# Game Production Workbook

## Team Details:

### Team Members and Roles

|  |  |
| --- | --- |
| Team Members | |
| Name | **Role** |
| Emma Walters | Programmer, Designer |

## Your Role:

<Rather than identify your role as ‘Programmer’ or ‘Designer’, try to write one or two sentences describing what you will work on. For example: ‘For this project, I will program the character control scripts and make sure the player character responds correctly, including playing the correct animations’   
1 – 2 sentences.>

As the programmer and designer, it is my job to creating the playing environment, design the levels, Program the player’s character, the enemy/enemies as well as the enemy AI.

## Your Tasks:

<List the tasks you are responsible for. What specific things are you working on for this project? Examples might include finding or editing sound effects, level design, scripting specific level interactions or characters, et cetera.  
Bullet-point tasks list (around 4-5, or as many as you are responsible for)>

Adding environment and scene models and collision boxes to each, programming the player controls, adding collision boxes to the player and enemy/enemies, a navigation mesh for the enemy’s movement ai, programming the shooting scripts for both the player and enemy, movement scrips and a game manager script.

## Team Goals:

<What does your team want to achieve, and by when?  
A list of 2 or 3 goals for each milestone. Consider 2-3 project milestones. Consult your trainer for guidance>

|  |  |  |
| --- | --- | --- |
| Milestone | Date | Goals |
| Level with colliders and player tank movement | 27/2/23 <week 4> | Create Level and add colliders  Create Player Tank and Enemy Tank and add colliders  Add Player Tank movement and controls |
| Enemy tank movement | 1/3/23 <week 4> | Create Enemy Tank Nav Mesh  Enemy Tank movement script  Enemy tanks follow player trigger |
| Set up tank turret movement | 6/3/23 <week 5> | Set up enemy tank turret  Set up player turret using layer mask |
| Camera system and shooting systems | 8/3/23 <week 5> | Both Player and Enemy shooting  Following camera system |
| Game manager and game states | 13/3/23 <week 6> | Game Manager Script and game states operating |
| Working high score system and game GUI | 15/3/23 <week 6> | Save and Load High scores  Game GUI, Create a beefy enemy variant |
| Working Spawn Points add additional enemy tank styles | 22/3/23 <week 7> | Ensure Tanks Spawn at spawn points at start of game and with each reset after death, add a swift enemy tank variant |

## Digital Communication Tools:

<What communication tools are your team using, and why. Write 2-3 or more bullet-points or sentences on why you choose each tool.  
List each tool you are using – at least one, but may be more.>

|  |  |
| --- | --- |
| Digital Communication Tool | Selection Criteria |
| Trello | Industry Standard  Team uses this software  Suitable for Production style |

## Version Control:

<What version control system is your team using? Where is this hosted and what client will you use?  
For example, your team select a git repository hosted on GitHub, with the SourceTree client used to commit your code.  
Write 2-3 or more bullet-points or sentences on why you choose each component (version control system, host, and client).>

|  |  |
| --- | --- |
| Version Control System | Selection Criteria |
| GitHub | Most widely adopted version control system for software development  Collaborate with other people |
| Source Tree | Local client that can then update changes onto the server GitHub Server  Easy to manage changes |

## Integrated Development Environment:

<Visual Studio is not the only IDE that works with Unity.  
Identify at least one other alternative and select which IDE you will use for this project (note, each team member may select a different IDE, so describe *your* selection criteria.)  
Write 2-3 or more bullet-points or sentences on why you choose this IDE.>

|  |  |
| --- | --- |
| Integrated Development Environment | Selection Criteria |
| Visual Studio | Industry Standard  Team uses this software  Suitable for Production style  Supports a number of languages |
| Alternate IDE(s) | |
| Xcode, QtCreator, Eclipse, Code::Blocks, Sublime Text, Atom, Vim, Emacs | |

## Team Planning Discussions:

<Keep a record of when your team met to discuss the project, and what you contributed to the discussion. Give details of at least 3 meetings. You can attach screenshots of any emails or chat logs that show team planning discussions.  
Bullet-point list or 1-2 sentences for each team meeting. Provide enough evidence to demonstrate your active participation in team planning discussions.>

|  |  |
| --- | --- |
| Team Meeting Date | Description of Contribution |
| <Example> | <In today’s meeting I discussed the trouble I was having implementing the enemy AI algorithms and suggested we cut back the scope of the project. Team agreed to have simpler enemies.> |
| 20/3/23 wk 7 | For the tanks game I want to add levels, with increasing number of enemies and increasingly difficult enemies, maybe some powerups maybe an alternate game mode that is player vs player or someway to change the player’s tank or level up the players tank |

