

- This character can be built off T1.2
- The character should be able to attack objects from a long range
- Should shoot some sort of projectile

Requirements Use Cases

RU1.1 Use Case for US1-8: Knight character
Scope: Game Level
Level: user goal
Context: Players can select a Knight character which attacks enemies and destroys buildings
Frequency of occurrence: Any time a Knight character is chosen in a level
Open Issues: How much space do Knights take up in army selection?
Knight health level? Knight attack power?

RU1.2 Use Case for US1-9: Warrior character
Scope: Game level
Level: user goal
Context: Players can select a Warrior character which is better at destroying buildings and attacking enemies than the Knight.
Frequency of Occurrence: Any time a Warrior character is chosen in a level
Open Issues: How much space do Warrior characters take up in army selection?
Health Level? Attack power? What level is this character first introduced?

RU1.3 Use Case for US1-10: Archer character
Scope: Game level
Level: user goal
Context: Players can select Archers in theri army which can attack enemies and destroy buildings from a long range
Frequency of Occurrence: Any time an Archer is chosen in a game level.
Open Issues: How much space do Archer characters take up in army selection?
Health Level? Attack power? What level is this character first introduced?

Design Use Cases

Use Case 1.1:
UC1-1: **Use case:** Knight character
UC1-2: **Author:** AU
UC1-3: **Purpose:** Have a base character that fights enemies and can causes damage to buildings
UC1-4: **Overview:** Starts when the player selects 1 or more Knights for their army.
If their army is not full (based on total army strength allowance parameter), **alternative 1:** the player can add more characters. **Alternative 2:** character selection is complete and the level starts.
As the level begins the Knight will begin to, **alternative 1:** fight the enemy characters, **alternative 2:** destroy buildings
Knights have a certain health level and this depletes as enemies attack them, **alternative 1:** Healter character can restore some of the Knights health, if selected in the army. **Alternative 2:** No Healer characters available and the Knights health fully depletes. **Alternative 3:** No Healer character available but Knight is able to defeat enemy.
If all player characters die before enemy is defeated, then the player loses the level
UC1-5: **Cross References:** RU1.1, US1-8, US1-5, US2-11, UC2.2, UC2.3
UC1-6: **Actors:** Player
UC1-7: **Pre Condition:** The player has select Knights in their army

Use Case 1.2:
UC2-1: **Use case:** Warrior Character
UC2-2: **Author:** AU
UC2-3: **Purpose:** Have a character that is better at destroying buildings and enemies than the Knight character
UC2-4: **Overview:** Starts when the player selects 1 or more Warrior characters for their army.
If their army is not full (based on total army strength allowance parameter), **alternative 1:** the player can add more characters. **Alternative 2:** character selection is complete and the level starts.
As the level begins the Warrior will begin to, **alternative 1:** destroy buildings **alternative 2:** fight enemies
Strong men have a certain health level (they have more health than Knights) and this depletes as enemies attack them, **alternative 1:** Healer character can restore some of the Warrior's health, if selected in the army. **Alternative 2:** No Healer characters available and the Warrior's health fully depletes. **Alternative 3:** No Healer character available but the Warrior is able to defeat the enemy.
If all player characters die before enemy is defeated, then the player loses the level
UC2-5: **Cross References:** RU1.2, US1-8, US1-9, US1-5, US2-11, UC2.2, UC2.3
UC2-6: **Actors:** Player

UC2-7: **Pre Condition:** The player has unlocked a level where the Warrior character has been introduced

Use Case 1.3:
UC3-1: **Use case:** Archer Character
UC3-2: **Author:** AU
UC3-3: **Purpose:** Have a character that is able to destroy buildings and fights enemies from a long range using projectiles.
UC3-4: **Overview:** Starts when the player selects 1 or more Archers for their army.
If their army is not full (based on total army strength allowance parameter), **alternative 1:** the player can add more characters. **Alternative 2:** character selection is complete and the level starts.
As the level begins the Archer will begin to, **alternative 1:** destroy buildings **alternative 2:** fight enemies
Archers have a certain health level (they have less health Knights and Strong men) and this depletes as enemies attack them, **alternative 1:** Healer character can restore some of the Archers health, if selected in the army. **Alternative 2:** No Healer characters available and the Archer's health fully depletes. **Alternative 3:** No Healer character available but the Archer is able to defeat the enemy.
If all player characters die before enemy is defeated, then the player loses the level
UC3-5: **Cross References:** RU1.3, US1-9, US1-10, US1-8, US1-5, US2-11, UC2.2, UC2.3
UC3-6: **Actors:** Player
UC3-7: **Pre Condition:** The player has unlocked a level where the Archer character has been introduced

Test Cases

Test Case for US1-1:
Git repository - Access Unity Git Repository:
A registered user on Gitlab who is invited to the team grey project should be able to access the git repository for the Unity game project.

Precondition: The user must have been invited to join as a developer to the SE Grey Team's project.

Assumption: A supported web browser or gitlab application is being used.

Test steps:
1. Follow the link on the developer invitation email from William Chambers.
2. Register your email and choose a password to create a Gitlab account.
3. Access the SE Grey Team project "Coursework 2 - Game"
4. Open the Coursework 2 - Game project for access to the repository for the Unity project.

5. Future logging into Gitlab.com or using a downloaded gitlab instance will provide access to the repository.

Expected Result: Users have easy access to the Unity repository by logging into their gitlab accounts and opening the Coursework 2 - Game project.

Result: All members of the team received an invitation email from William Chambers and managed to successfully follow the test steps to gain access to the Unity project repository.

Test Case for US1-3:

Game backstory - A backstory for the game is written and all members of the team understand the narrative of the game

Precondition: The game is a real time strategy game with a space theme.

Test steps:

1. Game backstory is presented to the group as a video animation.
2. Feedback on the backstory is given by each member of the group and discussed.
3. Backstory is presented to the customer during a customer meeting.
4. Feedback from the customer is given and recorded.

Expected Result: The user is engaged by the backstory and feels more immersed in the game. The customer likes and approves the backstory for the game.

Test Case for US1-4:

Pathfinding AI - A pathfinding AI is created and used for game characters.

Precondition: A working Unity game project with a 3d environment and game character models.

Assumption: The AI is written in C# and compiled successfully in Unity.

Test steps:

1. The AI script is attached to a game character in Unity.
2. The scene in Unity is played.
3. The character's is asked to move in the game environment by a user mouse input
4. The character's behaviour and movement is observed.

Expected Result: The game character moves efficiently from where it was to where the user directed it to go after user input.

Test Case for US1-5:
Create wall - A simple wall is created that other objects in the game can interact with.

Precondition: The unity game has a ground and characters which can move around using a pathfinding algorithm and user input.

Test steps:
1. A wall model is created and added to the game environment.
2. The scene is played in Unity.
3. The characters are directed towards the wall and the other side of the wall.
4. The interaction between the characters and the wall are observed.

Expected Result: When a character is moved at a wall its movement will be impeded. The character will navigate around the wall to reach its target.

Test Case for US1-6:
Create ground - A plane is created and added to the game environment which other game objects will move across and be placed on

Precondition: The unity game has a 3d environment.

Test steps:
1. A plane is visible in the Unity scene.
2. Add moving character objects and static objects to the plane.
3. Play the scene.
4. Observe the objects and whether or not the static objects remain static and the character objects move across the plane

Expected Result: There is a visible plane which objects can be added onto. The characters move across the plane and static objects stay in place on the plane.

Test Case for US1-7:
Intro Screen - An aesthetically suitable introduction screen is created.

Precondition: The game is a space themed RTS game.

Assumption: The intro screen will be suitable for adding to the start of the Unity game.

Test steps:
1. View the design for the intro screen.
2. Check that the screen conveys the space theme.
3. Identify a "play" button on the screen for starting the game.

Expected Result: The intro screen is appealing to look at and conveys the space theme to the viewer. There is a suitable design for a play button to launch the game.

Test Case for US1-8:

Knight Character - A Knight character is created which causes close proximity damage to enemy characters and itself has depletable health.

Precondition: There is a 3d game environment with a ground for characters to move across.

Test steps:

1. Create a Unity scene with a player Knight character and an enemy Knight character.
2. Play the scene.
3. Move the Knight character towards the enemy characters.
4. Observe the health of the characters and how close the characters must be to inflict damage on each other.

Expected Result: There are Knight characters which can inflict damage on each other when they are in close proximity. The characters have depletable health which when fully depleted makes the characters disappear.

Test Case for US1-9:

Warrior Character - A Warrior character is created which causes a high amount of close proximity damage to enemy characters and itself has depletable health.

Precondition: There is a 3d game environment with a ground for characters to move across.

Test steps:

1. Create a Unity scene with a player Warrior character and an enemy Warrior character.
2. Play the scene.
3. Move the character towards the enemy character.
4. Observe the health of the characters, how close the characters must be to inflict damage on each other and the amount of damage inflicted.

Expected Result: There are Warrior characters which can inflict a high amount of damage on each other when they are in close proximity. The characters have depletable health which when fully depleted makes the characters disappear. The amount of damage caused by the Warrior character is larger than that of the Knight character.

Test Case for US1-10:
Archer Character - An Archer character is created which causes a long range damage to enemy characters, fires a visible projectile and itself has depletable health.

Precondition: There is a 3D game environment with a ground for characters to move across.

Test steps:
1. Create a Unity scene with an Archer character and an enemy Archer character.
2. Play the scene.
3. Move the character towards the enemy character.
4. Observe the health of the characters, how close the characters must be to inflict damage on each other and if there is a visible projectile fired by the characters when they are inflicting damage.

Expected Result: There are Archer characters which can inflict damage on each other from long range. The characters have depletable health which when fully depleted makes the characters disappear. There are visible projectiles fired by the characters when they are inflicting damage.

CRC Cards

CRC card links to US1-4

Class Name: InputHandler	
Responsibilities: ● Handles the unit movement ● Select units	**Collaborators:** ● BasicBuilding ● LayerMask (unity class)

Class Name: BasicBuilding	
Responsibilities: ● Obstructs movement in the game ● Not movable	**Collaborators:** ● GameObject

CRC card links to US1-5

Class Name: BuildingHandler	
Responsibilities: ● Initialises various basic building	**Collaborators:** ● BasicBuilding ● LayerMask (unity class)

CRC card links to US1-5

Class Name: BuildingStatTypes	
Responsibilities: ● Basic characteristics of building characters like health, attack.	**Collaborators:** ● Stand-alone class

CRC card links to US1-8, 1-9, 1-10

Class Name: BasicUnit	
Responsibilities: ● Helps to create different types of units	**Collaborators:** ● GameObject (unity class) ● UnitStatTypes.Base

CRC card links to US1-8, 1-9, 1-10

Class Name: UnitHandler	
Responsibilities: ● Sets the stats for instances of units	**Collaborators:** ● BasicUnit ● LayerMask (unity class)

User Interface Design Elements

Design Element 1.1: Intro Screen Design Version 1

- Includes title logo, start button, options button, and exit button in stylized font on a starry black and white background
- Fulfills User Story 1-7

Title Logo:

UI design: We chose a gun metal texture over the top of the basic title font to construct the logo along with a large sword in between ‘space’ and ‘Knight’ to convey that the game is about strategic war between Knights. It’s lit from below making the logo seem close to a star underneath, grounding the logo in its background, yet creating contrast with the dark blacks in order to stand out.

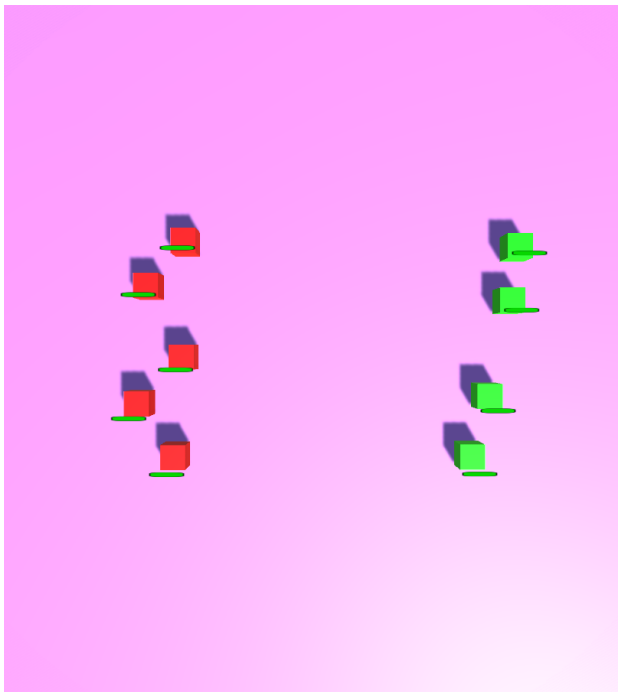
UX design: The logo is positioned large in the center of the screen above the start button. This is the focal point of our scene and it must grab attention. It should guide users to the smaller text immediately below. Notice we have an inverted pyramid structure leading the player downwards from the logo.

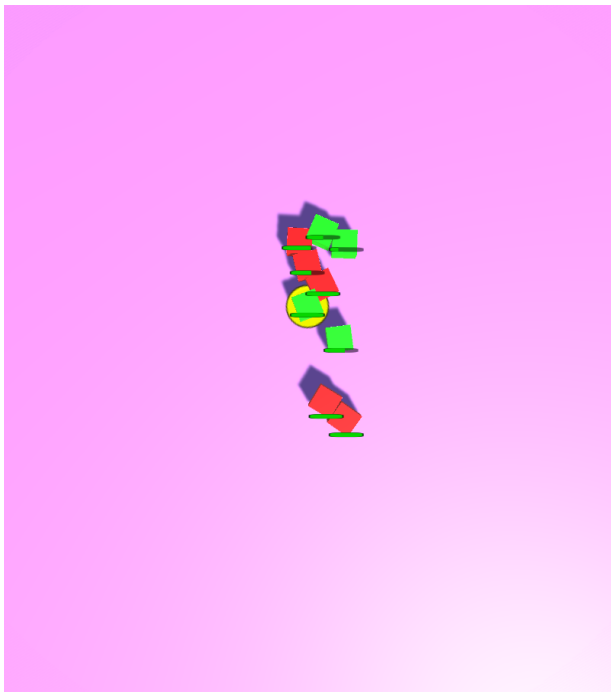
Start Button:

UI design: We chose to use a white font as it is consistent with the background of stars and helps to convey our space theme to the user. The font itself is basic title font which is simple and clear, much like the game itself.

UX design: We positioned that start button in the middle of the screen to guide the user to noticing it first. We make the font larger than that of the other buttons as this button is more important and we wanted to help guide the user towards using it as they need to press the button to begin the game. We made the button flashing on the screen to further distinguish it from the other buttons and encourage the user to click on it.

Design Element 1.2: Game Demo Shown in Customer Meeting 3





- The green squares represent the player's characters and the red squares represent enemy characters. The green and red colors were chosen as they are prototypical of "good" and "evil", respectively. This is a visual aid for the player's to distinguish their characters from the enemies.
- The player can select a fighter and choose where on the map to place them. Once the fighter is in a particular radius of the enemy, the enemy begins to attack. The green 'health bar' below all of the squares shows the individual character's health level, and this can be seen going down when a character is under attack. The health bar was added so that the player can visually keep track of which characters are under attack and may need to be moved out of the line of fire. The character is defeated once their health level reaches 0 (depicted as a fully black bar symbolizing no health remaining).
- The pink ground color was chosen because 1) the green and red color of the characters are easy to see and distinguish against the pink ground, 2) the pink color is unique and eye-catching, and 3) it enhances the modern, minimalist space theme of the game.
- Satisfies basic game mechanics for User Stories 1-4, 1-5, 1-7, and 1-9

Sprint 1 Game Progress

A basic title screen was implemented that shows the space Knights logo. It also features 3 buttons; Start, options and quit. The functionality of these menu buttons has yet to be implemented.

Game mechanics progress

The game currently has three character types (Knight, Warrior and Archer). Only the Knight has been implemented into an in-game scene however.The 3 character types can exist as both player characters and enemy computer controlled characters. The player Knight can be selected by a user, commanded where to go and when in a certain proximity of an enemy, will fight. The enemy, when within a certain proximity will also attack the player. Both feature assignable attack and health values. If the health of either character type reaches zero, the character will 'die' and be removed from the game. Makes use of Unity's NavMeshAgent for path finding and AI. The game view is top down.

Sprint 1 Retrospective

- Explained game mechanism to customer
- Began to develop fighting mechanism for the first fighter type and its enemy
- Created an intro cut-scene that explains game back-story
- Created a start screen that takes player to a 'map' screen where they can see what level they are on
- What went well
    - Good meeting attendance
    - Good demo to show
    - Able to identify blockers in our process and find ways to get around them
    - Everyone happy with the tools we've been using: teams, confluence and jira
- What went wrong
    - Underestimated Unity difficulty
    - Communication, not everyone was always on the same page
- What we learnt
    - Learnt Unity and c#
- What we can improve on in the next sprint
    - Rank our user stories
    - Communicate whatever changes we make or what we are working on
    - Agree on times for next meeting ahead of day (more consistent meeting)
    - Share knowledge of errors we've come across
    - Try to meet/work in person as much as possible

Sprint 2 Overview: 21 Nov 2021 - 28 Nov 2021

- Adjust our project management for sprint 2 in accordance with what was discussed in the Sprint 1 Retrospective
- This sprint we set out to accomplish the following:
    - Develop the first iteration of a fully functioning game
    - Animate all character models
    - Begin level design for more levels beyond level 1
    - Implement the character selection logic and UI feature

Sprint 2 Meeting Minutes

Team Meeting 8: 22 Nov 2021

Minutes: 15:00 - 16:00 @ Virgil Building

Participants
- George Griffiths
- William Chambers
- Ani Udom
- Vladislav Shishkin via Microsoft Teams
- Dinil Wickramasinghe
- Vinay Kumar Gupta via Microsoft Teams
- Yash Malik via Microsoft Teams

Goals
- Close Sprint 1
- Plan Sprint 2 and start it

Discussion Topics
- Assigning each task to individuals
- Organising Jira board and moving incomplete tasks from sprint 1 to sprint 2

Action Items:
- Everyone to update stories on Jira as they begin to work on it
- Start on tasks for sprint 2 as outlined in teams Jira board

Customer Meeting 4: 24 Nov 2021

Minutes: 10:30 – 10:45 @ Chancellor's Building 5.13

Participants

- Ashton Mitchell
- Ani Udom
- William Chambers
- Vladislav Shishkin
- Yash Malik
- Vinay Kumar Gupta

Goals

- Demonstrate current version of game
    - Updated intro screen with graphics
    - 4 fully functioning character types (Knight, Archer, strongmen, Healer), but only Knight is fully animated at present

Discussion Topics and Analysis

- Discussion of game mechanics for each character currently implemented:
- Player gets to move specific characters or groups of characters into or out of battle
- Different characters have different moving speeds and traits
- Discussion of next steps:
    - Group of chosen characters will spawn together at start of level
    - Larger map size and more developed terrain
    - All characters fully animated
    - More visuals and design elements
    - Mechanic of players ability to chose their teams at the start of each level
- Customer questioned whether the terrain would remain flat - suggested that some topographic features or buildings would add interest to the game play
- Customer impressed with the logical strategy of the game
- Customer okay with mechanics of game taking priority over art and visuals
- Customer suggested implementation of character that can rearrange building/level layout
- Fancy animations not essential, focus on clean overall design
- Customer questioned fully animated character models and suggested simplistic "cube" character models instead for a cleaner and unique game aesthetic.

Action Items

- Look at wider timeline and make overall plan instead of solely working sprint by sprint
- Consider Customer's suggestion of implementing distinct "cube" character design instead of messy animations

Team Meeting 9: 24 Nov 2021

Minutes: 11:00 - 12:00 @ Computer Lab 2.56

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals

- Work on backlog task assignments together to get progress done in synch with one another

Team Meeting 10: 26 Nov 2021

Minutes: 11:00 - 12:00 @ Virgil Building

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- Vladislav Shishkin via Microsoft Teams
- Dinil Wickramasinghe via Microsoft Teams
- Vinay Kumar Gupta via Microsoft Teams
- George Griffiths via Microsoft Teams

Goals

- Decide on animation
- Check in how everyone is getting along
- Horizon planning

Discussion Topics

- What should we do about the animation?
    - Commit to cube-based design
    - Or commit to actual animated characters
    - Or create two scenes with both options and ask customer which is preferred

- Level design
    - Need to get character selection scene working first @Vladislav Shishkin and @Dinil Wickramasinghe
    - @William Chambers Suggest the selection ‘scene’ as an overlay over the game arena rather than a separate scene
- Start focus on Documentation
- Merging branches everyone has worked on
- How to add the ability to select characters
- Horizon Planning:
    - Should have project done around the 15th Dec
    - We should have the game “done” by the second to last customer meeting at the latest (8th Dec)
- What we have left to accomplish
    - Level design @Dinil Wickramasinghe and @Ani Udom
    - UI for character selection @Vladislav Shishkin
    - Win/lose scene @Vinay Kumar Gupta
        - Win condition is to destroy the ‘enemy core’: if core building = 0, show win screen
        - Lose condition is if all player characters are defeated: if player characters = 0, show lose screen
    - Sprint Documentation @Ashton Mitchell and @Ani Udom
    - Game documentation
    - Create video demonstration

Decisions

- Stick with green squares as this would take up less time and allows for a more uniform design

Action Points

- Everyone make user stories for parts they’ve worked on. Possibly break them up into smaller bits
- Start work on Level design
- Everyone to give status updates on teams each day

Team Meeting 11: 29 Nov 2021

15:00 - 16:00 via Microsoft Teams

Participants
- Ashton Mitchell
- Ani Udom
- Dinil Wickramasinghe
- George Griffiths
- Vladislav Shishkin
- Vinay Kumar Gupta
- Yash Malik

Goals
- Sprint 2 Retrospective
- Discuss progress and timeline until deadline
- Discuss what will be shown to customer at next meeting

Discussion topics
- Presentation of win/lose screen: design is good but logic may need adjusting
- Change to check whether features exist instead of keeping track of the number of characters
- Level design still to do
- Character selection i.e. “team-building” feature left to implement
    - Both mechanics and UI
- More team members to join in on documentation

Sprint 2 Requirements List

Explanation of Changes to Requirements List

- R2.5 was updated based on Team Meeting discussions of what could be achieved to a high quality within the given timeframe.
    - In addition to this update, R3.2 was created as the team decided it was important for players to have a chance to get used to the gameplay before diving into difficult levels.

