Capstone Active Shooter

Bradley Grapes
Melissa Maillot
Isaac Achten
Jordan Russell

Week 8 Report

Semester 2 2022

Contents

Introduction

The Capstone Active Shooter project is, at its core, designed to assist staff and volunteers in high-risk locations to train and ideally intrinsically understand the sequence of events in an active shooter environment: Run -> Hide -> Fight.
The intention of this project is to build a **workflow** and **basis** with which a 3D-scanned location can be inserted into a Unity or Unreal Engine project, set up with minimal excess work and then used in a virtual reality setup.

The project sponsor is Professor Regan Potangaroa of the School of the Built Environment, Massey University. The supervisor is Athar Imtiaz, lecturer of Computer Science at Massey University. Professor Potangaroa has been involved in meetings with the development team and has been included in discussions, providing input and feedback throughout.

We have made significant operational progress within the previous four weeks, and we intend to polish some existing features and implement some additional features prior to our upcoming 12 week report/presentation.

Process

Project Plan

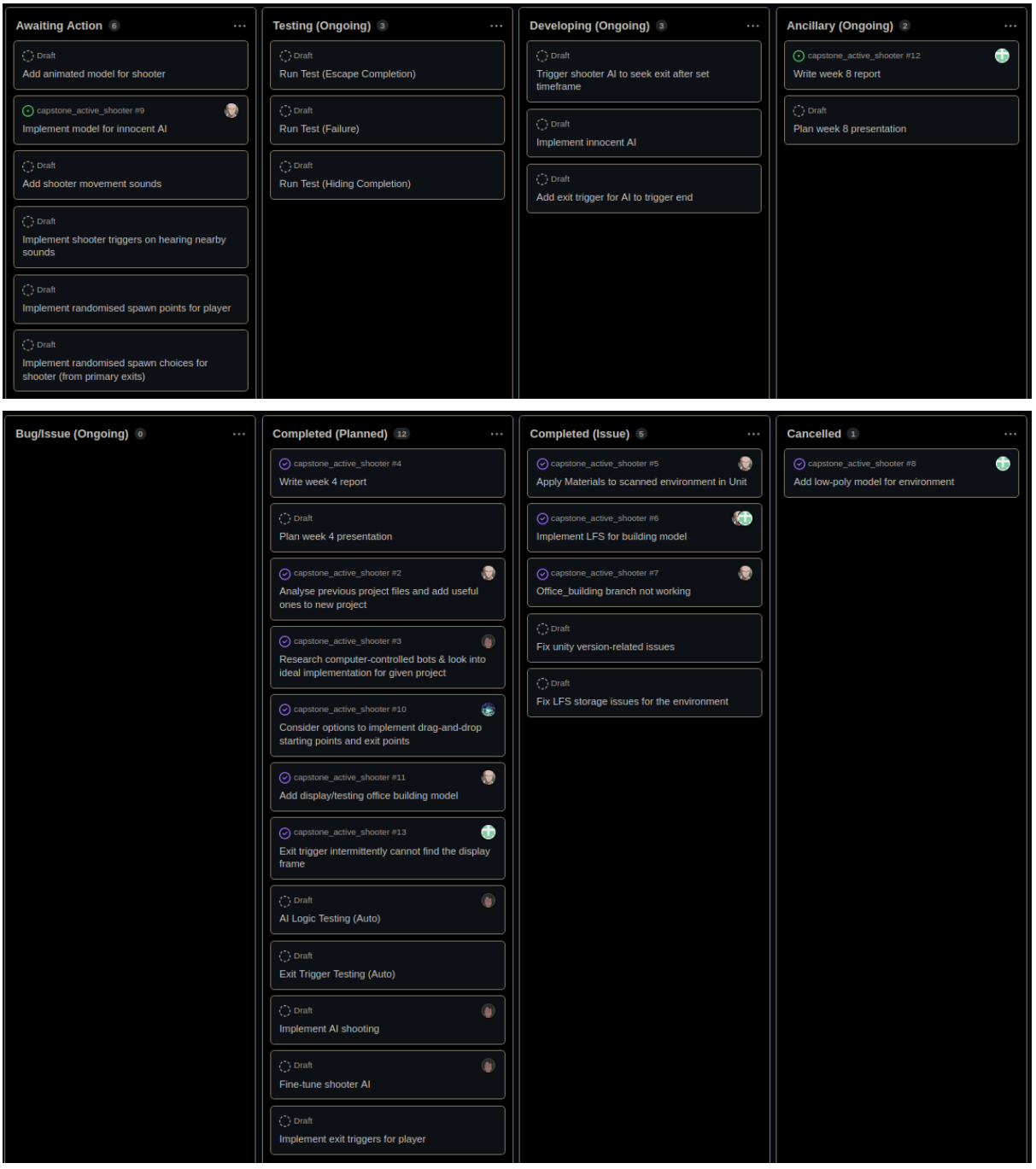
As one might expect, the project plan has developed as issues and information from the sponsor have arisen; unfortunately due to time constraints and complexities in the prioritised tasks, some future features will likely not be achievable within the timeframe of this capstone project. The sponsor has been made aware of limitations and is understanding of them, as well as supportive of our priority of building a basis upon which a more refined and immersive active shooter environment can be completed. Some examples and reasoning of targets which have been discussed or planned and have either been cancelled or moderated are:

- Window escapes: The sponsor wanted the possibility of an asset being picked up and thrown through a window to open it as an escape. However, due to the nature of laser-scanned environments, windows and assets are not modelled individually and this would require incredibly complex automation, or manual model manipulation. The possibility to place a secondary exit at windows is still included.
- Virtual Reality: While this is a reasonably simple implementation for Unity, none of the development team are in possession of a compatible device, and the sponsor has indicated that this is important for a final implementation but not within the timeframe we are working on.

We have additionally added or moved some targets, some of which are upcoming and some which have been met since week 4:

- Improved Environment: As a display environment - i.e. something that can be used to demonstrate the functionality of the product - we have considered and implemented a reasonably lightweight but versatile environment which has provided an improved visual experience for presentations, has allowed us to test our AI with a different environment and additionally improves our AI testing capabilities (i.e. how it handles stairs, etc).
- Innocent AI: While the sponsor wanted this from the start, his emphasis on immersion has indicated that we should place a priority on some basic immersion features. Innocent AI is a feature that we can at least get a basic implementation of.

The current status of the project can be visualised with the following screenshots from our development tracking software (Github Projects). Notably, we have chosen to move to a dynamic development environment where our plans can actively be adjusted fit our capabilities and priorities, and to fit around others' efforts.



Resource Allocation

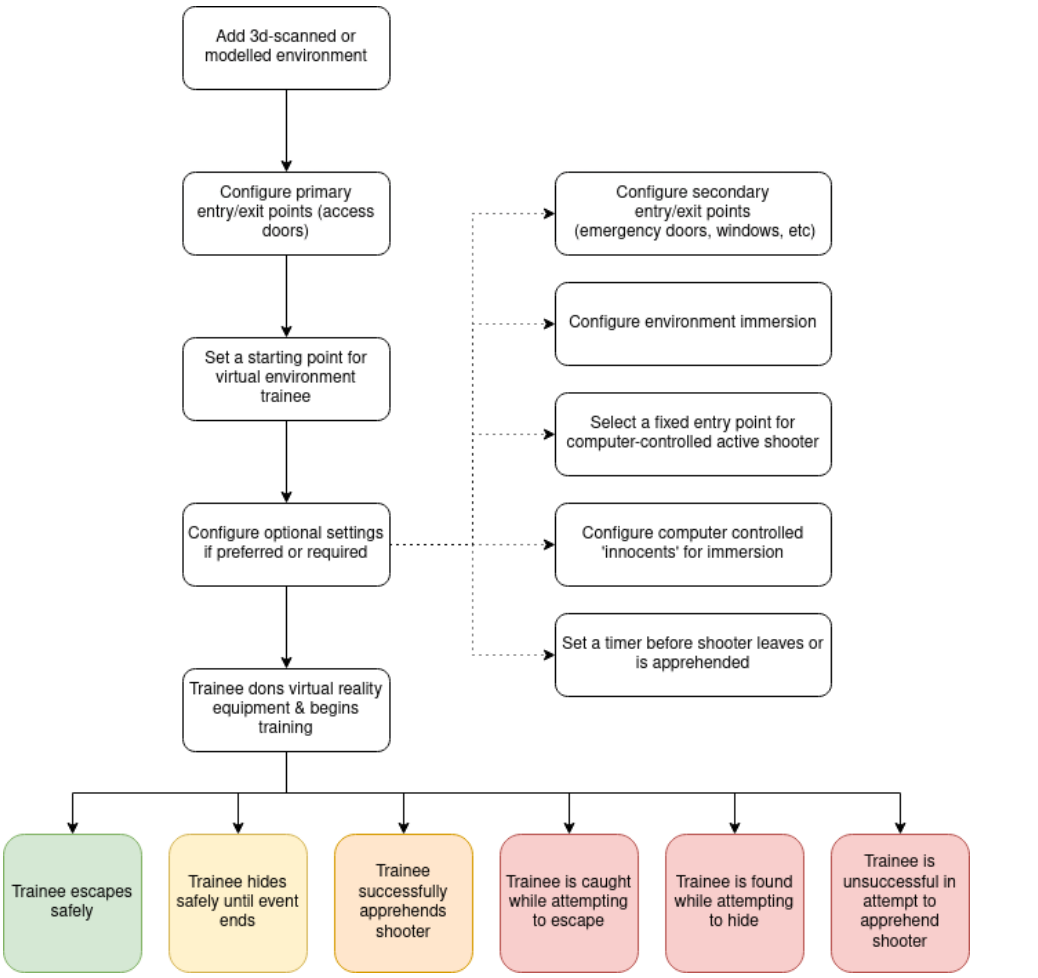
Certain members of the group have been allocated particular tasks due to proximity with a central goal, or capabilities or limitations. For example: Isaac has been allocated all tasks regarding AI, with a standing offer from Jordan for support where needed. This allows him to focus on the details of what is a very important factor in the improvement of this project and improve his own understanding of unity's navigation & AI features without distractions. We are utilising Bradley's advantages of better internet speeds to work through the significant amount of asset data made available to us from the sponsor, where network speed