# Post-Project Analysis

<Complete this section towards the end of your project>

## Project Feedback:

<List 2-3 pieces of feedback you received, and any action you took in response. This could be bug reports from playtesting, or feedback from your trainer or peers>

|  |  |
| --- | --- |
| Feedback | Actions Taken |
| Need more variety in enemies | Created 2 different variants of the enemy, one that is quicker and one that is slower with more health |
| Beefy Enemy has way too much health | Reduced starting health from 500 to 300 |

## Game Engine Analysis:

<Write 200-300 words on “What makes Unity a suitable game engine for professional game development?”. In your analysis, identify at least one other modern game engine, comparing it to Unity and identifying which may be more suitable for use in a game studio>

|  |
| --- |
| Unity – Game Engine Analysis Unity is a suitable game engine for professional game development because of its accessibility, easy access to online resources with tutorials for beginners and advanced programmers often meaning employees will have previously learnt to use unity and may not need any further training as they would almost always need if employed to a company with a unique game development engine as newly employed programmers likely would have never used before.  The cost of having a Game Engine Designed from scratch is far more expensive than simply purchasing a yearly subscription to unity, hence making it the more appealing option for many companies. Unity is easy to use to create games supported on a large variety of programs and features tools for both creating 2D and 3D games. It is also supported in making VR and AR games and is highly user friendly when making mobile games.  Unity is popular among indie video game developers as well with low revenue or funding either being free to use or $40 a month or for high revenue customers earning $200k or more annually for $185 a month to gain access to industry and commercial standard tools on a widely used and highly popular game development engine with a large variety of easy to access learning resources and content. Unreal Engine 5 – Game Engine Analysis Similarly to Unity, Unreal Engine supports a variety of platforms and has features to support both 2D and 3D game design. It is widely used and has a variety of easy to access tutorials available online. Also, alike Unity, Unreal Engine is free to use until your title earns over $1 million USD at which point that charge a 5%, and they also have premium options that provide extra support for large companies including premium supports and options for private training.  Generally, Unreal Engine is considered to have better simulation graphics and better lighting simulation than Unity however is not as user friendly when games and may require further experience to be proficient. |

## Technical Issues and Problems:

<List at least one issue or problem ***you*** encountered during the project, and what you did to resolve it. This could be a problem with the game, or with your development blog.  
For example, you might have had trouble programming a specific feature and decided to redesign it with your team, or ask your trainer for advice on how to implement it.  
Write 1-2 sentences for both the problem description and resolution.>

|  |  |
| --- | --- |
| Technical Issue or Problem | Resolution |
| When the player dies, they respawn the next game exactly where they were before | Create an empty game object and have the tank spawn there every reset. Was also added for enemies |
| Enemies would not reset to their spawn points at the start of New Game. | Ensure that all tanks would be disabled at the end of a game, and the enabled again at the start of a new game, then applying the reset spawn function on enable meant all the tanks returned to their spawn points at the start of a new game. |
| New game button would not work from the high scores GUI | Removed the Image that filled the background of the high scores panel |
| Enemy variants would not return to their patrol path or would not leave patrol path | Believe this was cause by having the range on the trigger collider the same as the distance the tank would stop and shoot from, ensuring the stopping distance was less than the range of the collision trigger. |
| Health Display would display enemy health when an enemy was hit | Creating a separate script for the health of the player and the health of the enemies so that only the script for the player health refers to the heath UI display. |
| Player would not take damage from enemy shells | Edit the Shell script to identify if the collider it intercepted was the player and apply the TakeDamage function from the Player Health Script, rather than the Enemy Health Script which the player did not have. |

## Copyright Details:

<List any assets you used that you did not create yourself, and details of where you found them or their copyright information.>

|  |  |
| --- | --- |
| Asset Filename or Description | URL or Copyright Information |
| Tank Game Models | <https://aie.instructure.com/courses/1030/files/697722?wrap=1> |
| Explosion Particle Systems | <https://aie.instructure.com/courses/1030/files/698296?wrap=1> |

## Industry Technology Selection:

<Throughout this subject you used several tools used by game students for professional game development.   
In future subjects in this course you will have more opportunities to create games. List the tools you or your team used for this project, indicate whether you will use this tool in future subjects, and why.>

|  |  |  |
| --- | --- | --- |
| Tool | I will use this tool again in future subjects (Y/N) | Reason |
| Trello | Yes | The opportunity for cross collaboration and ease of informing teammates of what is being done and what need to be done makes this a highly effective way to communicate progress to teammates on an easy-to-use platform. |
| SourceTree | Yes | Easy to use, industry standard, makes it easier to use GitHub without having to reupload the entire project each time a change is made and easy to pull changes made by collaborators too. |
| GitHub | Yes | Makes it easy to collaborate on a project without accidentally corrupting files or creating overlapping changes. |