- Based on design feedback from the customer in Customer Meeting 4, the team decided in Team Meeting 10 to update R2.8 from fully animated character models to simplistic, distinctive “cube” character models.
    - Following on from this decision, it was also decided that the different characters (between character types as well as between player’s characters and enemy’s characters) needed clear visual distinctions by color and symbol. R3.1 was created for this.
- R3.3 was created based on Team Meeting discussions and approved by customer. Helpful to inform the player of how well they are doing in the game.

Updated List Version 2

As each requirement is met, the corresponding test or design element will be listed in parentheses at the end of the requirement.

Key: ~~Strikethrough~~ indicates removed requirement

R1.1: ~~Dungeon themed game~~
R1.2: Game that will run on a laptop
R1.3: ~~Game that will run on a phone~~
R1.4: ~~A “bot” that can play the game on it’s own or against other bots~~
R1.5: Ability to understand how well the player is doing in the game (DE1.2)
R1.6: Game that has increasing difficulty through progression
R1.7: Gameplay modelled after real-time strategy (RTS) (see RTS Style Game Storyboard)
R2.1: Character selection feature as part of gameplay strategy
R2.2: Intro screen with title, “START”, and “OPTIONS” buttons (see DE1.1)
R2.3: Cut scene to establish game backstory (DE2.2)
R2.4: Level selection functionality (DE2.5)
R2.5: 6 levels
R2.6: Camera zoom functionality (Test Case for US2-3)
R2.7: Enemy characters for player to battle against (DE1.2)
R2.8: Simplistic “cube” character models for cleaner, unique design
R2.9: Game music throughout to enhance story and theme
R3.1: Clear distinction between character types
R3.2: Tutorial level to familiarize player with gameplay
R3.3: Win/lose screens to inform player of level outcome

Sprint 2 Backlog

B1.8: Create ground (US1-6)
B2.1: Level selection screen (US2-2)
B2.2: Camera movement/input (US2-3)
B2.3: Healer character (US2-10)
B2.4: static building/enemy (US2-4)
B2.5: 3d model for strong fighter (US2-5)
B2.6: 3d model for Archer (US2-6)
B2.7: landscape graphics (US2-8)
B2.8: 3d model alien enemy (US2-7)
B2.9: Refine backstory in context with gameplay (US2-9)
B2.10 Add in the cutscene video to the game (B1.10, US2-11, DE2.2)

Complete Backlog Tasks

B1.8: Create ground
B2.9: Refine backstory in context with gameplay
B2.3: Healer character (DE2.3)
B2.4: static building/enemy (DE2.4)
B2.8: 3d model alien enemy (DE2.1)
B2.5: 3d model for strong fighter
B2.2: Camera movement/input
B2.1: Level selection screen (DE2.5)
B2.11: Create Intro Screen Button Functionality (US2-1)

New Backlog Tasks

B2.11: Create Intro Screen Button Functionality (US2-1)

Sprint 2 Exception Handling

- In Customer Meeting 4, the customer suggested a big change to the team's character design plan. The customer liked the simplistic cube design of our character 'placeholders' and said that he would prefer those in the final game over fully animated character models, as they could end up looking 'messy', hard to distinguish, and like every other game. The customer thought the cube design was unique and would set the game apart.
    - So, the team decided (by a vote) to switch gears and pursue the cube character model design. New decisions then had to be made regarding color of each character type and how to distinguish the cubes from one another. These decisions were pushed to sprint 3.
- The team also struggled to meet most of the tasks set out at the start of sprint 2 due to other university commitments. The team handled this by accepting and understanding the difficulty of this week, and pushing tasks to sprint 3. By pushing the tasks to sprint 3 we all know that we must be dedicated to working even harder next week to complete what is needed. The handling of this situation created a great sense of unity in the team.

Sprint 2 Product Documents

User Stories

User Story 2-1: Create Intro Screen Button Functionality

AS A Player *I WANT TO* Have functioning buttons on a menu screen *SO THAT* I can start and exit the game and change music options	*Acceptance Criterion:* The buttons on the intro screen lead to the appropriate scene.

User Story 2-2: Create Level Selection Screen

AS A Player *I WANT TO* A level selection screen *SO THAT* I can see what levels I have completed and how many levels I have left. I would also like to have the option of replaying certain levels.	*Acceptance Criterion:* A functional screen that shows the level progression. Displays levels that are locked. *Comments:* Players must win one level before the next is unlocked

User Story 2-3: Work on Camera Movement/Input

AS A Player *I WANT TO* Be able to adjust the camera angles and zoom in and zoom out *SO THAT* I can have a clearer picture of certain areas of the game and move from one side of the made to the other	*Acceptance Criterion:* Camera movements work smoothly and are efficient to use.

User Story 2-4: Create Building (Static Enemy)

AS A Player *I WANT TO* Have so enemy buildings *SO THAT* I have some static enemies to fight against. Giving me a variety of in game tasks	*Acceptance Criterion:* Distinct buildings that players can interact with and destroy are created.

User Story 2-5: 3D Model for Strong Fighter

AS A Player *I WANT TO* Have a visually appealing animated character *SO THAT* I can tell the different characters in the game apart	*Acceptance Criterion:* An animated character is created with distinct features and interacts well with other objects in the game

User Story 2-6: 3D Model for Archer

AS A Player *I WANT TO* Have a visually appealing animated character *SO THAT* I can tell the different characters in the game apart	*Acceptance Criterion:* An animated character is created with distinct features and interacts well with other objects in the game

User Story 2-7: 3D Model for Alien Enemy

AS A Player *I WANT TO* Have a visually appealing animated character *SO THAT* I can tell the enemies apart from the fighter characters	*Acceptance Criterion:* An animated character is created with distinct features and interacts well with other objects in the game

User Story 2-8: Create Landscape Graphics

AS A Player *I WANT TO* Have nice visuals for the landscape *SO THAT* The game is aesthetically appealing	*Acceptance Criterion:* Create non-interactable objects in the game terrain to set the scene of the game and add obstacles for players

User Story 2-9: Refine Backstory in Context with Gameplay

AS A Player *I WANT TO* Have a backstory for the game *SO THAT* I have some context for the game mission and be engaged	*Acceptance Criterion:* A story for the game is created that is fun and engaging

User Story 2-10: Create Healer Character

AS A Player *I WANT TO* Have a Healer character *SO THAT* I can restore the health of my fighter characters that are about to die	*Acceptance Criterion:* A character that restores the health score for characters on their side.

User Story 2-11: Add Intro Cutscene Video to the Game

AS A Player *I WANT TO* Have an intro video for the game *SO THAT* I can have an introduction to the game backstory	*Acceptance Criterion:* Have a video that plays at the start of the game with a brief story of the game. *Comments:* There should be an option to skip this video

Revised User Stories - Final Set

Following the discussion we had with the Customer during this sprint (See Customer Meeting 4), we decided to keep the game design as simple as possible. This meant we would no longer create animated models for each character (US2.5 2.6, 2.7 and 2.8) type but rather stick to the cube design we had been developing. We would distinguish each cube by making them different shades of green (for the player characters) and different shades of red (for the enemy characters). We also included different images at the top of the cubes to discern them (see DE2.3 and 3.5) These are the final set of user stories for this sprint:

User Story 2-1: Create Intro Screen Button Functionality
User Story 2-2: Create Level Selection Screen
User Story 2-3: Work on Camera Movement/Input
User Story 2-4: Create Building (Static Enemy)
User Story 2-9: Refine Backstory in Context with Gameplay
User Story 2-10: Create Healer Character
User Story 2-11: Add Intro Cutscene Video to the Game

Requirements Use Cases

RU1.2 Use case for US1-9: Healer Character
Scope: Game level
Level: user goal
Context: Players can select a Healer character which works to replenish the health of the player's other characters instead of attacking the enemies.
Frequency of Occurrence: Any time a Healer character is chosen in a level
Open Issues: How much space do Healer characters take up in army selection?
Health Level? Healing power? What level is this character first introduced?

Design Use Cases

Use Case 2.1:

UC1-1**: Use Case:** Flow from Start Screen to Gameplay

UC1-2: **Author:** AM

UC1-3: **Purpose:** Establish the flow of screens that the player can navigate through before beginning gameplay in level 1.

UC1-4: **Overview:** Starts when the player runs the game from a computer. The intro screen appears first, with background music and 3 button options to click: **alternative 1:** the player can click the "START" button, **alternative 2**: the player can click the "OPTIONS" button, or **alternative 3**: the player can click the "EXIT" button. **Alternative 1** will lead to the cutscene backstory video, then to the instruction screen. After reading the instructions, the player must click "NEXT" to progress to the level selection screen. At the level selection screen, the player can **alternative 1.1**: click on "Level 1", as all other levels will be locked until beating level 1, or **alternative 1.2**: click "Return to Main Menu", where they can restart their flow of navigation from the intro screen. **Alternative 2** will lead to the options screen, where **alternative 2.1**: the player will be able to toggle the background music on or off, or **alternative 2.2**: the player can click "Return to Main Menu", where they can restart their flow of navigation from the intro screen. **Alternative 3** will exit the player out of the game entirely.

UC1-5: **Cross References:** US 1-3.2, 1-8, 2-1, 2-2, 2-9, 2-11

UC1-6: **Actors:** Player

UC1-7: **Pre-Conditions:**

UC1-Pre-1: The player successfully runs the game from their computer by having all of the necessary applications installed properly.

UC1-8: **Post-Conditions:**

UC1-Post-1: The player is able to navigate through the initial screens to **alternative 1**: begin gameplay or **alternative 2** exit out of the game.

Use Case 2.2:

UC2-1: **Use Case**: Healer Character

UC2-2: **Author:** AU

UC2-3: **Purpose:** Have a character that can increase the health score of other fighter characters during gameplay.

UC2-4: **Overview:** Starts when Player selects 1 or more Healer characters for their army.

If their army is not full (based on total army strength allowance parameter), **alternative 1:** the player can add more characters. **Alternative 2:** character selection is complete and the level starts.

As the level starts other characters will begin to fight the enemy and their health score will start to decline.

The player can move the Healer character to other characters that are running low on health score. **Alternative 1:** The Healer character begins to replenish the health of character on their side. **Alternative 2:** The Healer character begins to lose health if they get attacked by an enemy.

UC2-5: **Cross References**: US 1-9, 1-10, 1-11, 3-5 and 3-6

UC2-6: **Actors:** Player

UC2-7: **Precondition**:

UC2-Pre-1: The player has unlocked the level that introduces the Healer character.

Use Case 2.3:

UC3-1: **Use Case:** Buildings

UC3-2: **Author:** AU

UC3-3: **Purpose:** Have a static enemy for players to destroy

UC3-4: **Overview:** There are 3 types of buildings that could be in a level of the game.

There will always be an 'enemy core' building. (If this building is destroyed the player wins the game). There would also be a combination of regular buildings (that serve as an obstacle for player) and assault buildings that can cause harm to the players.

Buildings are given a health score and players can use their fighter characters to destroy these buildings.

As characters attack the buildings, their health depletes, and they can be destroyed.

Alternative 1: Player successfully destroys building (if an enemy core building, they win the game). **Alternative 2:** Enemies can attack characters while they are destroying buildings and they are defeated. **Alternative 3:** Player chooses the wrong type of character to destroy a building and the character's health depletes without damaging the building.

UC3-5: **Cross References:** US 1-9, 1-10, 1-11, 2-4

UC3-6: **Actors:** Player

UC3-7: **Post Condition:**

UC3-Post-1: If building is destroyed **alternative 1:** If the building is the enemy core, the player wins the level, **alternative 2:** If the building destroyed is any other type of building the player can carry on other tasks in the level and would potentially be able to get closer to the enemy core building.

Tests

Test Case for US 2-1:
Start Screen - A screen at the start of the game with buttons to lead to the game, options and exit.

Assumption: The user is running the correct build type for their operating system.

Precondition: A unity game build has been created which includes multiple scenes for the start screen to lead to.

Test steps:
1. Run a built version of the game.
2. Check to see that the start screen loads correctly.
3. Interact with the start screen buttons by left mouse clicking on each of them.
4. Observe if the start button leads to the start of the game.
5. Observe if the option button leads to an options menu
6. Observe if the exit button exits the game.

Expected Result: The start screen loads after running the game build and the buttons function correctly to start the game, open an options menu and exit the game.

Test Case for US 2-2:
Level Selection - A screen in the game with buttons to lead to the different game levels.

Assumption: The user is running the correct build type for their operating system.

Precondition: A unity game build has been created which includes multiple scenes for the level selection screen.

Test steps:
1. Run a built version of the game.
2. Click start at the Start Screen
3. Check to see that the level selection screen loads correctly.
4. Interact with the level selection buttons by left mouse clicking on each of them.
5. Observe if level selection buttons lead to their respective levels..
6. Observe if the exit button returns you to the start screen.

Expected Result: The level selection screen loads after running the game build and navigating to it from the start screen. The buttons function correctly to start the game levels and exit the level selection screen to.

Test Case for US 2-3:
Camera movement - A moveable camera is implemented for the game levels which can be panned and zoomed in and out.

Assumption: The user is running the correct build type for their operating system.

Precondition: A unity game build has been created which includes a loadable game level to use the camera in.

Test steps:
1. Run a built version of the game.
2. Click start at the Start Screen
3. Click on a level at the level selection screen
4. Check to see that a level loads correctly.
5. Provide user input using the WASD keys to pan the camera and observe if the camera correctly pans around the level.
6. Provide user input by holding the SHIFT key while using the mouse to left click and drag to pan the camera and observe if the camera correctly pans around the level.
7. Provide user input by using the RF keys to zoom the camera in and out and observe if the camera correctly zooms in and out.
8. Provide user input by adjusting the scroll wheel to zoom the camera in and out and observe if the camera correctly zooms in and out.

Expected Result: The camera correctly pans around the level with either input from the WASD keys or from holding SHIFT and draggin the camera around with the mouse using the left mouse click.

Test Case for US 2-4:
Building - Distinct buildings exist within game levels that players can interact with and destroy.

Assumption: The user is running the correct build type for their operating system.

Precondition: A unity game build has been created which includes a loadable game level with characters to move and buildings.

Test steps:
1. Run a built version of the game.
2. Click start at the Start Screen
3. Click on a level at the level selection screen
4. Check to see that a level loads correctly.
5. Move the player characters towards the building.
6. Observe how the player characters interact with the building and if the characters can walk through the buildings or have to navigate around them.

7. Observe if the player characters will attack the building.
8. Observe the health of the building and whether or not it is destroyed and removed from the level after being attacked.

Expected Result: The buildings provide an obstacle for the player characters' movement which they cannot pass through and instead will navigate around. The player characters will attack the buildings when in close enough proximity. The buildings have depletable health which is depleted while being attacked until eventually being destroyed and will disappear from the game level.

Test Case for US 2-5:

Strong fighter - A 3D model for a strong fighter with animation for movement.

Assumption: The user has a suitable computer program installed such as Blender or Unity for opening the 3D model and viewing the animations.

Precondition: A 3D model for a strong fighter has been created with animation for the movement of the 3D model.

Test steps:

1. Open the 3D model using blender.
2. View the 3D model in blender in the rendered view.
3. Observe the appearance of the model.
4. View the model animations in blender by playing them back using the timeline.

Expected Result: The 3D model for the strong fighter opens correctly, has the desired appearance and the animations function correctly.

Test Case for US 2-6:

Archer - A 3D model for an Archer with animation for movement.

Assumption: The user has a suitable computer program installed such as Blender or Unity for opening the 3D model and viewing the animations.

Precondition: A 3D model for an Archer has been created with animation for the movement of the 3D model.

Test steps:

1. Open the 3D model using blender.
2. View the 3D model in blender in the rendered view.
3. Observe the appearance of the model.
4. View the model animations in blender by playing them back using the timeline.

Expected Result: The 3D model for an Archer opens correctly, has the desired appearance and the animations function correctly.

Test Case for US 2-7:
Alien enemy - A 3D model for an alien enemy with animation for movement.

Assumption: The user has a suitable computer program installed such as Blender or Unity for opening the 3D model and viewing the animations.

Precondition: A 3D model for an alien has been created with animation for the movement of the 3D model.

Test steps:
1. Open the 3D model using blender.
2. View the 3D model in blender in the rendered view.
3. Observe the appearance of the model.
4. View the model animations in blender by playing them back using the timeline.

Expected Result: The 3D model for an alien enemy opens correctly, has the desired appearance and the animations function correctly.

Test Case for US 2-8:
Landscape objects - 3D objects to act as terrain and obstacles for player characters.

Precondition: A game level is created with landscape objects in.

Test steps:
1. Play a Unity game level scene which includes player characters and landscape objects.
2. Direct the player characters to move to the landscape objects.
3. Observe how the characters interact with the landscape objects.
4. Direct the player characters to move beyond the landscape objects.
5. Observe how the characters behave when they encounter the objects on their path.

Expected Result: The landscape objects provide a barrier which the player characters cannot move through. The player characters don't interact with the landscape objects other than navigating around them and having their path blocked by the objects.

Test Case for US 2-10:
Healer character - A Healer character which is controlled by the player in the same way as the fighting characters that will restore the other characters' health when it is in close enough proximity.

Precondition: A game level exists with enemy characters and player characters.

Test steps:
1. Play a Unity scene for a game level containing player Knight characters, an enemy Knight character and at least one player Healer character.
2. Direct the player Knight character towards the enemy night so that the player character has some of its health depleted but survives either by running out of the enemy's proximity or destroying the enemy.
3. Move the Healer character towards the injured Knight character.
4. Observe the health of the Knight character.

Expected Result: When in close proximity to a Healer character, the Knight character's health will be restored gradually.

Test Case for US 2-11:
Intro scene - An intro video plays at the start of the game which provides some visuals and backstory to engage the player.

Precondition: A Unity game exists that has multiple scenes which includes a scene at the start of the game which playbacks an intro video.

Test steps:
1. Launch a built version of the game.
2. Observe if an intro video with backstory plays after the game has launched.
3. Observe if after the video has finished the rest of the game loads.

Expected Result: After you open the game an intro video initially plays with the rest of the game loading after the video ends.