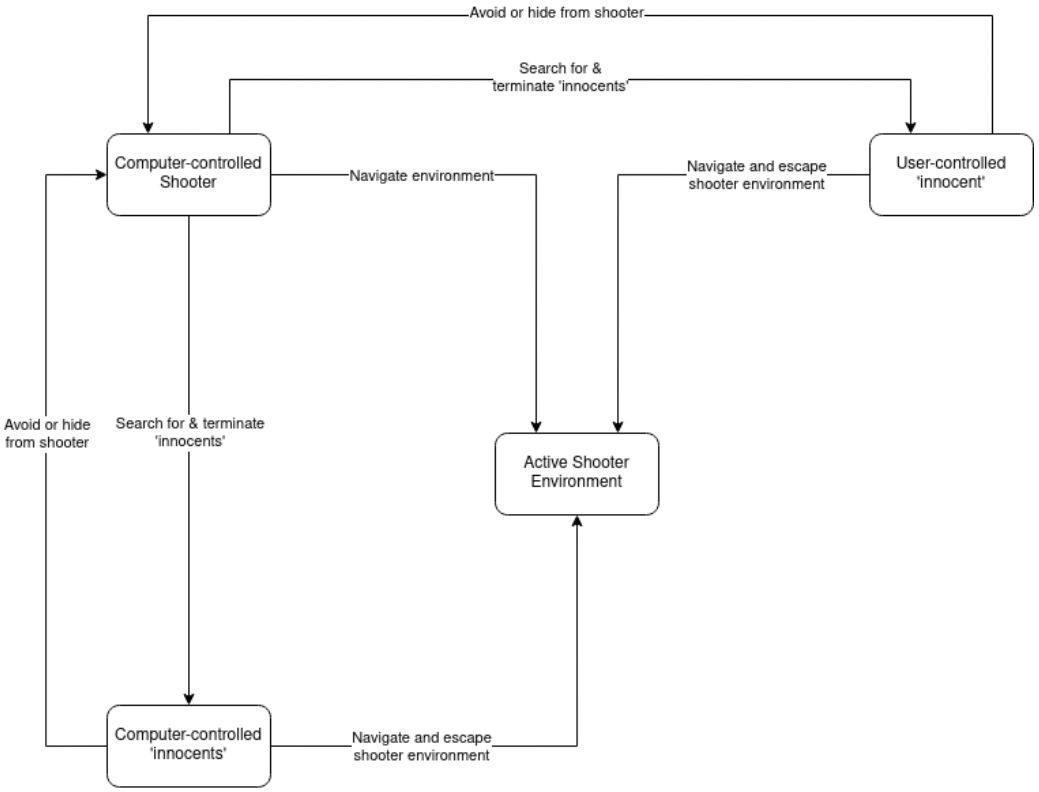
limitations would restrict Jordan or Melissa's ability to complete particular tasks in a reasonable timeframe.

Architecture/Design

The project's overarching design and architecture has not changed since the week 4 report. The diagrams from that implementation are included and are as follows: Diagram 1 is an implementation workflow which indicates the required and optional steps that a **fully completed** implementation would require, plus potential outcomes (coloured in relation to 'outcome preference'). The second diagram is an operational workflow - how the modules in the environment should operate in relation to each other. These workflows do not indicate the expected state of this project at the end of the semester; the full scope and sponsor's final goals are significantly outside of the capabilities of a 4-person team within a semester. The sponsor is understanding of this and expects that this will be a basis for continued development. However, we expect that we will be able to complete the requisite goals for the workflow prior to the end of the semester.

The project is built upon Unity's empty 3D template, and the only non-standard assets used as yet are models from the previous Unreal Engine project supplied by the sponsor, and the selected office building presentation model, which Bradley has sourced from the Unity Store. We are avoiding using proprietary assets wherever possible.


Add 3d-scanned or modelled environment
Configure primary entry/exit points (access doors)
Set a starting point for virtual environment trainee
Configure optional settings if preferred or required
Trainee dons virtual reality equipment & begins training
Configure secondary entry/exit points (emergency doors, windows, etc)
Configure environment immersion
Select a fixed entry point for computer-controlled active shooter
Configure computer controlled 'innocents' for immersion
Set a timer before shooter leaves or is apprehended
Trainee escapes safely
Trainee hides safely until event ends
Trainee successfully apprehends shooter
Trainee is caught while attempting to escape
Trainee is found while attempting to hide
Trainee is unsuccessful in attempt to apprehend shooter


Avoid or hide from shooter
Search for & terminate 'innocents'
Computer-controlled Shooter
Navigate environment
Navigate and escape shooter environment
User-controlled 'innocent'
Avoid or hide from shooter
Search for & terminate 'innocents'
Active Shooter Environment
Computer-controlled 'innocents'
Navigate and escape shooter environment

Usage of Project Infrastructure

As outlined in the week 4 report, we are using Github for version control and Github Projects as our project management software. These are interlinked, and modules within the project can be raised as issues in the github repository; however we are avoiding doing this with general development where possible and instead are using the issues system as intended, for actual issues.