CRC Cards

CRC card links to US2-3

Class Name: CameraController	
Responsibilities: ● Zoom in and zoom out functionality	**Collaborators:** ● Transform (unity class) ● Vector3 (unity class)

CRC card links to US2-4

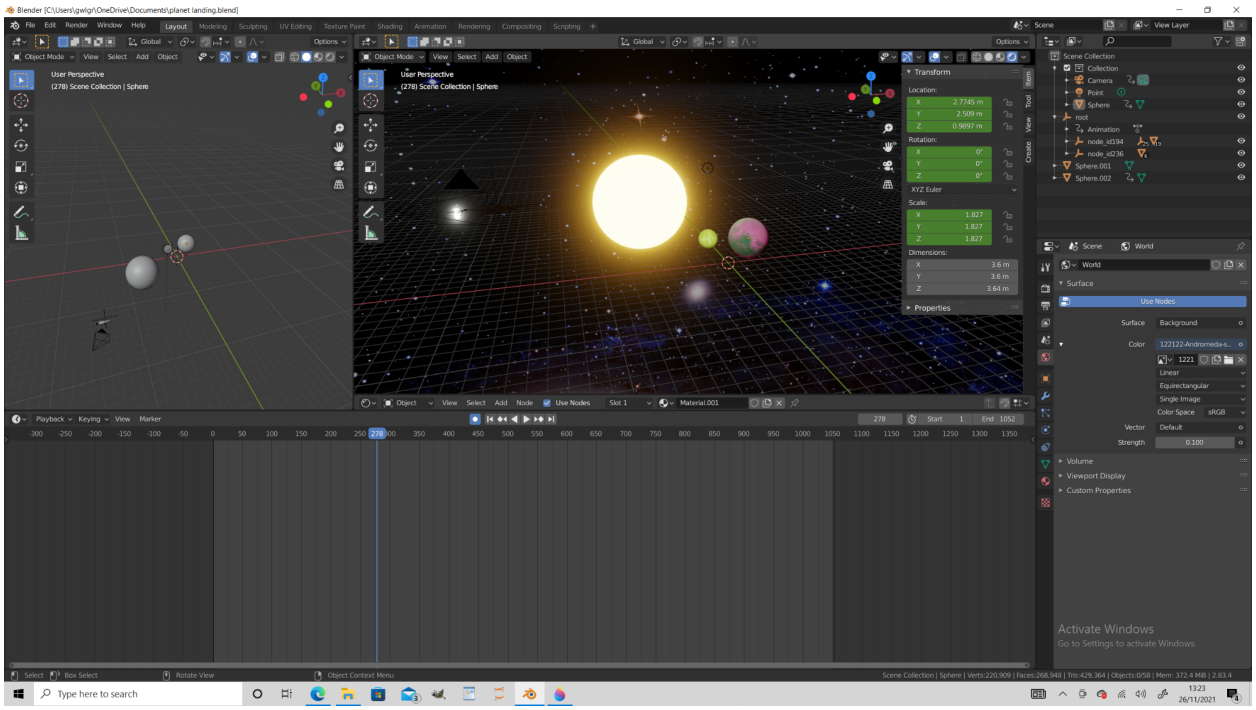
Class Name: EnemyBuilding	
Responsibilities: ● Non movable object in the game ● Has attack capabilities	**Collaborators:** ● NavMeshObstacle ● BasicBuilding ● BasicUnit ● BuildingStatTypes.Base ● DrawProjectile

User Interface Design Elements

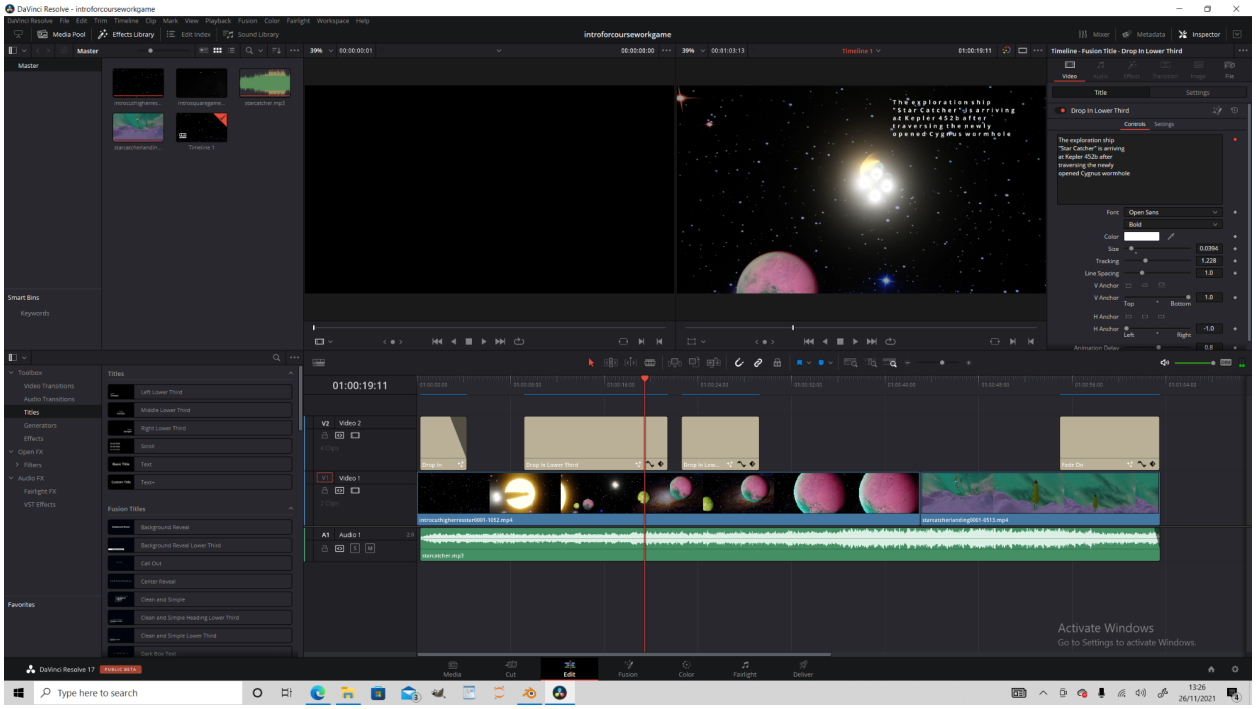
Design Element 2.1: 3D Model for Alien Enemy

- In response to User Story 2-7 an alien model was created.
- The model of the alien was designed to look familiar to other creatures on Earth but also different. The familiar aspects allow the user to interpret the character as being a living creature while also looking quite alien. The character model has elements that look fish-like and also insect-like.
- The main colour of pink was chosen because it is in keeping with the colours of the game and the planet that the game is set on. The brighter and contrasting colours on the top of the model allow it to be easily distinguished against a pink background and easily identifiable as being an enemy.

Design Element 2.2: Intro Cutscene Video for Game Backstory



- A 3D animation was made using blender. Models for a spaceship, star, planet and a moon were made in blender and animated into a scene where a spaceship approaches the purple planet.
- Purple and pinkish colours were used for the planet model as they were consistent with our chosen colour scheme for the game and suggested that the planet was an alien one.



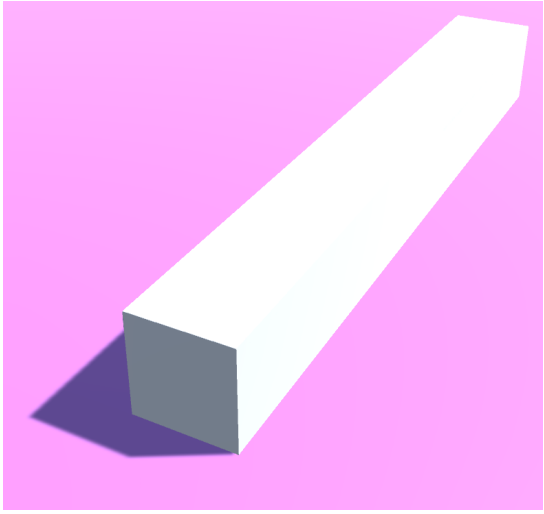
- The 3D animations were then edited together using the NLE, DaVinci Resolve.
- Captions containing the backstory were also added to satisfy User Story 2-9. A white font was used to contrast with the dark background and be easily read. The captions appear on the sides of the picture so they don't overlap with the moving objects in the scene and remain legible.
- Music was added to the video and it was exported as an mp4 file.

Design Element 2.3: Healer Character

- Healer characters restore the health of the respective ally units. The creation of the character followed the same procedure as for the other characters. The Healer uses the same aggro procedure to lock onto other characters, except that the Healer targets allies. Furthermore, the attack procedure is the same, except that the Healer restores health rather than depleting it. Healers can be multi-selected and commanded together with other player characters, however, they will ignore the command to attack enemies. Instead, if the Healer is selected and is given a command to follow an ally, the Healer will follow and heal the selected ally until that ally is destroyed or another command is given to the Healer. If the Healer is not currently executing a command, it will search for injured allies in its field of vision and follow them to heal.
- Fulfills User Story 2-10.

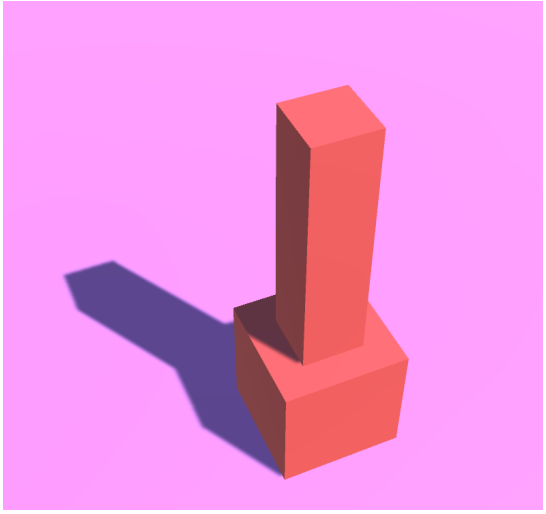
Design Element 2.4: Buildings

Boulder



This is a ‘boulder’, Players cannot interact with it but it can act as an indestructible barrier or obstacle for the player. It is white in color to stand out as unique from the enemy attack buildings.

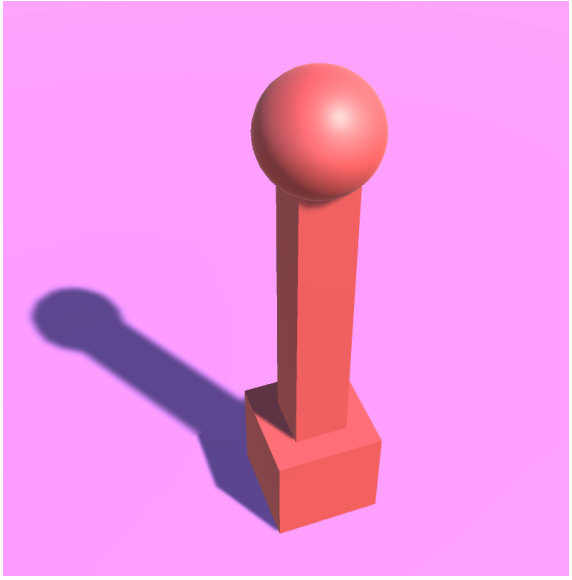
Basic Building



The red buildings all act as ‘static’ enemies. These buildings either protect the enemy camp or can attack the player’s characters. They were designed to be red to stay consistent with the color scheme of the other enemy characters. The visual cue of ‘red’ alerts the player to danger.

Players can attack and destroy this building. When designing levels we would use this as a destructible barrier that ‘protects’ the enemy players and some enemy buildings. It is introduced to the game at the same time the Warrior character (UC1.2) is introduced. See DE3.6 for the statistics of this object.

Attack Tower

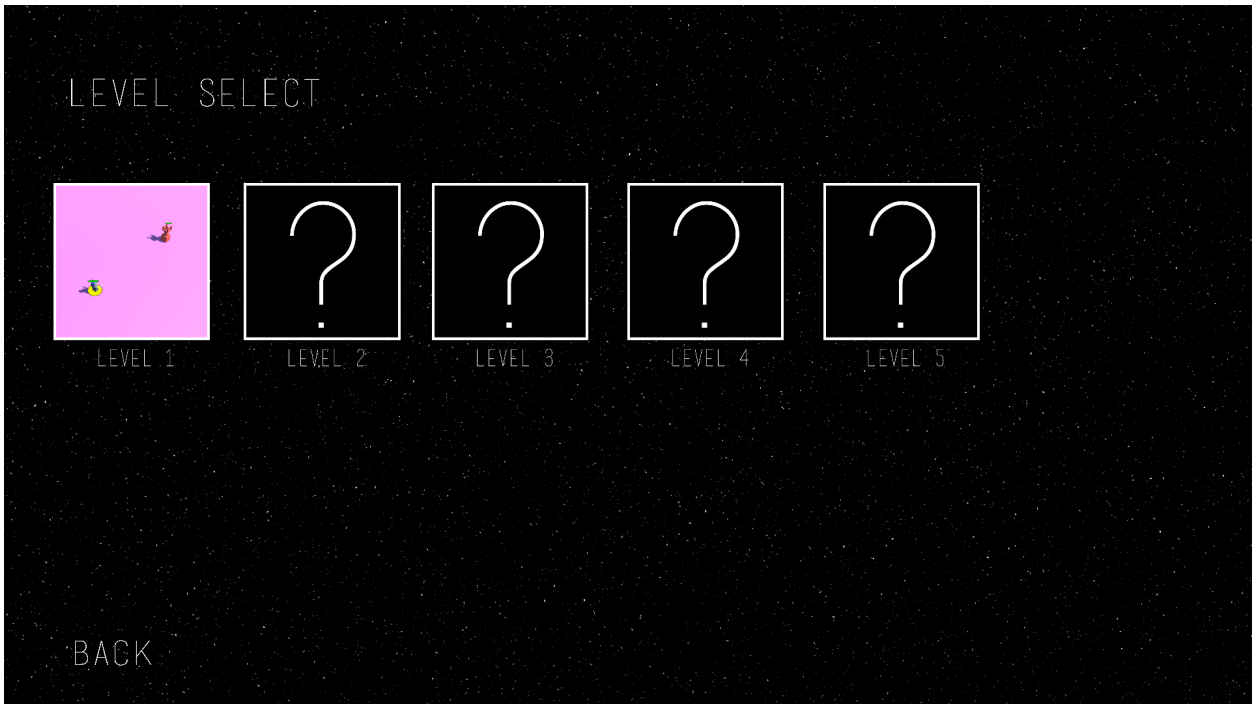


This building can attack characters from a long range. Similar to the Archer character, it can shoot beams at players. Players can also attack and destroy this building. It is introduced at the same time as the Archer character (US1.3). See DE3.6 for the statistics of this object.

Enemy Core Building

This building is the most important in the game, reflected by the complexity of its design. Once this building is destroyed the player wins the level (UC2.3, US3-4). The building is placed in every level and other enemy buildings and characters protect it from player characters. See DE3.6 for the statistics of this object.

Design Element 2.5: Level Select Screen Design (Version 1)



- Initial Level Select screen design demonstrating the total number of levels in the game, but only showing the levels unlocked one at a time, as the player wins each level. This choice was made so that the player is easily guided to selecting the most current level. In addition, the question mark symbol hiding the locked levels helps to keep the player engaged as they will be motivated to unlock these levels.
- The player will be able to return to any previous level after unlocking more levels.
- The player can also select the “BACK” button to return to the Intro Screen.

Sprint 2 Game Progress

The title screen now functions. The start menu progresses the user to the level selection screen, the options button takes the user to a menu that allows them to change in game volume and the exit button quits the game. A level selection scene was added that let's users choose which level to play. Currently as there is only one 'test' level, all buttons send the user to the same level. The Knight character was changed from a green cube to a small Knight character model. Animations were added to make the Knight more fluid. It featured stationary, running and attacking animations. Other character models will be added to the other player types in future sprints.

Game mechanics progress

The other character types (Warrior and Archer) have been added to the level. Archer attacks from a longer distance than the other character types (long range). The Warrior type is slower and has a stronger attack. A Healer character was also added. This character follows player characters and restores health when they are attacked. To balance playability, the Healer character itself has very low health so cannot be overused. The view of the game was also updated. The camera position now follows suit with other RTS games. It has been tilted to 45º angle and the ground plane has been rotated by 90º. The user can also move the camera (pan around the map) with the WASD keys or Click + drag, and zoom in and out with R and F or scroll wheel. Along with the Healer character, building structures were added. Previously the customer had mentioned it would be advisable to create some sort of obstacle rather than have a flat plane. Because of this, the buildings were created. Four building types exist; Attack tower, 'fence', obstacle and core tower. The attack tower functions like an enemy Archer but is static. The fence is a non attacking structure that can be used for blocking entrances. It is destroyable. The obstacle is a stationary object that cannot be destroyed or moved. The core tower is an object that can be destroyed and in doing so wins the game.

Sprint 2 Retrospective

- This sprint we accomplished:
    - Developed our first iteration of a functioning game
- What went well:
    - Communicated better about what we were working on
    - Able to quickly adapt to the customer's changing requirements through frequent team meetings
    - Better dividing of tasks and better task assignment clarity
- What went wrong:
    - General productivity this sprint was lacking - the team failed to meet a lot of the set tasks due to other university commitments
    - The customer suggested a change that significantly altered our design plan in Customer Meeting 4. Instead of implementing fully animated character models, the team decided to adapt the plan to appease the customer and instead decided to pursue simplistic cube character models as suggested.
    - The character selection feature is proving difficult to implement and has been moved to the next sprint, again. This will need to be addressed in the next sprint for feasibility.
    - Level design was started through the creation of level design elements (see Design Element 2.4), however the actual design of each level has had to be pushed to the next sprint.
- What we can improve on in the next sprint:
    - Greater team focus on documentation and the final project deliverables
    - Recording tests
    - Try to meet in person as much as possible and complete work on the project together
    - More communication about what features are feasible to implement in each sprint timeframe

Sprint 3 Overview: 28 Nov 2021 - 05 Dec 2021

- Produce a “complete” version of the game, including:
    - Updated intro screen with new “Square Wars” title and finished menu option buttons (see Intro Screen Design Version 2)
    - Updated “Square Wars” backstory intro cutscene video (see Intro Cutscene Video for Game Backstory Version 2)
    - Updated level selection screen design including tutorial level and 6 further levels (see Level Selection Screen Design Version 2)
    - 6 complete level designs of increasing difficulty (see Level Designs)
    - Finished cube characters (see Cube Character Models)
    - Screens to provide unlocked character introductions (see Character Introduction Screens)
    - Win/lose screens to inform the player of the outcome of each level (see Win/Lose Screen Design)
- Replace some team meetings where we just discuss the project with team meetings to actively work together on the development of the project
- Goal to be more efficient than previous sprints based on ability to produce “finished” deliverable by next customer meeting

Sprint 3 Meeting Minutes

Team Meeting 12: 01 Dec 2021

Minutes: 13:45 - 15:00 @ Virgil Building

Participants
- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths via Microsoft Teams
- Vladislav Shishkin
- Dinil Wickramasinghe via Microsoft Teams
- Vinay Kumar Gupta via Microsoft Teams
- Yash Malik

Goals
- Discuss how to handle the remaining time in sprint 3 after the customer cancelled this sprint’s meeting
- Updates on team progress and next steps

Discussion topics

- Task assignment conflict: two team members created different designs for the win/lose screens
    - Team discussion of which designs to implement - going with the screens which are more complete, but will put the correct background from the main menu screen behind the text
    - Team discussion of why this error in communication occurred: splitting the task into design and logic unclear
    - Moving forward the team must consult the task assignments on the Jira page before starting work to avoid unnecessary replication
- Discussion of next steps/changes to design:
    - need to give each character type a specific color cube
    - need to implement updated backstory cutscene after player clicks starts from the main menu
    - character selection screen too difficult to implement in the timeframe instead just include certain characters on each level and change features such as enemy count and obstacles to adjust level difficulty
    - character “health” and “attack” weightings determined (based on maximum health 100): see Table in DE3.7.
    - Character and level cost determined (setting the boundaries for strategy and how to win/lose)
    - Design 6 levels plus tutorial level:
        - Knight introduction before tutorial level
        - Archer (plus enemy tower) introduced before level 2
        - Warrior introduced before level 3
        - Healer introduced before before level 4
        - Level 5 and level 6 increase in difficulty
        - Character introductions need to include equivalent enemy character introduction (including small image showing color) and also what each character is good at (e.g. better at attacking enemy towers, heals your team, etc.)

Action items

- Need to get updated backstory cutscene into game - George Griffiths
- Updated title main menu screen - Dinil Wickramasinghe
    - Remove “options” button
- Change colors of cubes for each character type (i.e. Knight, Archer, Warrior, Healer)
- Level design sub-team
- Screenshots of each character model as they appear in the game - Vladislav Shishkin

Decisions

- Remove character selection feature
- Get “complete” version of game done by next customer meeting on Dec 8

Team Meeting 13: 03 Dec 2021

Minutes: 11:00 - 1:30 @ Computer Lab 2.56

Participants
- Ashton Mitchell
- William Chambers
- Ani Udom
- Dinil Wickramasinghe
- Yash Malik

Goals
- Work on the team project together in person
- Divide into sub-teams to tackle level design (Ani, Dinil, and Yash) and documentation (Ashton and Will)

Team Meeting 14: 07 Dec 2021

Minutes: 11:30 - 12:15 via Microsoft Teams

Participants
- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals
- Close Sprint 3
- Sprint 4 Planning
- Discuss what to present at Planning Meeting

Discussion Topics
- What to present:
    - Show the game design so far
    - Tell customer we decided not to have character selection
- Issue with lose screen (would not appear sometimes) Vinay Kumar Gupta
- Glitch in Healer character (Was healing enemies and buildings) Vladislav Shishkin
- Sprint 3 Retrospective discussion (see Sprint 3 Retrospective)

Sprint 3 Requirements List

Explanation of Changes to Requirements List

- In Team Meeting 12, the team decided it was not feasible and not necessary to implement the character selection feature, so R2.1 was removed.
    - Following on from this decision, the team decided that the player would “unlock” characters as they progressed through the levels. In order for the player to understand how to use each of the characters and to know which character is which, character introduction/instruction scenes are required before each relevant level. R4.1 was created for this.
- In Team Meeting 12, it was realized that the “options” button on the main menu screen was not useful for anything, so R2.2 was updated to reflect the removal of this feature.

Updated List Version 3

As each requirement is met, the corresponding test or design element will be listed in parentheses at the end of the requirement.

Key: ~~Strikethrough~~ indicates removed requirement.