Issues

At the current time, we have 2 open and 11 closed issues, 10 of which were completed/committed and 1 of which was cancelled. Of all the issues:

- 1 open and 3 closed (1 cancelled) are assigned to Jordan
- 2 open and 4 closed are assigned to Bradley
- 1 closed is assigned to Isaac
- 1 closed is assigned to Melissa
- 1 closed is assigned to both Jordan & Bradley

As both Isaac and Melissa have been tasked with broad tasks (Shooter AI & Trigger Assets, respectively), they have not been addressing issues with the overall project. Bradley and Jordan have been dealing with smaller tasks on a more frequent basis, allowing them to address issues as they arise.

Repository

Over the life of the repository to the point of writing this report, there have been a total of 7 branches (on top of the master branch). Three of these remain open at the time of writing. These have primarily been dedicated to larger tasks such as AI development and trigger asset creation. At the time of writing, the master branch is up-to-date with all external branches.

There have been 45 commits, 6 of which are merges of branches. All others have been commented (though one does not have a descriptive comment - “oops minor insignificant change”). Of these commits:

- 16 were by Jordan
- 5 were by Melissa
- 17 were by Isaac
- 7 were by Bradley

It’s fair to note that Bradley’s commits have been largely finished changes, while Isaac and Jordan have been committing with more iterative and minor changes; hence the difference in commit count. The commit timeline has been fairly consistent since the implementation of the project began at the start of August. There was, however, a period of downtime of about a week during late August. The slightly late start and the downtime were contributed to by some issues with our team’s time availability and scheduling issues - Melissa is overseas and is not available for evening meetings, while Jordan and Bradley work full time so are

limited to evenings or weekends. Further, in the past week we have encountered some issues caused by Github Large File Storage and Unity versions, so there has been a reasonably significant concentration of commits within that time period to fix these issues.

Quality Assurance

Manual Testing

The active shooter workflow & implementation testing cannot be entirely automated, and as such this project will rely largely on manual testing to ensure that the workflow is effective and the product produced when that workflow is followed is acceptable in its implementation. Items that will require testing for an initial implementation:

- Workflow can successfully be followed to configure a given shooter environment
    - Test with multiple environments, both scanned and modeled
- Starting points and exits work properly
    - Test in dry runs without shooter, or without shooter damage enabled in multiple environments
- Shooter can navigate environment
    - Test in dry runs without a user, allow shooter to navigate area while being monitored a number of times in different environments

Of these, play-through (user tests) have been completed for exit triggers, user controls, and shooter AI navigation. While the workflow has been outlined, we have not finalised it and this is not being tested for consistency until the basis is complete.

Automated Testing

Despite the limitations of automated testing in the context of this project, we are able to implement some white-box testing in order to ensure the sound function of certain trigger assets and the AI navigation functionality. Automated testing operates using Unity Testing Framework (UTF) which allows for tests to trigger in Unity's Edit Mode and Play Mode.

Test triggers and results are recorded in the Unity Test pane, and can be logged for ease of use. Below is an example of a triggered test in UTF.

At the time of writing, we currently have tests for the following implemented:

- Exit zone action upon player trigger
- Exit zone action upon shooter trigger
- Shooter logic when no player is detected (Search mode)
- Shooter logic when player has been seen (Chase mode)
- Shooter logic when player is inside shooting range (Engage mode)

We plan to expand unit testing to player triggers (end when shot), innocent AI logic (once implemented) and spawning logic (once implemented).

Product

This project does not specify that any specific executables are created. Instead, the deliverable is a modular Unity project that can be adapted to a given environment.

At the time of writing, we have an operational basis for the project, including a player prefab asset, a shooter prefab asset (capable of patrolling, chasing and shooting) and trigger assets. We've also implemented some 'sample' options for an environment and a result screen (for when the player succeeds or fails). With these, we have the ability to begin testing our workflow: create a new scene; add environment, insert/adjust triggers, insert shooter, insert player and insert end screen.

As this is a Unity project (and the sponsor has suggested that there is no necessity for it to be compiled into an executable), it will remain as a unity project and while we may consider the ability to compile the game as a stretch goal it is not a primary concern at this stage.

The presentation will show videos of the current state of workflow and demonstrations of the shooter AI and player triggers.