R1.1: ~~Dungeon themed game~~
R1.2: Game that will run on a laptop
R1.3: ~~Game that will run on a phone~~
R1.4: ~~A “bot” that can play the game on it’s own or against other bots~~
R1.5: Ability to understand how well the player is doing in the game (DE1.2)
R1.6: Game that has increasing difficulty through progression (DE3.2)
R1.7: Gameplay modelled after real-time strategy (RTS) (see RTS Style Game Storyboard)
R2.1: ~~Character selection feature as part of gameplay strategy~~
R2.2: Main menu screen with title and “START” button (see DE3.1)
R2.3: Cut scene to establish game backstory (DE2.2)
R2.4: Level selection functionality (DE2.5, DE3.2)
R2.5: 6 levels (DE3.2)
R2.6: Camera zoom functionality (Test Case for US2-3)
R2.7: Enemy characters for player to battle against (DE1.2, DE3.5)
R2.8: Simplistic “cube” character models for cleaner, unique design (DE3.5)
R2.9: Game music throughout to enhance story and theme
R3.1: Clear distinction between character types (DE3.5)
R3.2: Tutorial level to familiarise player with gameplay (DE3.4)
R3.3: Win/lose screens to inform player of level outcome (DE3.7, DE3.8)
R4.1: Character introductions/instructions

Sprint 3 Backlog

B3.1: Level Design (2/3 tutorial + 3 normal levels) (US3-1)
B3.2: Lose Scene Design (US3-2)
B3.3: Win/Lose Scene Logic (US3-4)
B3.4: Character Selection UI (US3-5)
B3.5: Win Scene Design (US3-3)
B3.6: Character Selection logic (US3-6)
B3.7: Create game music (US3-7)
B3.8: Add music to game (US3-7)
B3.9: In the lose scene have a restart level button (US3-2)
B3.10: In the win scene have a new level button (US3-3)
B3.11: Have main menu button at the end scene of every level (US3-2, 3-3)

Complete Backlog Tasks

B3.1: Level Design (2/3 tutorial + 3 normal levels) (DE3.4)
B3.2: Lose Scene Design (DE3.7)
B3.3: Win Scene Design (DE3.8)
B3.7: Create game music
B3.3: Win/Lose Scene Logic
B3.9: In the lose scene have a restart level button
B3.16: Change Intro scene to include game name (DE3.1)

New Backlog Tasks

B3:12 Create user manual
B3.13: Create maintenance guide
B3.14: Create Installation guide
B3:15: Add character description intro scenes
B3.16: Change Intro scene to include game name

Sprint 3 Exception Handling

- The customer cancelled the Customer Meeting for this week, which left our team unable to get feedback on our progress. This was particularly challenging to have happen during this sprint, as a great amount of progress and some big changes had been made. To mitigate this, we held a team meeting in place of the customer meeting (see Team Meeting 12) to discuss the current state of the game and make decisions regarding character design (i.e. colour choice) and level design. We also decided to have another team meeting (see Team Meeting 13) dedicated to working together on the project in person in order to make the most progress.
- Continuing on from last sprint, there were also other conflicting university deadlines that threatened to cause slowdowns with progress this sprint. However, the team was able to communicate better based on what was learned from the shortcomings in Sprint 2, and tasks were better delegated based on work-load resulting in the most productive sprint yet.
- When the sub-team was working on level design, they discovered that the player's own Healer characters would incorrectly heal the enemies and enemy buildings' health. Through the level design team's rigorous testing, this problem was able to be quickly addressed and fixed.
- There was another issue with communication, when 2 team members completed the same task thus wasting work. This was addressed in Team Meeting 12 so that it wouldn't happen again.

Sprint 3 Product Documents

User Stories

User Story 3-1: Level Design

AS A	*Acceptance Criterion:*
Player *I WANT TO* Have a variety of levels with increasing difficulty. Starting with a few intro level to learn about characters and buildings *SO THAT* I can learn how to play the game and be engaged	Different levels with win/lose criterion.

User Story 3-2: Lose Scene

AS A	*Acceptance Criterion:*
A player *I WANT TO* To have as screen that pops up to notify me if I have lose the level *SO THAT* I can know if I have lost and have the option to restart the level or go back to the level selection screen	A screen that pops up when the player has lost notifying them. **(See US3-4)** *Comments:* There will be two buttons on the screen. One that restarts the level and one that takes the player back to the level selection screen.

User Story 3-3: Win Scene

AS A Player *I WANT TO* To have a screen that pops up to notify me when I have won the level *SO THAT* I can know when I have won and have the option to go to the next level or go back to the level selection screen.	*Acceptance Criterion:* A screen pops up when the player has won, notifying them. **(See US3-4)** *Comments:* There will be two buttons on the screen. One that takes the player to the next level and one that takes the player back to the level selection screen.

User Story 3-4: Win/Lose scene logic

AS A Developer *I WANT TO* I want to create a criteria for winning and losing each level *SO THAT* The player has a clear objective of the game	*Acceptance Criterion:* A logic that works and produces the appropriate win/lose scene. *Comments:* The criteria for winning the game would be destroying the enemy core building before player characters lose their health **(See UC 2.3).**

User Story 3-5: Character Selection UI

AS A Player *I WANT TO* Have a way to select the characters I want in my army depending on the level design *SO THAT* I can create a strategy surrounding what type of characters I think would help me leave the level.	*Acceptance Criterion:* A way for players to pick characters for each level *Comments:* There will be a predefined limit to the army size based on weighting system we will give to the player characters **(See US3-6)**

User Story 3-6: Character Selection Logic

AS A Developer *I WANT TO* Create a limit for how many characters a player can choose on each level. Based on a character weighting system. *SO THAT* The player cannot choose more than a certain number of players	*Acceptance Criterion:* Logic works and players cannot exceed the predefined army limit.

User Story 3-7: Create game music

AS A Player *I WANT TO* Have background music in the game *SO THAT* The game is more engaging	*Acceptance Criterion:* Music that plays when players starts the game

User Story 3-8: Restart Level button on lose scene

AS A Player *I WANT TO* Have a button that lets me restart the level a lost *SO THAT* I can try again to win it	*Acceptance Criterion:* A working restart level button that starts the level from the beginning

User Story 3-9: New Level button on win scene

AS A Player *I WANT TO* Have a button the takes me to a new level after I have won the previous one *SO THAT* I can progress in the game	*Acceptance Criterion:* A new level button that takes a user to the level that immediately follows the one they were on

User Story 3-10: Have level selection button at the end of each level

AS A Player *I WANT TO* Have a button that takes me back to the level selection screen *SO THAT* I can see my game progress or to replay levels I have already won	*Acceptance Criterion:* A working level selection button that takes the user to the level selection button

Revised User Stories - Final Set

Halfway through this sprint we realised that creating the logic and UI for character selection may be too complicated for this version of our game. Following agile principles, we believe in keeping the game design as simple as possible to successfully meet the minimum requirements. We instead decided to design the levels with a pre-decided “army” of characters. We felt confident this still gave the player room to be strategic in how they moved the characters around the map and there were multiple ways for the players to win/lose. This decision was

going to be proposed to the customer during this sprint's customer meeting, however, as the customer meeting was unexpectedly cancelled this sprint, the team decided to pursue this change and discuss with the customer at the next possible customer meeting. Therefore, US3-5 and 3-6 were removed, leaving the final set of User Stories for Sprint 3 as follows:

User Story 3-1: Level Design
User Story 3-2: Lose Scene
User Story 3-3: Win Scene
User Story 3-4: Win/Lose scene logic
User Story 3-7: Create game music
User Story 3-8: Restart Level button on lose scene
User Story 3-9: New Level button on win scene
User Story 3-10: Have level selection button at the end of each level

Requirements Use Cases

RU1.2 Use case for US3-5 and 3-6: Character Selection Feature
Scope: Game level
Level: functional level
Context: Players can select the quantity of different character types they wish to comprise their army and beat each level.
Frequency of Occurrence: At the start of each level, after the player has the chance to preview the level design
Open Issues: How much space does each character take up in the army selection? Based on Health Level or Attack Power or both? In which level is each character introduced?

RU3.1: Use case for US3-2: Lose scene (also references US3-8 and US3-10)
Scope: End of level
Level: functional level
Context: If a player loses a level we need to have a game scene appear that notifies them of this and lets them choose what to do next.
Frequency of occurrence: Any time a level is lost
Open Issues: What buttons to include in this scene? What are the conditions for losing? Can they quit the game at this point?

RU3.2: Use case for US3-3: Win scene (also references US3-9 and US3-10)
Scope: End of level
Level: functional level
Context: If a player loses a level we need to have a game scene appear that notifies them of this and lets them choose what to do next.
Open Issues: What buttons to include in this scene? What are the conditions for winning? Can they quit the game at this point?

Design Use Cases

- Use Case 2.1 was updated during this sprint to remove the “options” button from the main menu screen. More detail explaining this change is outlined in Team Meeting 12 and is reflected in the changes to the Sprint 3 Requirements List.
- New Use Cases 3.1 and 3.2 were created.

Updated Use Case 2.1:

UC1-1**: Use Case:** Flow from Start Screen to Gameplay

UC1-2: **Author:** AM

UC1-3: **Purpose:** Establish the flow of screens that the player can navigate through before beginning gameplay in level 1.

UC1-4: **Overview:** Starts when the player runs the game from a computer. The intro screen appears first, with background music and 2 button options to click: **alternative 1:** the player can click the “START” button, or **alternative 2**: the player can click the “EXIT” button. **Alternative 1** will lead to the cutscene backstory video, then to the instruction screen. After reading the instructions, the player must click “NEXT” to progress to the level selection screen. At the level selection screen, the player can **alternative 1.1**: click on “Level 1”, as all other levels will be locked until beating level 1, or **alternative 1.2**: click “Return to Main Menu”, where they can restart their flow of navigation from the intro screen. **Alternative 2** will exit the player out of the game entirely.

UC1-5: **Cross References:** US 1-3.2, 1-8, 2-1, 2-2, 2-9, 2-11

UC1-6: **Actors:** Player

UC1-7: **Pre-Conditions:**

UC1-Pre-1: The player successfully runs the game from their computer by having all of the necessary applications installed properly

UC1-8: **Post-Conditions:**

UC1-Post-1: The player is able to navigate through the initial screens to **alternative 1**: begin gameplay or **alternative 2** exit out of the game.

Use Case 3.1:
UC1-1**: Use Case:** Lose scene
UC1-2: **Author:** AU
UC1-3: **Purpose:** Have a scene that appears when player has lost the level based on US3-4
UC1-4: **Overview:** Starts when the player has lost the level based on the lose condition: all player characters are defeated before the 'enemy core' building is destroyed.

A scene appears that reads 'you lose' with 2 button options. **Alternative 1:** The player selects the 'restart level' button, this takes them to the start of their current level and they can try and win again. **Alternative 2:** The player selects the 'Level select' button and they are returned to the level selection screen where they can, **alternative 2.1:** choose a previous level they have played if they wish or the level which they have just lost or **alternative 2.2:** exit the game.

UC1-5: **Cross References:** US2-2, UC2.3, US3-2, US3-8, US3-10, RU3.2
UC1-6: **Actors:** Player
UC1-7: **Pre-Conditions:**

UC1-Pre-1: The player has lost the level

UC1-8: **Post-Conditions:**

UC1-Post-1: The player can **alternative 1:** restart the level or **alternative 2:** go to the level selection screen.

Use Case 3.2:
UC2-1**: Use Case:** Win scene
UC2-2: **Author:** AU
UC2-3: **Purpose:** Have a scene that appears when player has won the level based on US3-4
UC2-4: **Overview:** Starts when the player has won the level based on the win condition: enemy core building has been destroyed.
A scene appears that reads 'you win' with 2 button options. **Alternative 1:** The player selects the 'new level' button, this takes them to the start of the immediate next level. **Alternative 2:** The player selects the 'Level select' button and they are returned to the level selection screen where they can, **alternative 2.1:** choose a previous level they have played if they wish or the next level that has just been unlocked, **alternative 2.2:** exit the game.
UC2-5: **Cross References:** US2-2, UC2.3, US3-3, US3-9, US3-10, RU3.1
UC2-6: **Actors:** Player
UC2-7: **Pre-Conditions:**

UC1-Pre-1: The player has won the level

UC2-8: **Post-Conditions:**

UC1-Post-1: The player can **alternative 1:** go to the next level or **alternative 2:** go to the level selection screen.

Tests

Test Case for US 3-1:
Levels - Multiple levels

Precondition: A Unity game exists that has multiple scenes including an accessible level selection screen and multiple game levels.

Test steps:
1. Launch a built version of the game.
2. Navigate to the level selection screen by pressing start on the start screen
3. Select different levels on the selection screen
4. Observe if different playable levels load when selecting different levels on the level selection screen

Expected Result: There are multiple different game levels that have a variety of landscape objects, buildings and enemy characters. These are available to access from the level selection screen.

Test Case for US 3-2:
Lose Screen - A lose screen that pops up when the player has lost to notify them that they have lost.

Assumption: The user knows how to play a game level.

Precondition: A Unity game exists that has a playable game level.

Test steps:
1. Launch a game level scene either as a built game or in the Unity project.
2. Move the player characters to approach a group of enemies one at a time to try and ensure the player characters are defeated.
3. Once all the player characters are defeated observe what happens.

Expected Result: When all the player characters are defeated within a game level thelose screen will pop up.

Test Case for US 3-3:
Win Screen - A Win screen that pops up when the player has won a level to notify them that they have won.

Assumption: The user knows how to play a game level and win.

Precondition: A Unity game exists that has a playable game level.

Test steps:
1. Launch a game level scene either as a built game or in the Unity project.
2. Win the level by destroying the core building without having all the player character be destroyed first.
3. Once the core building has been destroyed observe what happens.

Expected Result: When the core building has been destroyed a win screen should pop up.

Test Case for US 3-8 and US 3-10:
Lose Screen Buttons - A lose screen that pops up when the player has lost to notify them that they have lost and allows them to restart the game or go back to the level selection screen.

Assumption: The user knows how to play and lose a game level.

Precondition: A Unity game exists that has a playable game level.

Test steps:
1. Launch a game level scene either as a built game or in the Unity project.
2. Lose the game level by having all your player characters destroyed by the enemy.
3. Click on the restart button on the lose screen and observe what happens.
4. Lose another game level and click on the select level button and observe what happens.

Expected Result: Clicking on the restart button on the lose screen restarts the current level. Clicking on the select level button on the lose screen loads the level selection screen.

Test Case for US 3-9 and US 3-10:
Win Screen Buttons - A win screen that pops up when the player has won to notify them that they have won and allows them to start the next level or go back to the level selection screen.

Assumption: The user knows how to play and win a game level.

Precondition: A Unity game exists that has a playable game level.

Test steps:

1. Launch a game level scene either as a built game or in the Unity project.
2. Win the game level by destroying the core building.
3. Click on the next level button on the win screen and observe what happens.
4. Win another game level and click on the select level button and observe what happens.

Expected Result: Clicking on the next level button on the win screen starts the next level. Clicking on the select level button on the win screen loads the level selection screen.

Test Case for US 3-7:

Music - Background music plays during game

Assumption: The user knows how to navigate through the game. The user has audio playback enabled for the game level.

Precondition: A Unity game exists that has a playable game level.

Test steps:

1. Launch a unity game level scene.
2. Observe the audio from the game.

Expected Result: While playing a game level there is background music.

CRC Cards

CRC card links to US3-2, 3-3

Class Name: LevelEndScreen	
Responsibilities: ● Display win/end screen for the user ● Helps to manage the levels	**Collaborators:** ● GameObject (unity class)

CRC card links to US3-5, 3-6

Class Name: MultiSelect	
Responsibilities: ● Draws the rectangle around the selected characters	**Collaborators:** ● Texture 2D (unity class)

Class Name: UnitStatDisplay	
Responsibilities: ● Displays the unit's health ● Manages the health ● Destroys of the unit	**Collaborators:** ● Image (unity class)

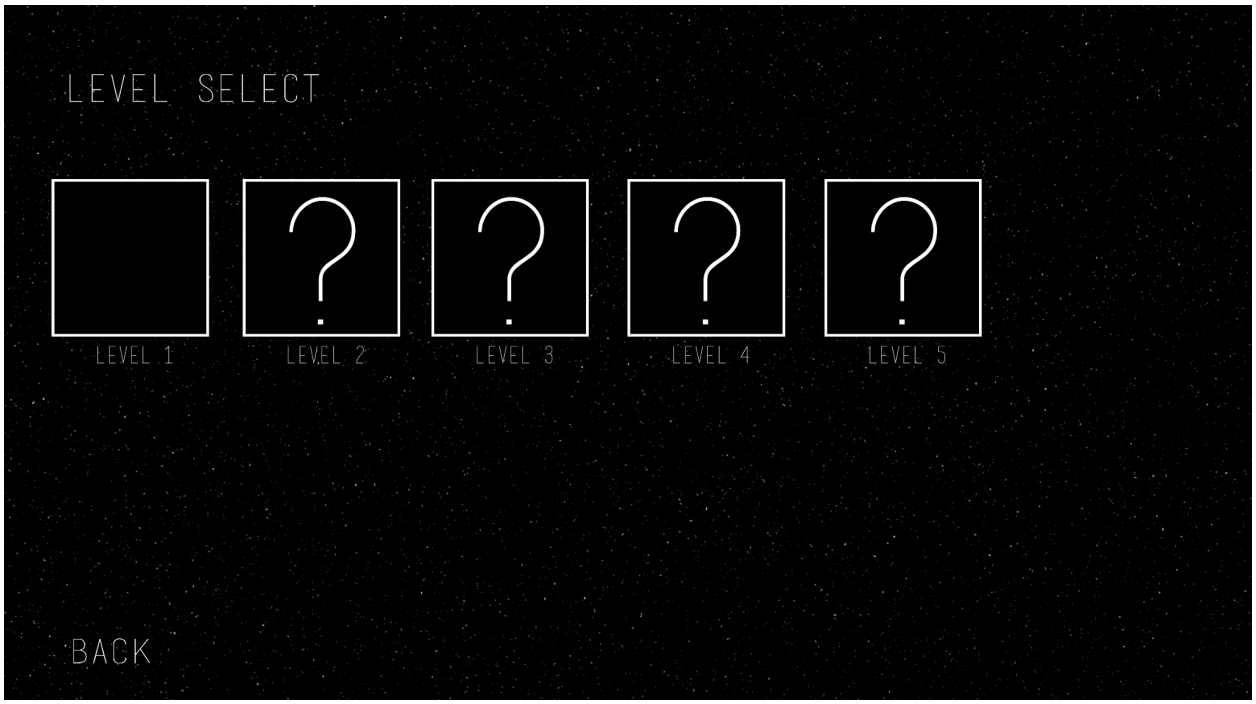
User Interface Design Elements

Design Element 3.1: Intro Screen Design Version 2

UI Design: Here we use the same design as in Sprint 1, but with the rebrand to “SquareWars” and the omission of the options button.

UX Design: User experience is similar to that of the Sprint 1 main menu. Inverted pyramid structure with buttons decreasing in size to levels of importance.

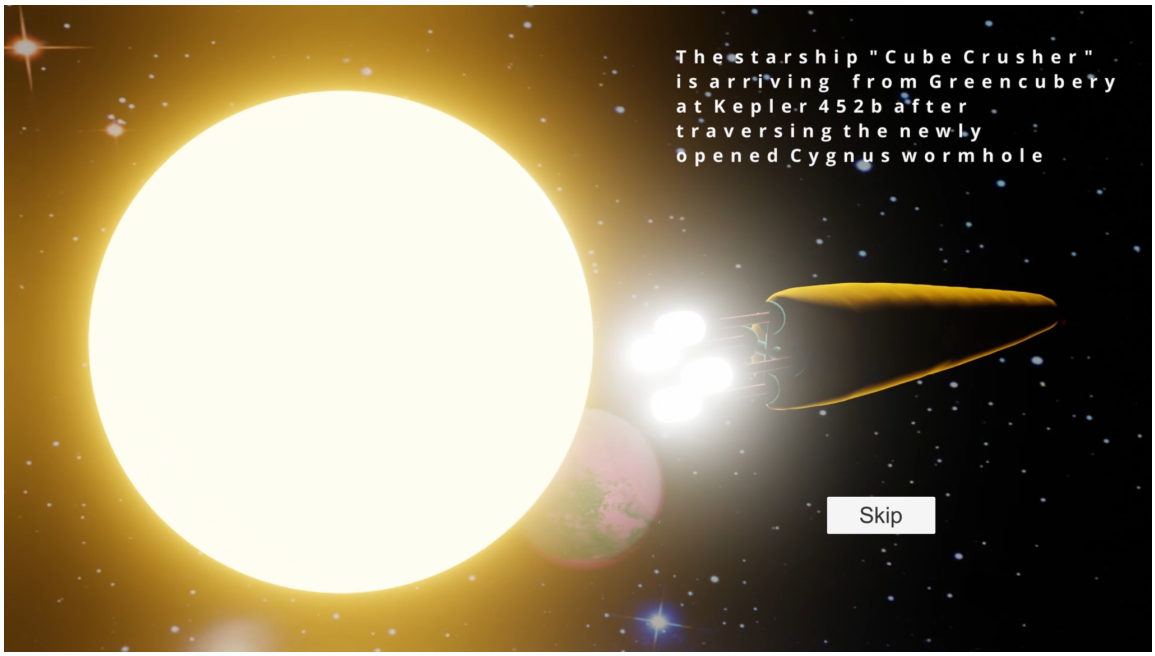
Design Element 3.2: Level Select Screen Design Version 2



UI Design: Here we use the same level select template from before. Due to time restraints we will not have a picture of the level on the level button. A simple black and white design is used similar to the main menu, this is also a nice contrast with the levels themselves which are vibrant. A back button is placed on the left of the screen for the user to enter the main menu once again.

UX Design: We have the levels large in the centre of the screen with the appropriate numbering below it. First thing the player will see is these level buttons, and intuitively they will go from left to right. Normally left means backwards and right means forwards in terms of movement so the back button is placed on the button left, so the player subconsciously knows the movement of this button.

Design Element 3.3: Intro Cutscene Video for Game Backstory Version 2

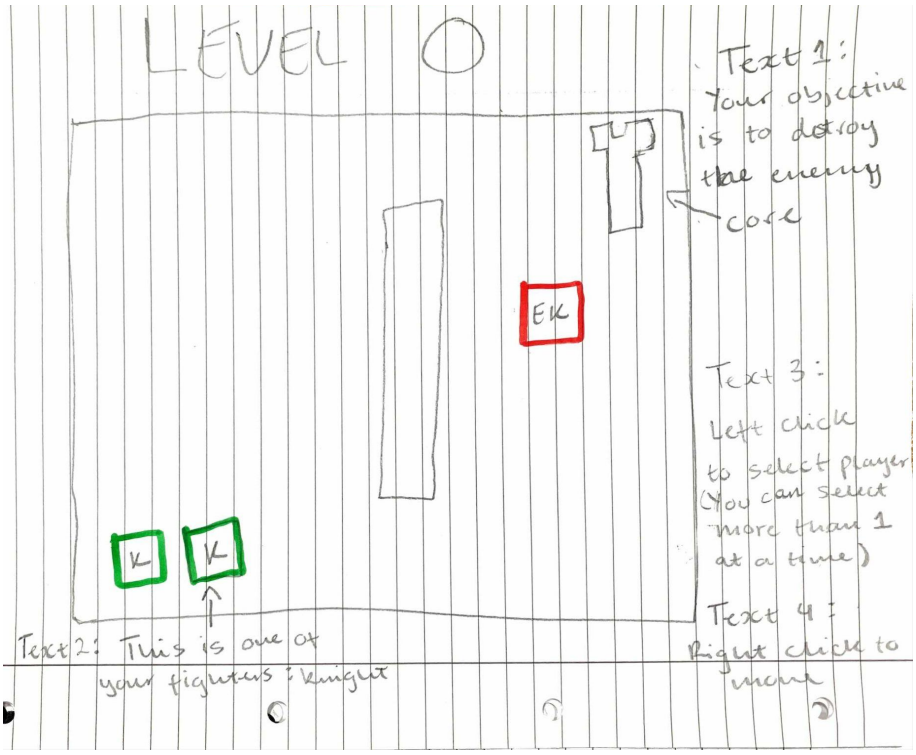


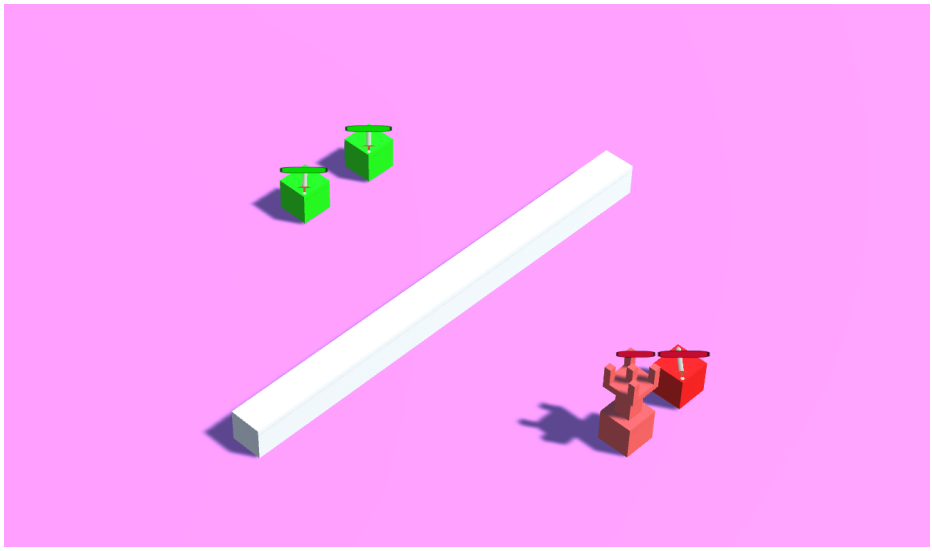
- The intro video was added to the game to satisfy User Story 2-11 and User Story 2-9.
- A Skip button was added to the video so that the user could skip to the game without having to watch the entire intro video. The skip button was designed to look like a button so that the user would be intuitively able to understand that it was interactive and a mouse click would skip the video. The button was made to be in the corner so it didn't obscure the visuals but was made large enough so it was easily visible for the user to identify. A background colour of white for the button was chosen as it was in contrast to the dark background of the video and easily distinguishable at the same time as being discrete enough not to ruin the enjoyment of the video.

Design Element 3.4: Level Design

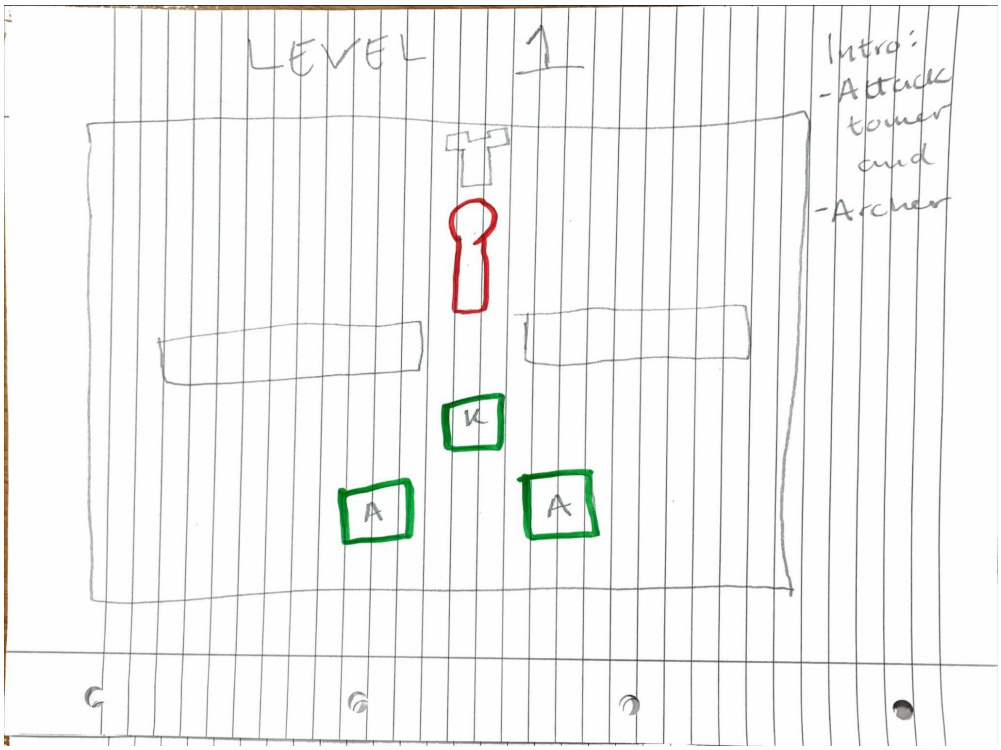
When designing the levels we started by making a rough sketch of each level. Objects outlined in green represent the player characters while objects outlined in red are enemies. Grey rectangles represent boulders (can not be interacted with). After trying out different sketches we then build each level of the game and test how the level works. The following shows initial sketches and how the levels look within the game.

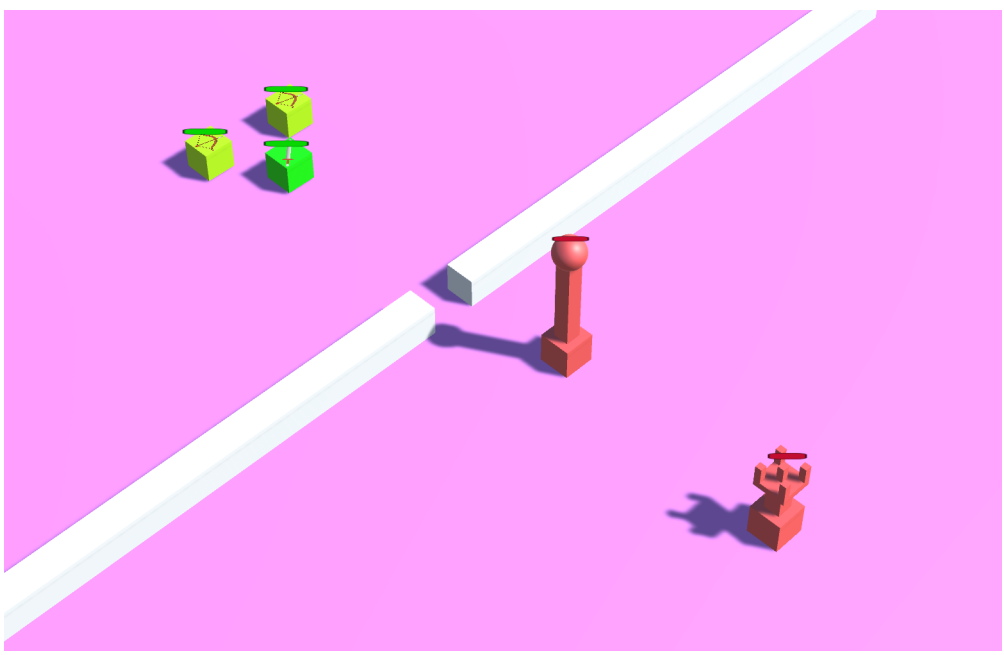
Level 1: Sketch and Final



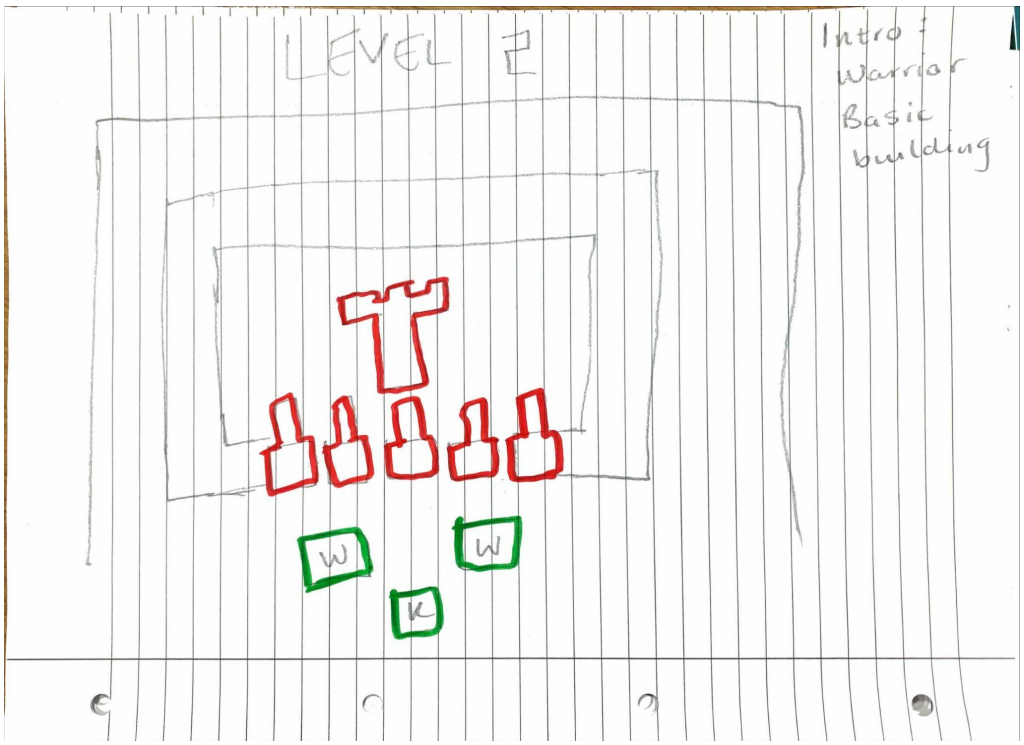


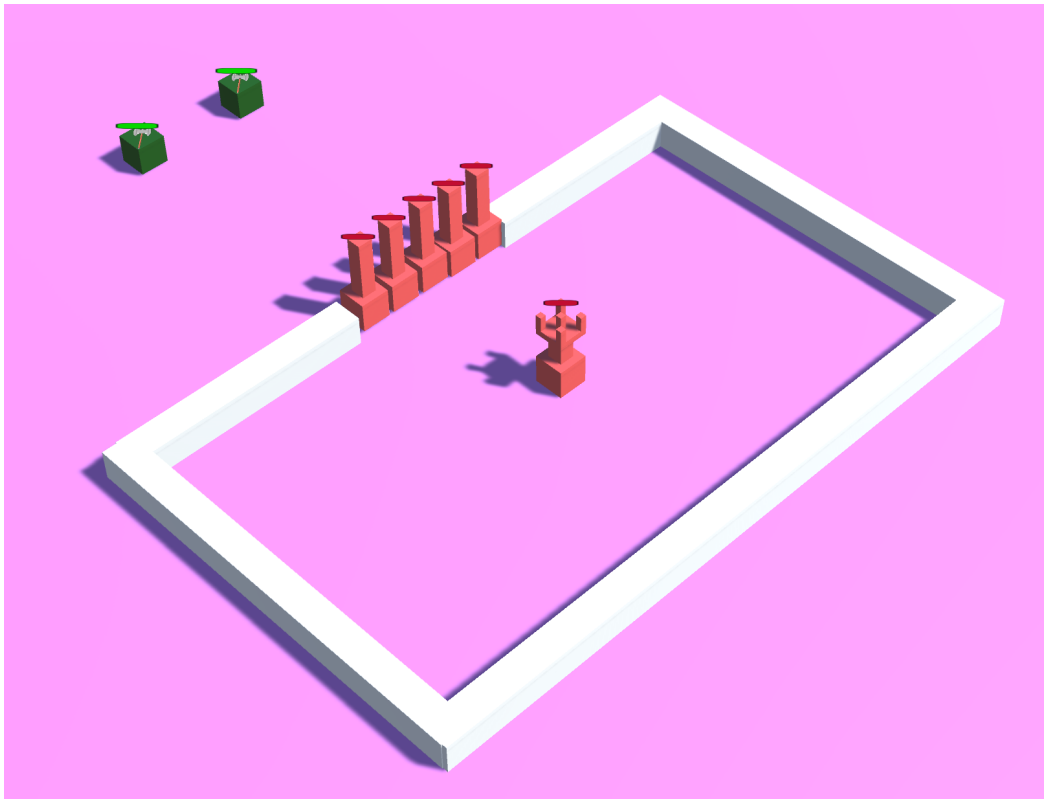
Level 2: Sketch and Final



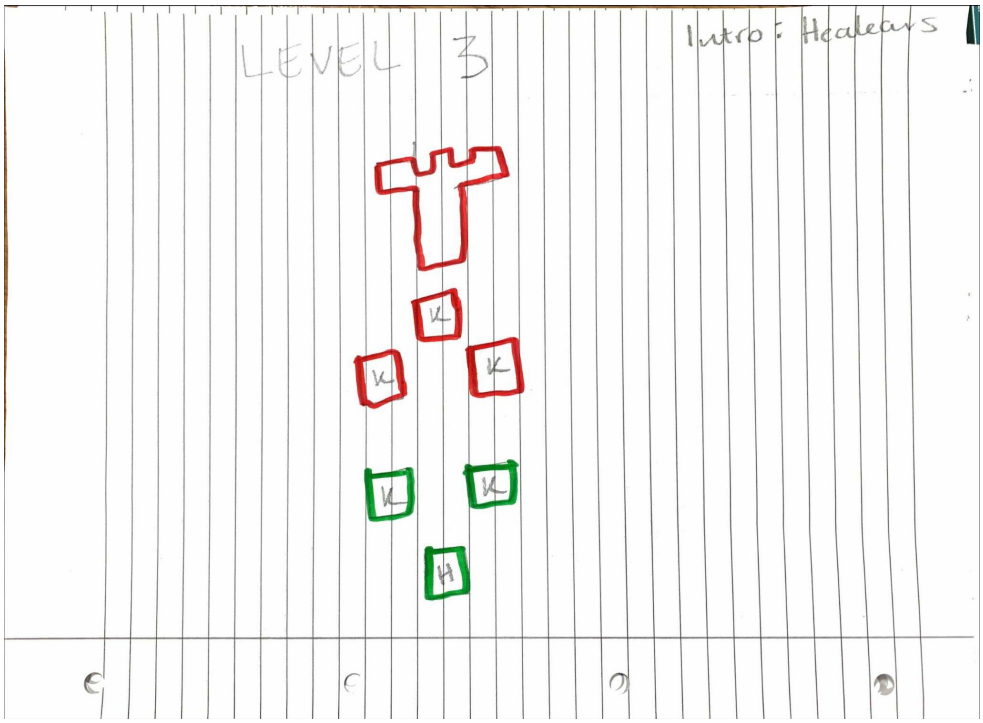


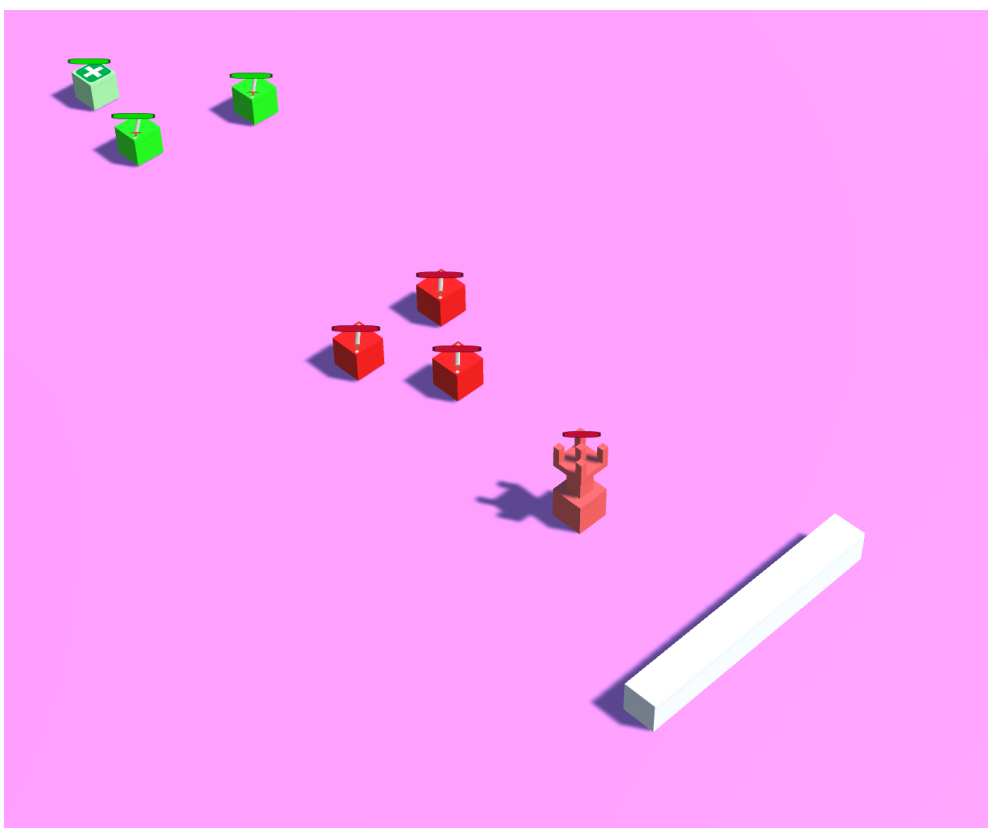
Level 3: Sketch and Final



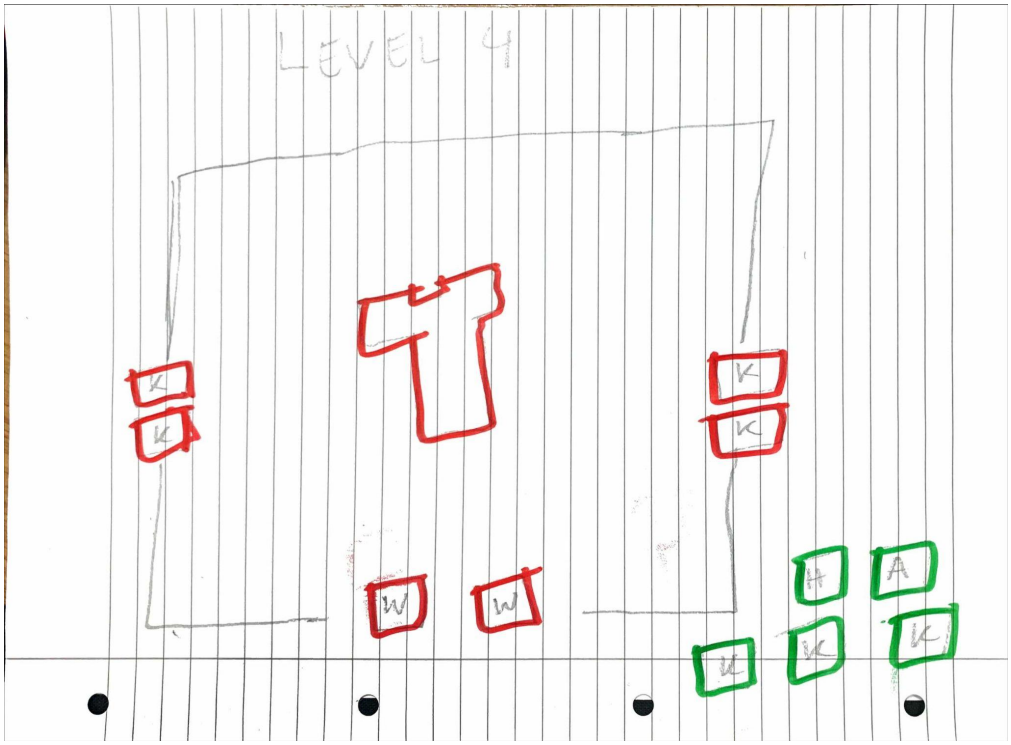


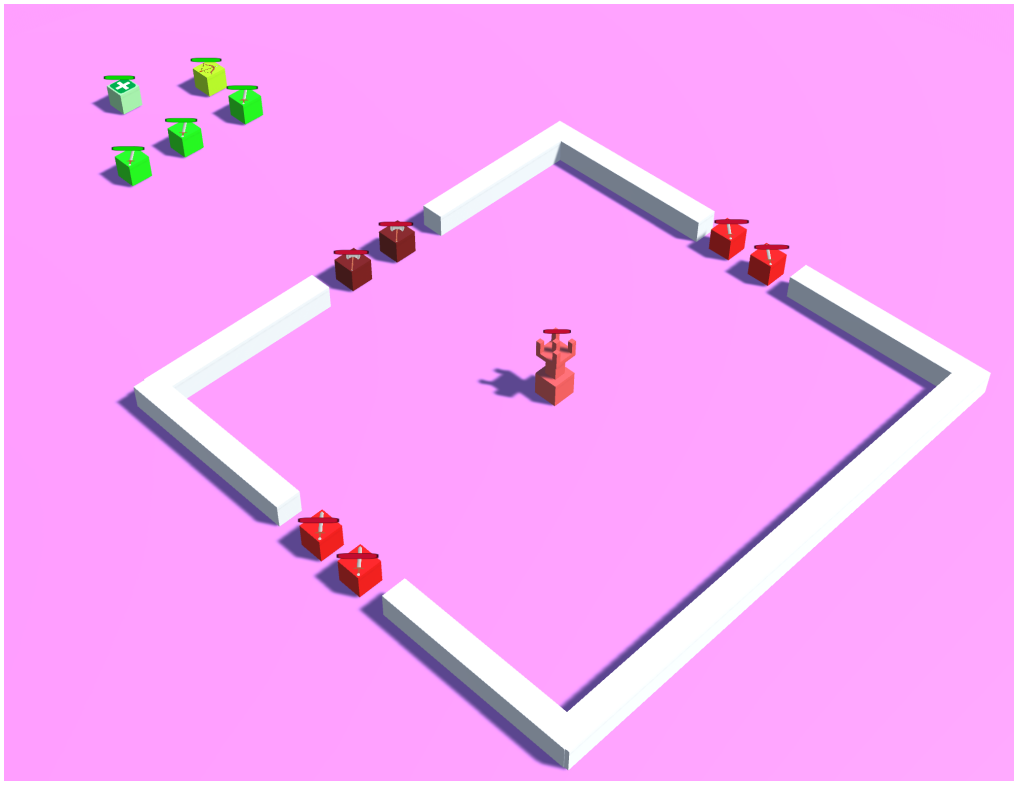
Level 4: Sketch and Final



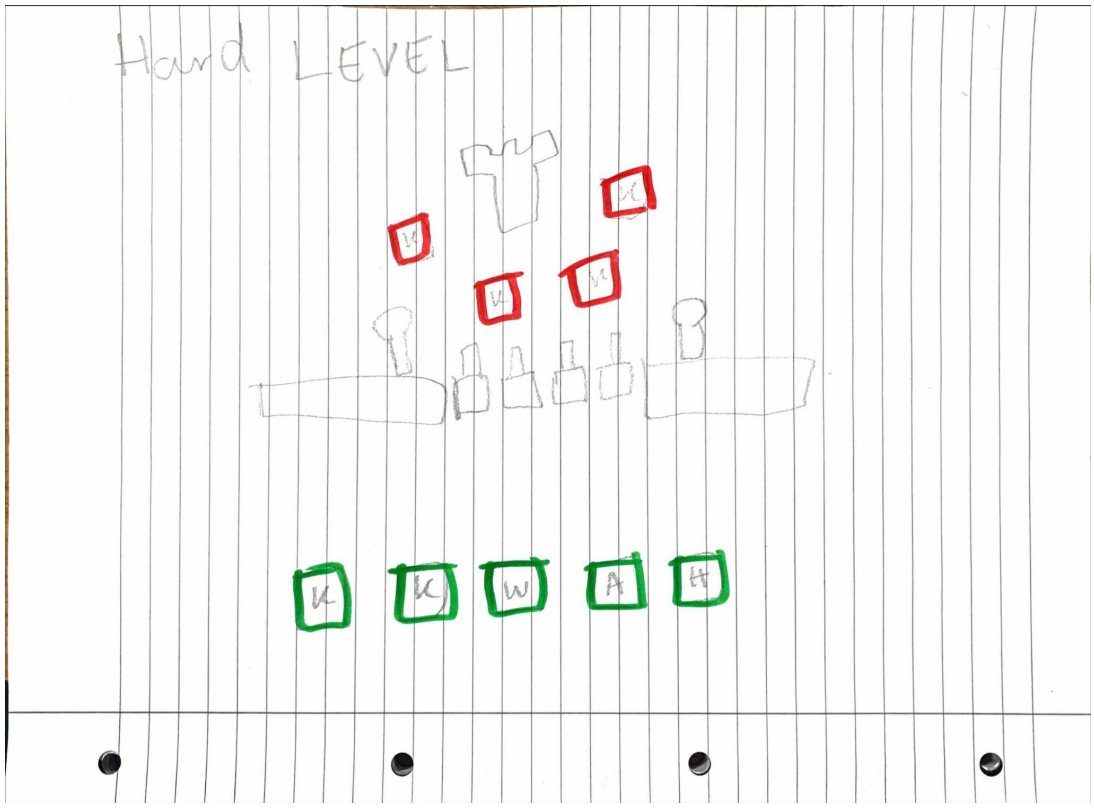


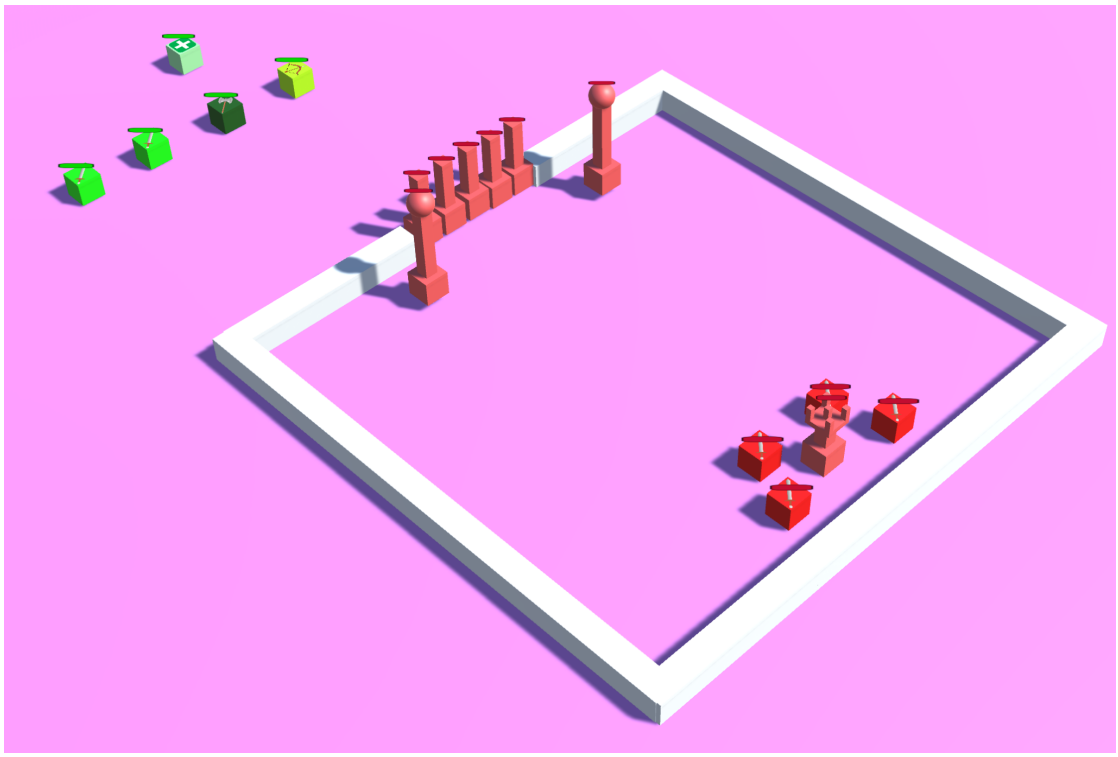
Level 5: Sketch and Final





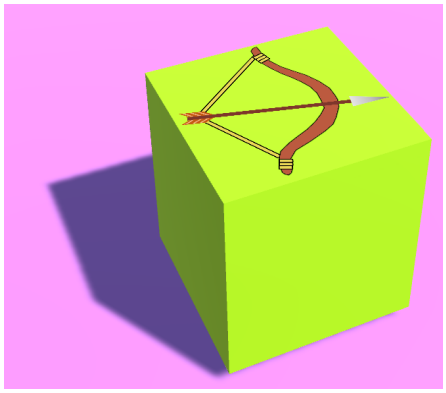
Level 6: Sketch and Final

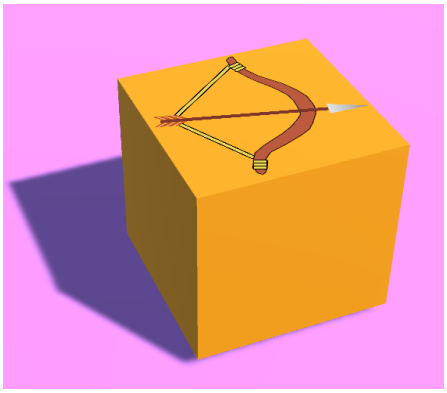




Design Element 3.5: Cube Character Models

Archers

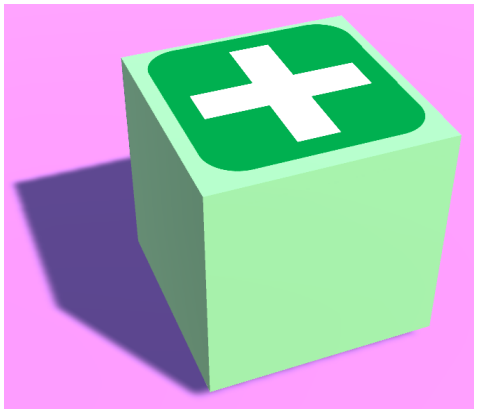


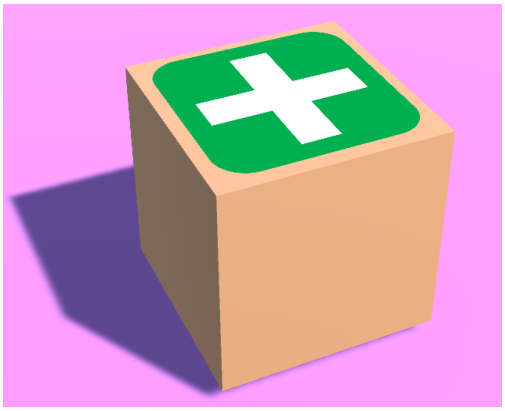


Description: This character can attack from a long range by shooting beams at players/buildings. It is introduced in level 2 along with the enemy attack tower(US1-10). See DE3.6 for the statistics of this object.

Design: Light, slightly askew shades of the green color for the player's Healers and of the red color for the enemy Healers were chosen to distinguish the Archers from the other character types. Bow and arrow symbols were placed on top of these cubes to easily identify them as Archers.

Healers

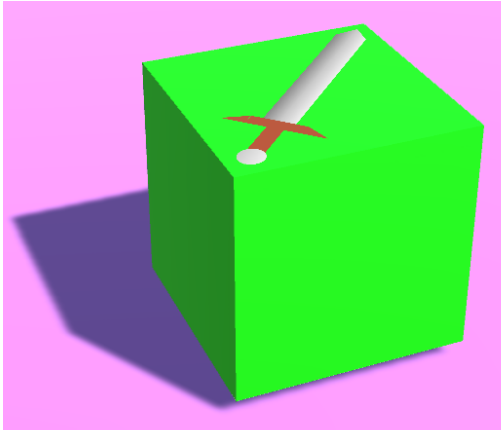


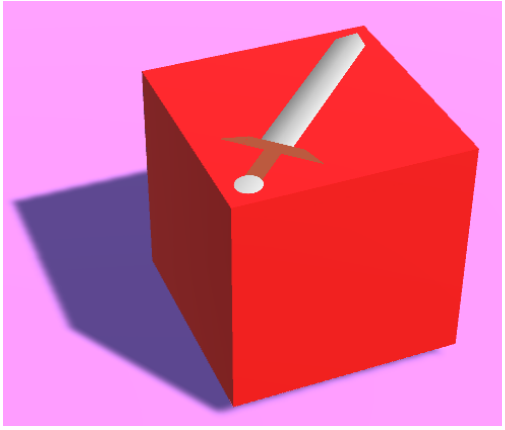


Description: This character is meant to accompany the other characters during the game to replenish their health, has no attack capabilities. Introduced in level 3 of the game (US 2-10). See DE3.6 for the statistics of this object.

Design: Whimsical shades of the green color for the player's Healers and of the red color for the enemy Healers were chosen to represent the magical-like nature of the Healers. Pharmacy cross symbols were placed on top of these cubes to easily identify them as Archers.

Knights

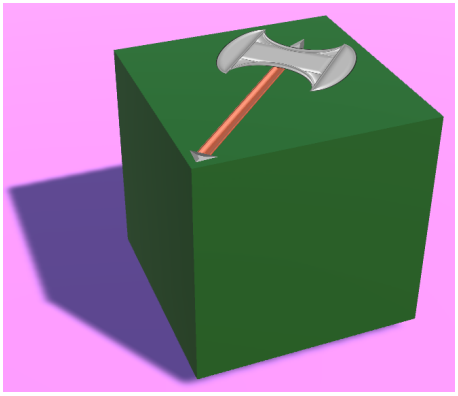


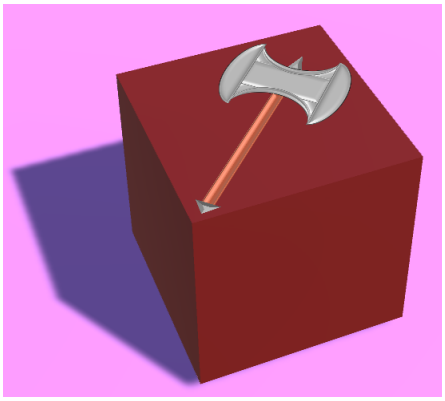


Description: This character has attack capabilities and is meant for close combat, it can destroy other player characters or the buildings. It has the highest movement speed and is introduced in level 1 of the game (US 1-8). See DE3.6 for the statistics of this object.

Design: Bright shades of the green color for the player's Knights and of the red color for the enemy Knights were chosen to show vibrancy and high contrast of the Knights when engaged in close combat. Sword symbols were placed on top of these cubes to easily identify them as Knights.

Warriors





Description: This character is primarily meant to destroy buildings as it has the highest attack and armor. It can also attack other characters, introduced in level 2 of the game (US 1-9). See DE3.6 for the statistics of this object.

Design: Dark shades of the green color for the player's Warriors and of the red color for the enemy Warriors were chosen to represent strength. Axe symbols were placed on top of these cubes to easily identify them as big and bulky fighters.

Design Element 3.6: Table with Object Sats

Type	Health	Armour	Attack	Attack Range	Attack Speed Cool	Cost	Aggro Range	Speed
Enemy Core	100	5	0	0	0	0	0	0
Enemy Attack tower	80	5	10	5	1	0	10	0
Enemy Basic Building	50	5	0	0	0	0	0	0
Barracks	500	25	0	0	0	0	0	0
Knight	50	1	7	1	0.5	10	10	5
Warrior	60	2	10	1	0.14	15	10	3.5
Archer	30	0	8	5	1	15	10	3.5
Healer	40	0	15	1	1	20	5	3.5

Description: Each object in the game was assigned the following:

- A health level: Once this is depleted the object is destroyed (and disappears from screen)
- An armour level: This acts as a defence from attacks
- Attack strength: Determines how powerful the attack is
- Attack range: The distance at which an object can attack another
- Attack speed cool: This is how often objects are able to attack (seconds)
- Aggro Range: This is the proximity that each object locks onto a target for attack.
- Cost: This would be used as a way to create a limit for how many characters a player can select for their army e.g. for a level can have spend of 50 and the total cost of characters selected cannot exceed this
- Speed: How fast objects can move around the plane.

Each object was weighted against each other. During level design, we would change different stats if objects were not interacting the way we wanted them to. This was the final version of the object stats.

Design Element 3.7: Lose Screen

Lose Screen is built to handle use case Use Case 3.1.

Description: If all the player characters die, without destroying the Enemy Core building, then the user loses the game. This triggers the “Lose Screen” pop-up, which shows a Game Over and ‘You Lose’ message. Screen also has two buttons: Restart and Select Level. Pressing the Restart button allows the user to try the level again. Pressing the Select Level button takes the user back to the MapMenu.

Design Element 3.8: Win Screen Version 1

Win Screen is built to handle use case Use Case 3.2.

Description: When the Enemy Core building is destroyed, the user wins the game. This triggers the “Win Screen” pop-up, which shows a ‘Game Over’ and ‘You Win’ message to the user. Like the previous screen, it also has two buttons: Restart and Select Level, both of which function the same.

UI: The background utilises the same one used within the title and Map selection screens. It is a deep black with white speckles. This design was chosen to imitate stars in deep space. The ‘You Lose/Win’ text colour is a calm yellow, with the remaining text being white.

UX: Using the same background across screens provides a consistency between game screens and makes scene changes less jarring. The ‘You Lose/Win’ text is placed centrally to clearly indicate the outcome of the player's game. The colour is yellow to provide a contrast to the black, and stand out from the other white text.

Sprint 3 Game Progress

Having shown the customer the animated Knight character, his preference was to keep the simplicity of the green and red cubes rather than add animated characters, that may add no real additional substance to the game. Because of this, the cubes' designs were updated to ensure a player could easily distinguish between what character type they were controlling. All characters had their corresponding emblems displayed on the top of the cube. The cubes were also shaded to differentiate between the character types, and owner of the character (player or enemy). Player characters are shades of green, whilst enemy characters are shades of red.

From this design decision, the game has been rebranded to 'Square Wars'. An updated title has been added to the main menu screen. An intro cut-scene has also been created to provide a backstory of the game. This has yet to be added to the build however.

Game Mechanics

6 new levels have been designed that introduce each new character. Three 'tutorial' style levels have been designed to specifically identify the strengths and weaknesses of newly introduced characters. The three remaining levels give access to all characters and require the user to be strategic about when and where they play their chosen units.

Win and Lose pages were also created (design elements 3.7 and 3.8). A game is won when the core tower (previously introduced) is destroyed. If this is completed successfully, the player will be greeted with a 'You Win' screen and allowed to advance to the next level or choose to go to the main menu. A game is lost if all player characters are killed before the core tower has been destroyed. If this occurs, the player is met with a 'You Lose' screen which allows the player to retry the level or go back to the main menu.

Sprint 3 Retrospective

- Things that went well:
    - In person meeting for level design was productive
    - Much more efficient as a team in general than the previous sprints, based on completion of set backlog tasks
    - Better at working with Git
    - Better at communication
- Things that went wrong:
    - Customer meeting was cancelled, so not able to check in with the customer to see if the project is on the right path
- Things to work on in the next sprint:
    - Recruit more of the team to help out with documentation

Sprint 4 Overview: 05 Dec 2021 - 12 Dec 2021

- Since Sprint 3 resulted in a mostly finished product, Sprint 4 is focused on making some final updates and finishing touches. The team planned to run through the game in order to identify any additional changes or updates. Additionally, the team planned to use the Customer Meeting in this sprint to verify that the current state of the game is to a point that the customer is happy with. Any suggestions from the customer during this sprint could still be implemented in the remaining time for the project.

Sprint 4 Meeting Minutes

Customer Meeting 5: 08 Dec 2021

Minutes: 11:15 – 11:30 @ Chancellor's Building 5.13

Participants
- William Chambers
- Ashton Mitchell
- Ani Udom
- Vladislav Shishkin
- Vinay Kumar Gupta
- Yash Malik
- Customer

Goals
- Discuss change in game strategy with customer due to time constraints
- Demo current version of the game and receive feedback to implement in current sprint

Discussion topics
- Discuss change in strategy with customer: player doesn't choose characters (removing character selection feature and giving certain set of characters each level), instead player chooses when and where to move their characters to attack
    - Necessary change due to time constraints
    - Customer is content with this change, confirms team decision from sprint 3
- Customer has concern that he is unable to discern which character is firing at which (enemy at character or character at enemy?)
    - Customer wants us to work on visual cues/feedback
    - Health bars are good - may investigate making the enemy health bar different from character health bar for added clarity
- Suggestion to add instruction screen so that players can understand the gameplay and strategy instead of being forced to learn by trial and error
    - The team decided this was a good idea to implement

Team Meeting 15: 10 Dec 2021

Minutes: 10:00 - 13:00 remotely

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals

- Everyone to run through the game and suggest any edits or implementations
- Designated working time for the team project
- Present any completed work to the team for review
- Each team member to announce the specific tasks they are working to complete

Team Meeting 16: 12 Dec 2021

Minutes: 14:00 - 15:15 via Microsoft Teams

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals

- Sprint 4 retrospective discussion
- Sprint 5 planning meeting
- Discuss what to present to customer at upcoming meeting

Sprint 4 Requirements List

Explanation of Changes to Requirements List

- Based on feedback from Customer Meeting 5, it was suggested that an instruction screen would be useful for players to understand the gameplay and strategy. R5.1 was created to reflect this desire.

Updated List Version 4

As each requirement is met, the corresponding test or design element will be listed in parentheses at the end of the requirement.

Key: ~~Strikethrough~~ indicates removed requirement.

R1.1: ~~Dungeon themed game~~
R1.2: Game that will run on a laptop
R1.3: ~~Game that will run on a phone~~
R1.4: ~~A “bot” that can play the game on it’s own or against other bots~~
R1.5: Ability to understand how well the player is doing in the game (DE1.2)
R1.6: Game that has increasing difficulty through progression (DE3.2)
R1.7: Gameplay modelled after real-time strategy (RTS) (see RTS Style Game Storyboard)
R2.1: ~~Character selection feature as part of gameplay strategy~~
R2.2: Main menu screen with title and “START” button (see DE3.1)
R2.3: Cut scene to establish game backstory (DE2.2)
R2.4: Level selection functionality (DE2.5, DE3.2)
R2.5: 6 levels (DE3.2)
R2.6: Camera zoom functionality (Test Case for US2-3)
R2.7: Enemy characters for player to battle against (DE1.2, DE3.5)
R2.8: Simplistic “cube” character models for cleaner, unique design (DE3.5)
R2.9: Game music throughout to enhance story and theme (Test Case for US3-7)
R3.1: Clear distinction between character types (DE3.5)
R3.2: Tutorial level to familiarise player with gameplay (DE3.4)
R3.3: Win/lose screens to inform player of level outcome (DE3.7, DE3.8)
R4.1: Character introductions/instructions (DE4.2)
R5.1: Instruction screen

Sprint 4 Backlog

B3.12: Create user manual (US5-1)
B3:13: Create maintenance guide (US5-2)
B3:14: Create Installation guide (US5-3)
B3.15: Add character description intro scenes (US4-1)
B3.8: Add music to game
B3.10: In the win scene have a new level button
B3.11: Have main menu button at the end scene of every level
B4.1: Level lock mechanism (US4-2)
B4.2: Add in the cutscene video to the game.
B4.3: Mechanism to adjust game resolution depending on the screen it is viewed on

Complete Backlog Tasks

B3.8: Add music to game
B4.2: Add in the cutscene video to the game.
B3.15: Add character description intro scenes (DE4.2)
B3.11: Have main menu button at the end scene of every level
B3.10: In the win scene have a new level button (DE4.1)
B4.3: Mechanism to adjust game resolution depending on the screen it is viewed on

Sprint 4 Exception Handling

- As the customer meeting was cancelled in Sprint 3, the team was forced to make a critical change to the game strategy without being able to consult the customer. So, in the current sprint we needed to sell this change to the customer. As documented in Customer Meeting 5, the customer was happy with the change to strategy as we did well to retain the essence of the gameplay that the customer desired.

Sprint 4 Product Documents

User Stories

User Story 4-1: Character intro scenes

AS A Player *I WANT TO* Have a description of characters and building when they are first introduced in levels *SO THAT* I have an understanding of what the objects do and know how to use them to win.	*Acceptance Criterion:* A scene that appears at the start of any level that introduces a new character. *Comments:* The scene will have a description of the character and what they are good at.

User Story 4-2: Level lock mechanism

AS A Player *I WANT TO* Have levels locked until I win the level before it *SO THAT* I have a greater challenge and more reason to play levels that are harder.	*Acceptance Criterion:* Level locked initially and unlocked when players win a level.

User Story 4-3: Mechanism to adjust game resolution

AS A Player *I WANT TO* Have a game resolution that is appropriate to whatever device I am playing on *SO THAT* The game can fit the screen on different devices.	*Acceptance Criterion:* Game automatically adjusts to fit any screen size.

Development Tasks

T4.1 for US4-1: Character Intro Scenes

- Design scenes that introduce characters with a picture and brief description of how they act
- The scenes should be placed at the start of the levels where the characters are first introduced

T4.2 for US4-2: Level lock mechanism

- Create a condition that locks levels until the level before has been won.

T4.3 for US4-3: Mechanism to adjust game resolution

- Create a condition for the game to adjust size based on the size of the screen that it is being played on.

Requirements Use Cases

RU4.1 Use Case for US4-3: Level Lock mechanism
Scope: Game level
Level: Functional level
Context: Each level of the game after level 1 will be locked until the player wins the unlocked level.
Frequency of occurrence: Each time a level is won a new level is unlocked
Open Issues: Do the levels ‘relock’ if the player exits the game?

Design Use Cases

Use Case 4.1:
UC1-1**: Use Case:** Level lock mechanism
UC1-2: **Author:** AU
UC1-3: **Purpose:** Each level of the game after level 1 will be locked until the player wins the unlocked level.
UC1-4: **Overview:** Starts when the player begins the game and views the level selection screen.
Only the 1st level will be playable.
Once that level is won, the next level is unlocked. This continues as the player progresses throughout the game.
UC1-5: **Cross References:** US2-2, UC2.1, UC3.1, UC3.2, US4-2, RU4.1
UC1-6: **Actors:** Player
UC1-7: **Pre-Conditions: (include how many pre and post-conditions are necessary, if any)**
UC1-Pre-1: Starting the game
UC1-Pre-2: Winning level 1

Tests

Test Case for US 4-1:
Character intro - Pop up scene that appears at the start of levels to introduce new characters to the player.

Assumption: The user knows how to navigate through the game menus to start a level.

Test steps:
1. Run a built version of the unity game.
2. Start the first game level.
3. Observe if a character intro scene pops up.
4. Click on start game on the pop up window.
5. Repeat for all the remaining levels which introduce new characters.

Expected Result: While starting a game level a screen pops up over the game which provides a description of any new characters that are included in the level. When you click on the Start Game button on the screen the screen disappears allowing the game to be played.

Test Case for US 4-2:
Level lock and unlock - Levels are unlocked after completing previous levels.

Assumption: The user knows how to navigate through the game menus, play the game and complete the levels.

Precondition: The game levels haven't been completed before on the user's computer.

Test steps:

1. Run a built version of the unity game.
2. Try to access each of the levels from the level selection screen.
3. Play the first level and complete it.
4. Return to the level selection screen and try to access all the levels.
5. Play through the subsequent levels and complete them.
6. Check to see which levels are accessible each time a level is completed.

Expected Result: Initially only the first level is accessible. After completing a level the subsequent level becomes available to play in addition to the previous levels.

Test Case for US 4-3:

Game Resolution - The game automatically adjusts to fit different screen resolutions.

Assumption: The user knows how to run the game and navigate through the different menus and levels.

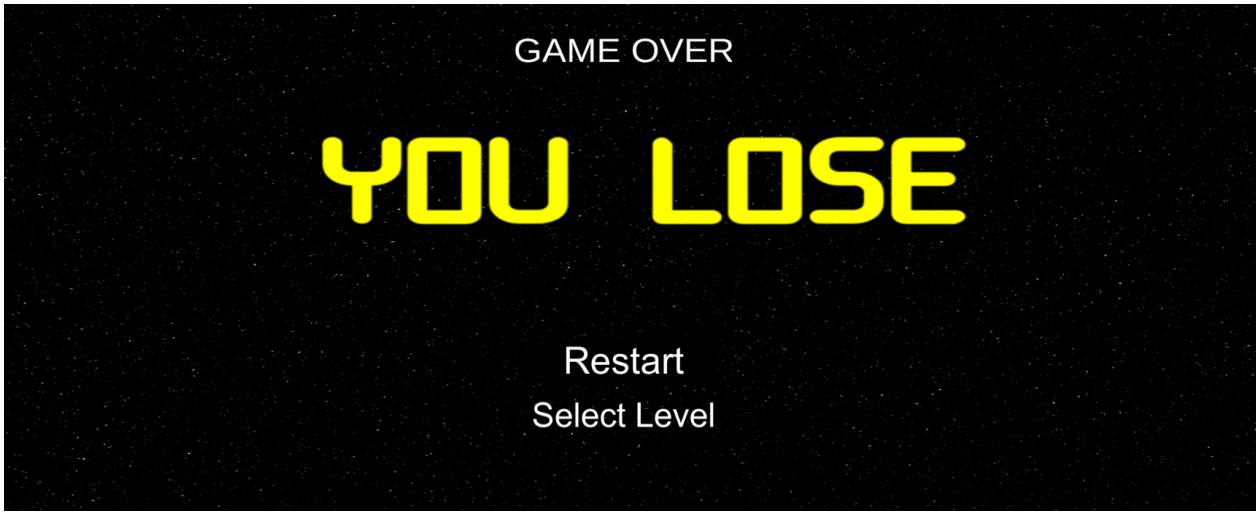
Test steps:

1. Run a built version of the unity game.
2. Play through the game.
3. Observe if the game is correctly displayed.
4. Change the resolution of the display.
5. Run the game again.
6. Play through the game.
7. Observe if the game is correctly displayed.

Expected Result: The game runs and displays correctly on computers running different display resolutions.

User Interface Design Elements

Design Element 4.1: Win/Lose Screens Version 2



Description: Can be seen in Design elements 3.7 and 3.8.

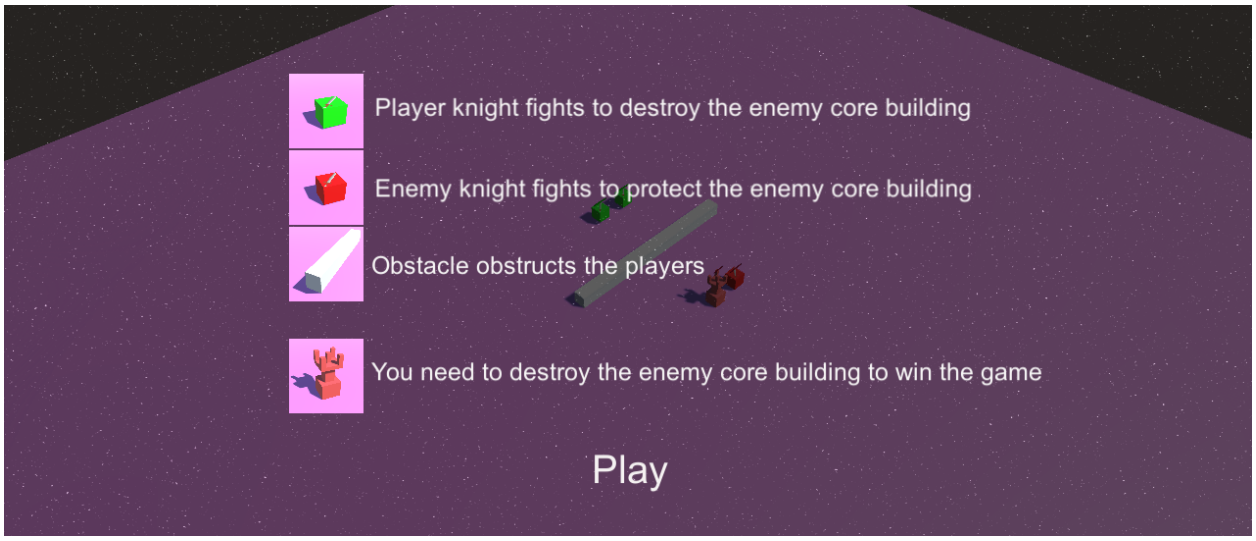
UI: The font has been updated to a more angular style, fitting with the Square/cube theme. The ‘Game over’ text has been removed on the win screen. When the mouse hovers on any of the buttons, it changes the colour of the button to blue.

UX: Though the game has finished, the game over text was removed from the win screen, as this text is often associated with losing. The outcome text is much larger, again to emphase how well the player did. Next Level and Select Level buttons now also ‘blink’. This is to indicate that user action is required. Restart on the win screen was also replaced with Next Level to indicate they can proceed.

Design Element 4.2: Character Intro Scenes

All player introduction screens were developed to fulfill **User Story 4-1.**

DE4.2.1: Level 1 Player Introduction Screen



Level 1 Player Introduction Screen introduces four entities which include player Knight, enemy Knight, obstacle building and enemy core building. The screen has a blinking play button and on mouse hover, changes the colour of the text to blue. On click of the Play button it starts the level.

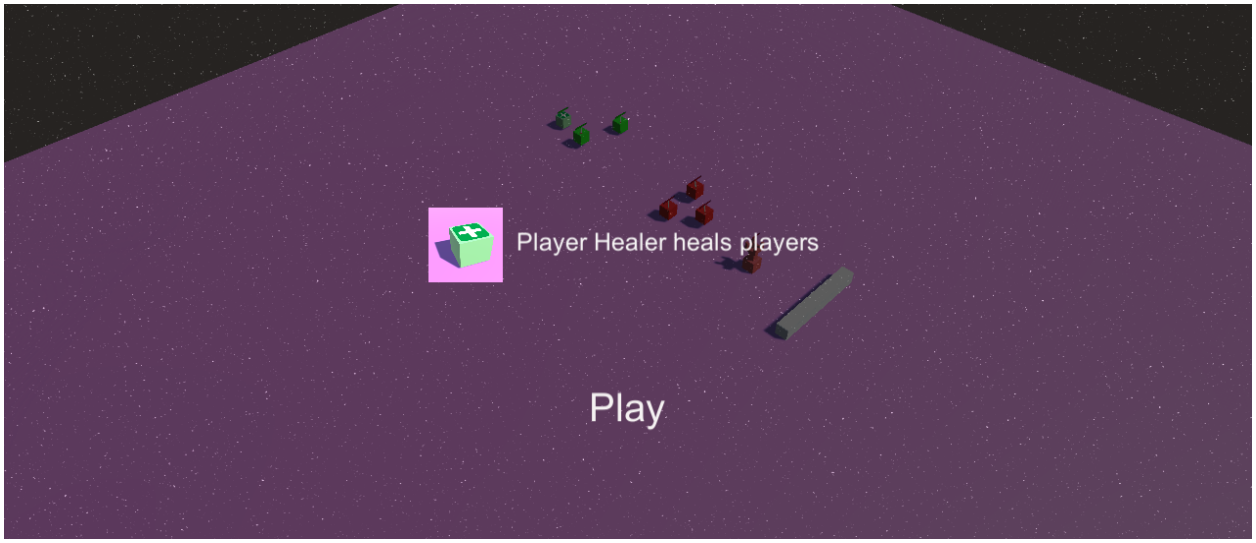
DE4.2.2: Level 2 Player Introduction Screen

Level 2 Player Introduction Screen introduces two new entities which include player Archer and enemy attack tower. Screen has a blinking play button and on mouse hover, changes the colour of the text to blue. On click of the Play button it starts the level.

DE4.2.3: Level 3 Player Introduction Screen

Level 3 Player Introduction Screen introduces two new entities which include player Warrior and enemy basic building. The screen has a blinking play button and on mouse hover, changes the colour of the text to blue. On click of the Play button it starts the level.

DE4.2.4: Level 4 Player Introduction Screen



Level 4 Player Introduction Screen introduces a new entity called player Healer. The screen has a blinking play button and on mouse hover, changes the colour of the text to blue. On click of the Play button it starts the level.

DE4.2.5: Level 5 Player Introduction Screen

Level 5 Player Introduction Screen introduces a new entity called enemy Warrior. The screen has a blinking play button and on mouse hover, changes the colour of the text to blue. On click of the Play button it starts the level.

DE4.2.6: Level 6 Player Introduction Screen

Level 6 Player Introduction Screen does not introduce any new player, just displays the target of the game. The screen has a blinking play button and on mouse hover, changes the colour of the text to blue. On click of the Play button it starts the level.

Sprint 4 Game Progress

The Win and Lose pages were updated to include messages that were clearer to the player. ‘Blink’ functionality was also added to the menu buttons, to better indicate interaction is required to proceed. The game also correctly scales across resolutions and aspect ratios.

Game mechanics progress

As each level is designed to introduce a new character, character introduction screens were added. These show at the beginning of each scene to outline what the new character looks like, what it’s aim is and what the goal of the level is. Both newly added player, and enemy characters are shown on this screen, as well as any new objects not previously encountered.

A level unlock system was also implemented. Now, when a player starts the game for the first time, only level 1 is unlocked. The following levels are greyed out until a player has completed the previous level. Level progress is saved across application closure.

Sprint 4 Retrospective

- The team had another strong week of progress adding some ‘final touches’ to the game. Although communication sometimes faltered, the team was able to stay on track and achieve what was set out.
- What went well:
    - Better clarity in assigning tasks (no work wasted)
    - Better at achieving set tasks within the set timeline
    - Developed “finished” version of the game
    - Adapted efficiently and effectively to customer’s changing requirements
- What to work on:
    - If you have a branch, make sure someone has looked at it and signed off before merging
    - There are too many branches that exist on GitLab that need to be deleted if they’ve been merged (keep locally though)
    - Attendance at meetings
    - Timely response to messages on Teams

Sprint 5 Overview: 12 Dec 2021 - 19 Dec 2021

- Focus for Sprint 5 is to make small, final enhancements to game and to finalize formatting for the process documentation. Additionally, sub-teams were created to develop the user manual and maintenance guide.
- Key team aspect of this sprint is communication. It is crucial that everyone is on the same page to finish the project strong.

Sprint 5 Meeting Minutes

Team Meeting 17: 13 Dec 2021

Minutes: 14:00 - 15:00 via Microsoft Teams

Participants
- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

- Sprint 5 (final sprint) objectives:
    - No more big changes to the game, only small changes/fixes to finalize
    - Ensure any changes are reviewed by all team members
    - Main focus on documentation: user manual, maintenance guide, finalizing sprint document with proper formatting
- Final task assignments for deadline on 20th Dec:
    - CRC cards: Yash Malik
    - Tests: George Griffiths
    - Backlog tasks & User Stories: Ani Udom
    - Requirements lists & exception handling: Ashton Mitchell
    - Final editing and formatting of sprint document: Ashton Mitchell
    - Maintenance guide & installation guide: William Chambers and Vladislav Shishkin and everyone else who contributed to code
    - User manual - written in lay terms for the average person and a brief demo video: Dinil Wickramasinghe to make outline and delegate sections
    - Final game edits (font size): Vinay Kumar Gupta

- Video Demonstration of Game (due 24th Dec) - demo video of the game showing off the features of the product. This shall be a screen capture with commentary uploaded to Panopto: TBD
- Team Questionnaire (due 24th Dec):
    - Scheduled meeting date: Monday Dec 20 - Wednesday Dec 22

Action items:
- Need to determine who will take the lead on the video demonstration
- Create WhatsApp group to improve team communication

Decisions
- Team decided to schedule a final meeting with customer to present final product

Customer Meeting 6: 15 Dec 2021

Minutes: 11:30 – 11:40 @ Chancellor's Building 5.13

Participants
- William Chambers
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik
- Customer

Goals
- Demonstrate final version of game to the customer

Discussion topics
- Customer suggested there should be more information for the user to know the different tactics that they can implement to win the levels.
    - Customer suggested this could be included in the user manual and the team agrees
- Customer was concerned that the colour of the enemies and the enemies' health bars were too similar to easily differentiate
    - Based on this feedback, the team decided to change all of the enemy's health bar colors to red, in order to be consistent with the red vs. green theme and improve the visual distinction between the player and the enemy armies. For example, see screenshot of gameplay from Design Element 5.2.

Action items
- Provide information for tactics in the user guide

Team Meeting 18: 19 Dec 2021

Minutes: 10:00 - 11:00 via Microsoft Teams

Participants

- Ashton Mitchell
- William Chambers
- Ani Udom
- George Griffiths
- Vladislav Shishkin
- Dinil Wickramasinghe
- Vinay Kumar Gupta
- Yash Malik

Goals

- Make sure team is up to date with state of project and submittables
- Assign final tasks

Discussion topics

- Set team deadline before the actual deadline to make sure everything is in order
    - Team deadline: 5pm on Monday Dec 20
- Discussion of final changes to the game:
    - Instruction screen and control screen
    - Access to pause menu when in a level (allows to resume, show controls, or return to main menu)
- Discussion of error with level selection screen and level lock feature:
    - It was discovered that upon completion of 1 level, the level selection screen would occasionally update to incorrectly provide the player with access to the next 2 levels, instead of just the next single level
    - The team quickly returned to B4.1 in order to fix the level lock mechanism
    - The team also decided to simplify the level screen design to show all level numbers from the start instead of hiding unlocked levels behind a question mark (see B5.5)

Sprint 5 Requirements List

Final List Version 5

As each requirement is met, the corresponding test or design element will be listed in parentheses at the end of the requirement. By the end of Sprint 5, the delivered product has met all of the essential requirements as listed below.

Key: ~~Strikethrough~~ indicates removed requirement.

R1.1: ~~Dungeon themed game~~
R1.2: Game that will run on a laptop (see Repository)
R1.3: ~~Game that will run on a phone~~
R1.4: ~~A “bot” that can play the game on it’s own or against other bots~~
R1.5: Ability to understand how well the player is doing in the game (DE1.2)
R1.6: Game that has increasing difficulty through progression (DE3.2)
R1.7: Gameplay modelled after real-time strategy (RTS) (see RTS Style Game Storyboard)
R2.1: ~~Character selection feature as part of gameplay strategy~~
R2.2: Main menu screen with title and “START” button (see DE3.1)
R2.3: Cut scene to establish game backstory (DE2.2)
R2.4: Level selection functionality (DE2.5, DE3.2)
R2.5: 6 levels (DE3.2)
R2.6: Camera zoom functionality (Test Case for US2-3)
R2.7: Enemy characters for player to battle against (DE1.2, DE3.5)
R2.8: Simplistic “cube” character models for cleaner, unique design (DE3.5)
R2.9: Game music throughout to enhance story and theme (Test Case for US3-7)
R3.1: Clear distinction between character types (DE3.5)
R3.2: Tutorial level to familiarise player with gameplay (DE3.4)
R3.3: Win/lose screens to inform player of level outcome (DE3.7, DE3.8)
R4.1: Character introductions/instructions (DE4.2)
R5.1: Instruction screen (DE5.2, DE5.4)

Sprint 5 Backlog

B3.12: Create user manual (US5-1)
B3.13: Create maintenance guide (US5-2)
B3.14: Create Installation guide (US5-3)
B4.1: Level lock mechanism (US4-2)
B5.1: Look into timing of audio
B5.2: Add square wars logo at the end of intro video
B5.3: Add each level title at corner of screen (US5-4)
B5.4: Update level selection screen to have numbers instead of question marks
B5.5: Create a pause menu available in each level (US5-5)

Complete Backlog Tasks

B5.1: Look into timing of audio
B5.2: Add square wars logo at the end of intro video (DE5.6)
B5.3: Add each level title at corner of screen (DE5.1)
B5.4: Update level selection screen to have numbers instead of question marks (DE5.5)
B5.5: Create a pause menu available in each level (DE5.3 & DE5.4)
B3.12: Create user manual
B3.13: Create maintenance guide
B3.14: Create Installation guide
B4.1: Level lock mechanism

Sprint 5 Exception Handling

- In the final run throughs of the game, the team identified errors in the level selection screen (documented in Team Meeting 18). With only 1 day until the project deadline, the team had to weigh up the options for how to fix this. The best solution was to change the level selection screen design (shown in DE5.5) to show all level numbers from the start, instead of requiring question marks to ‘unlock’ into level numbers as the player progressed. Additionally, the mechanism for the level lock feature did not work consistently, so the team worked quickly to fix this.

Sprint 5 Product Documents

User Stories

User Story 5-1: User Manual

AS A	*Acceptance Criterion:*
Player *I WANT TO* Have an instruction manual for the game *SO THAT* I can have a better understanding of the game and have a place to look if I don't know how to do something.	A manual that is easy to read and understand that explains the game's basic functions.

User Story 5-2: Maintenance Guide

AS A	*Acceptance Criterion:*
Developer *I WANT TO* Have a guide that explains the game's code. *SO THAT* If any changes or bugs need to be fixed in the future, I will have an understanding of how the code functions.	A more technical but still easy to understand document that explains the code's functionality.

User Story 5-3: Installation guide

AS A Player or Developer *I WANT TO* Have a guide that explains how to install the game *SO THAT* I can play the game or make changes to the game's code	*Acceptance Criterion:* A description of how to install the game that is clear and easy to understand. *Comments:* This guide can be added into the user manual and maintenance guide so it is specific to the reader.

User Story 5-4: Level title at the corner of level

AS A Player *I WANT TO* Have the level titles in the corner of the level I'm on *SO THAT* I can remember what level I'm on and can see how far I've progressed in the game.	*Acceptance Criterion:* A number in the corner of each level that corresponds to the level the player is on.

User Story 5-5: Pause Menu Available in the Level

AS A Player *I WANT TO* Have access to a pause menu *SO THAT* ● I can stop playing the game at any moment and have the ability to come back. ● View the controls if I need a reminder. ● Leave the game if I no longer wish to play the level	*Acceptance Criterion:* A menu that appears when a key is pressed (ESC) that pauses the game. When resume is pressed the game continues at the point in which it was paused. Controls button shows the user the controls. Main menu takes the user back to the level selection screen. If the main menu button is selected, if a user returns to the same level, the pause menu should not be open. If a user pauses the game, opens the controls menu and then un-pauses the game, the game should return to the previous pause menu and not the controls menu.

Development Tasks

T5.1 for US5-4: Level title at the corner of level

- Create an overlay that shows the level number
- Add this overlay to each level

T5.2 for US5-5: Pause menu on each level

- Add pause button to each level
- When this button is pressed a screen should appear with the following options:
    - Resume
    - Controls
    - Main menu
- The resume button should return the player back to the level and the point they has paused
- The controls button should show a screen that has description of all the game controls
- The main menu button should exit the level and take the player back to the menu screen

Requirements Use Cases

RU5.1 Use Case for US5-5: Pause button
Scope: Game level
Level: Functional level
Context: Each level will have a pause button to let players stop the game at any point
Frequency of occurrence: Any time the player clicks the button
Open Issues: No open issues

Design Use Cases

Use Case 5.1:
UC1-1**: Use Case:** Pause button
UC1-2: **Author:** AU
UC1-3: **Purpose:** There should be a pause button on each level so that a player can pause the game at any point.
UC1-4: **Overview:** The starts when a player enters a level.

If they click the pause button a scene appears with 'paused' written at the top, the game level is blacked/blurred out behind the scene and there are 3 button options. **Alternative 1:** They select the resume button, this takes away the pause scene and the player is able to resume the game at the point they stopped. **Alternative 2:** The player selects the controls button. This shows a new screen that explains all the game's controls to the player. **Alternative 3:** The player selects the main menu button and they are exited from the game level and returned to the main menu screen.

UC1-5: **Cross References:** US2-1, US5-5, RU5.1
UC1-6: **Actors:** Player
UC1-7: **Pre-Conditions:**

UC1-Pre-1: The player must have started playing a level of the game

UC1-8: **Post-Conditions:**

UC1-Post-1: The player is able to **alternative 1:** resume the level, **alternative 2:** read the description of the game controls or **alternative 3:** exit the level to the main menu screen.

Tests

Test Case for US 5-4:

Level Title - Level title at the corner of the screen.

Assumption: The user knows how to navigate around the game and load different levels.

Precondition: The game has multiple levels which are all accessible.

Test steps:

1. Run the unity game.
2. Play each level.
3. Observe the corner of the screen while the levels are playing.

Expected Result: There is a number in the corner of the screen that corresponds to the level which is being played.

Test Case for US 5-5:

Pause Menu - A menu that appears when the esc key is pressed during the game.

Assumption: The user knows how to navigate around the game and load different levels.

Precondition: There is a working game build.

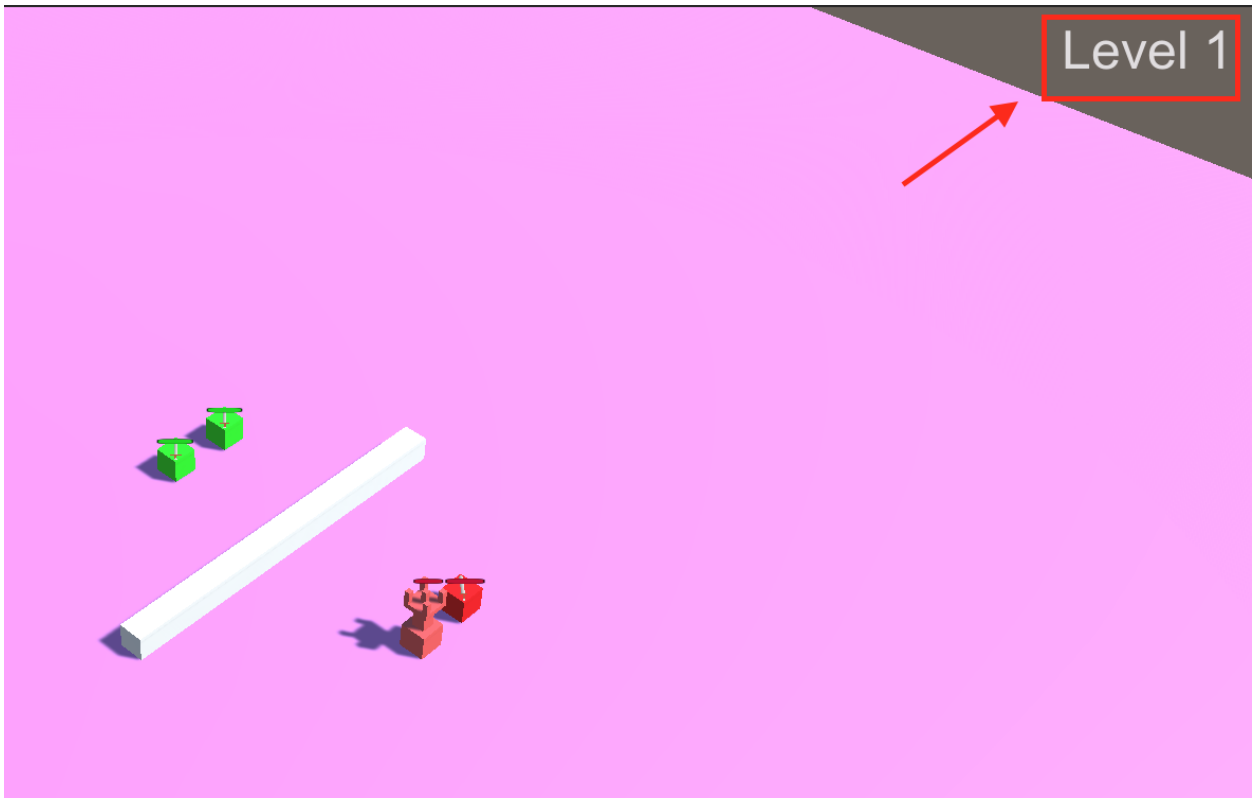
Test steps:

1. Start a game level.
2. Press the esc key on the keyboard.
3. Click on the controls button.
4. Press the esc key.
5. Press the esc key.
6. Click on the main menu button.

Expected Result: Pressing the esc key while playing a game pauses the level and opens a menu. Clicking on the controls button opens up graphics displaying the controls. Clicking on the main menu button returns the user to the game's menu. Clicking either the resume button or pressing the esc key while the pause menu is active closes the pause menu and returns the user to the game.

User Interface Design Elements

Design Element 5.1: Level # Overlay



UI Design: The chosen font is Arial, as it is a simple clear font that is easy to read. Text colour is white to adhere to the simple, but effective colour scheme that runs throughout the game.

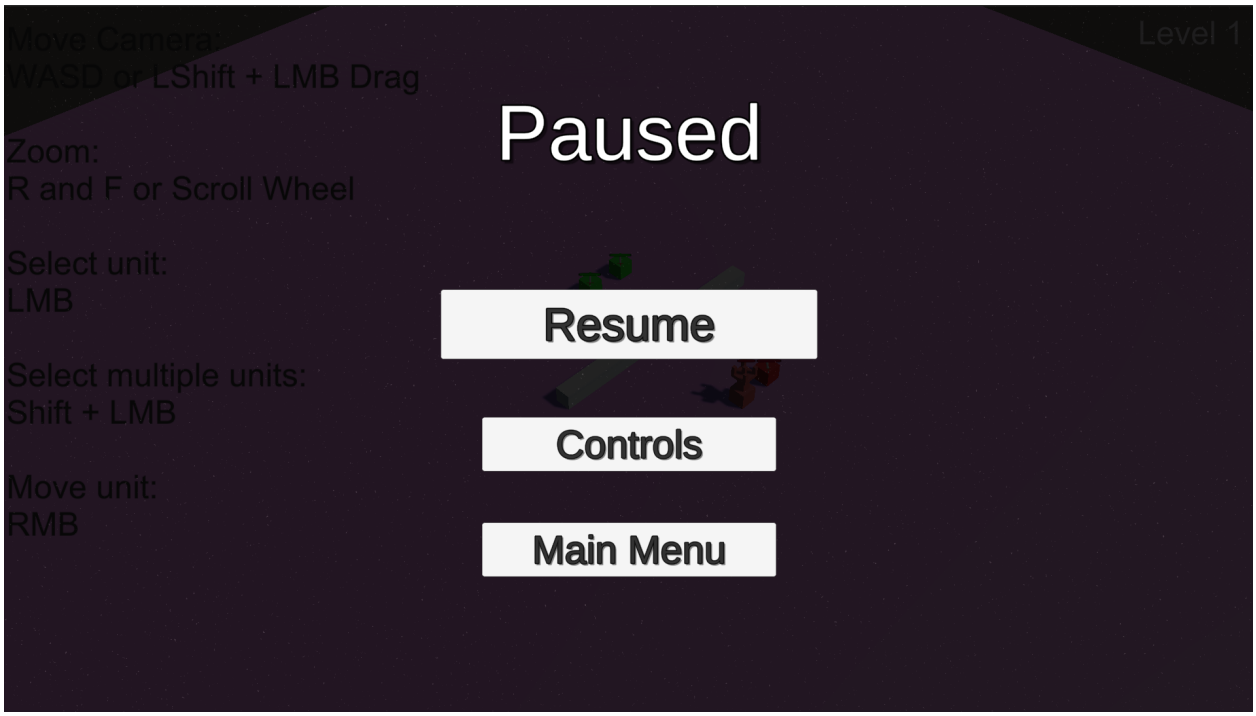
UX Design: Having the number listed on the screen ensures the user knows which level they are currently playing. This acts as an additional indicator of their progress. It is positioned in the top right hand corner, so that it doesn't distract from the main gameplay, but is easily visible.

Design Element 5.2: Instruction Screen

UI Design: The chosen font is Arial, as it is a simple clear font that is easy to read. The main aim of the text is to describe how to control the game, so being clear and concise is a key element. As opposed to the other text elements in the game, the font colour is black. This enables it to be easily read when it is on top of the game background.

UX Design: To ensure new players are aware of the control scheme, the basic controls are listed on the left hand side of the initial tutorial level. This enables any new player to instantly be aware of what inputs are required to interact with, and play the game. Each control mechanism is listed, with the required input to perform the mechanism listed below. To ensure the text does not clash with the character introduction text, the control text only appears once the previous dialogue has been closed. The text is positioned across the left hand side of the display as that is an area that can be quickly looked at to determine controls, but out of the way enough to not distract from the main game.

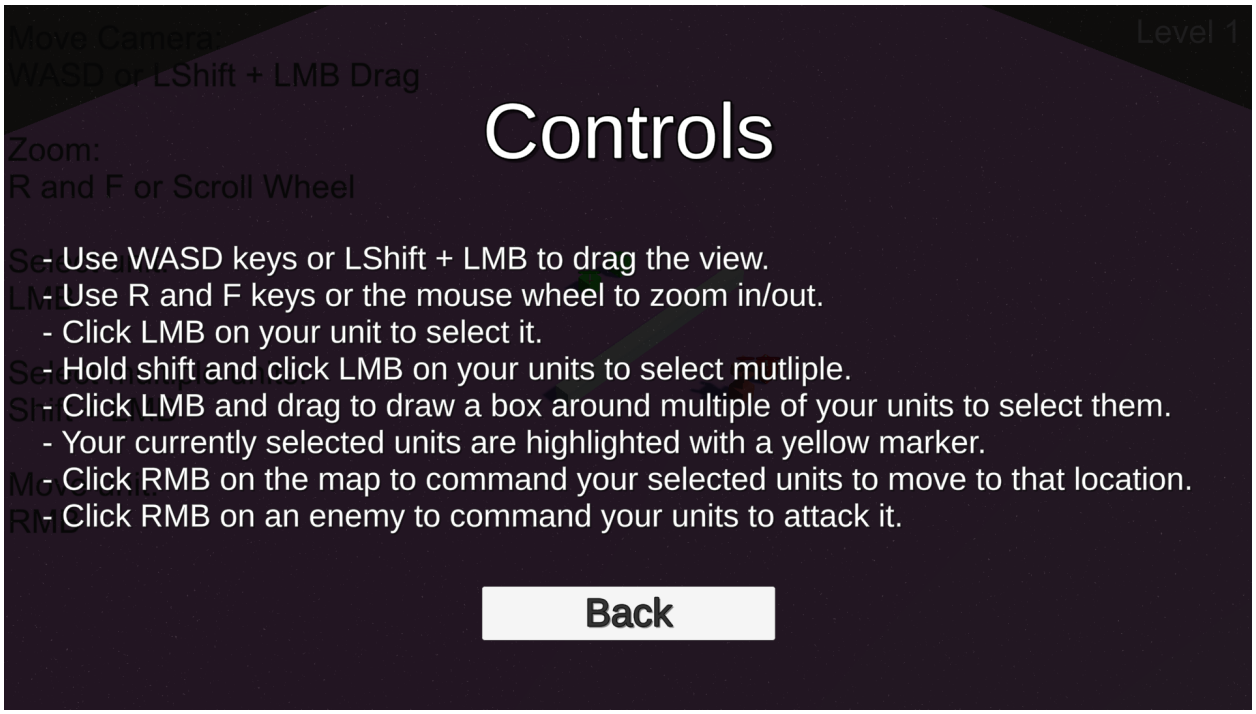
Design Element 5.3: Pause Menu



UI Design: Pause menu continues with the simple design language. White and black are used to adhere to the colour scheme. Again, a simple font is used. The background is a dark translucent black.

UX Design: Menu buttons are located centrally for ease of access. Resume button is the largest of the three menu options to compel the user to continue playing. Controls is listed as the second option to help advise the player of how to play. Main Menu is the last in the list to make it the least attractive option. Music continues playing whilst the pause menu is active to reassure the user that the application is still running. Gameplay does however stop, and resumes once the menu has been closed. To show that the pause menu is on-top of the game, the background is a dark translucent black colour. This helps to enforce that their game position is staying at the point at which they pressed pause.

Design Element 5.4: Controls Menu



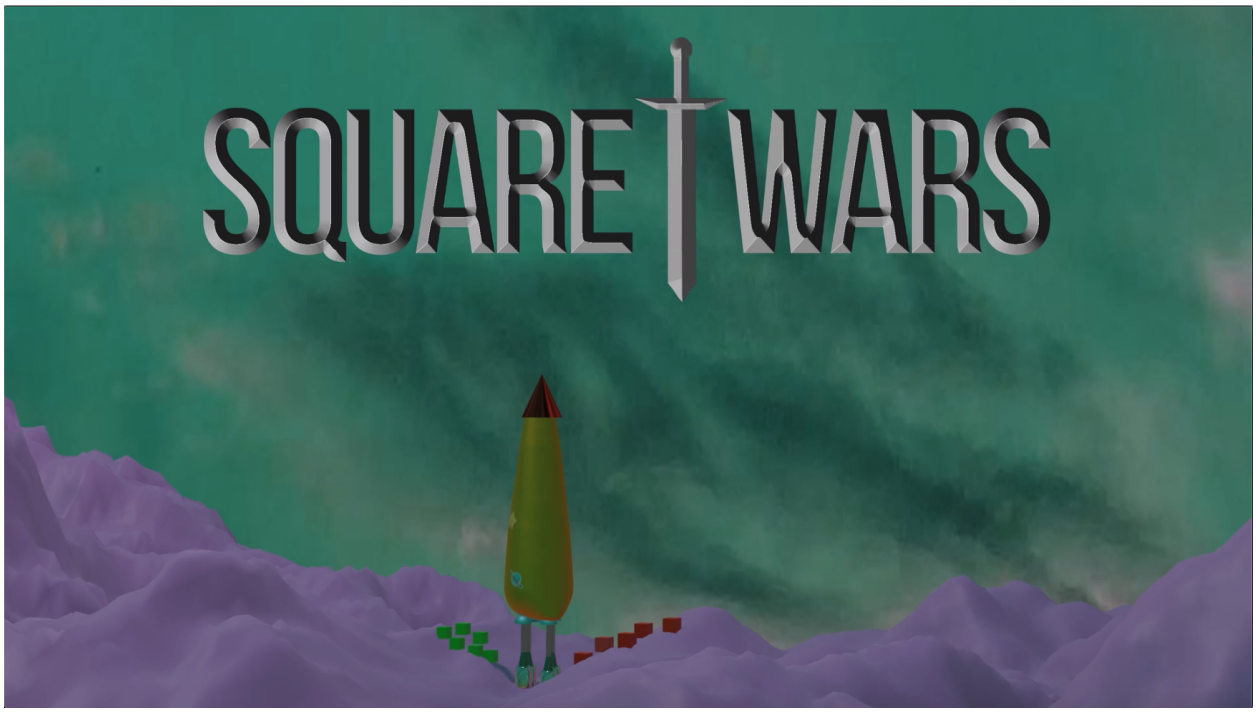
UI Design: Adheres to the design language used within the previous menu. White and black colour scheme. Again, a simple font is used. The background is a dark translucent black.

UX Design: More in-depth explanation of the controls to ensure the user, at all times, has access to the control scheme. Single button placed low and centrally to guide the player back to the previous page.

Design Element 5.5: Level Select Screen Version 3

- Updated in response to exception outlined in Sprint 5 Exception Handling. The level numbers are now listed from the start, instead of hiding unlocked levels behind question marks (see DE3.2 for previous design). This final design is crisp and clean, following the modern design of the rest of the game.

Design Element 5.6: Intro Cutscene Video for Game Backstory Version 3



- The Intro Cutscene Video was updated to include the ‘Square Wars’ logo at the end, ensuring the player can see the connection to the game title in the video. Additionally, the pink ground of the planet as well as the player and enemy cubes can be seen from this screenshot of the intro video.

Sprint 5 Game Progress

Each scene now features it's own music. There are individually composed pieces for the Title scene, MapMenu, and each level. The music has been composed to seamlessly loop. A level indicator has also been added, to ensure the user knows the current level they are playing. Text to show the basic controls is now included on the first tutorial level. This ensures the player is aware of what inputs are required to control the game. A game icon was also added, so the built applications were easily recognisable.

Game mechanics

A pause menu has been added to each level. This now enables the user to pause the game, view controls or return to the main menu. The controls listed on the 'Controls' tab are a more in-depth guide to how to play the game and can be accessed at any time. The MainMenu button allows the user to leave the level if they no longer wish to continue, which prior to this feature, was not available. The ESC key triggers the pause menu.

Sprint 5 Retrospective

- What went well:
    - Communication was stellar, everyone was easily reachable and communicated quickly/effectively using the team WhatsApp chat
    - All remaining backlog tasks were handled (see Complete Backlog Tasks) and the game reached its final, deliverable state
    - Sub-teams were able to create the user manual and maintenance guide in parallel to the team finishing the game product
    - Overall documentation was reformatted into a single process document
    - The team was unified and motivated for the final sprint of the project
- What went wrong:
    - A couple small errors and oversights were identified, which required last minute fixes.
    - Some aspects of the project had to be put on hold as members of the team juggled other university assignments and deadlines.