

Capstone Active Shooter

Bradley Grapes

Melissa Maillot

Isaac Achten

Jordan Russell

Week 4 Report

Semester 2 2022

Contents

Introduction	2
Process	3
Requirements	3
Infrastructure	3
Life Cycle Model & Project Plan	4
Technology	4
Architecture/Design	4
Risk Management	6
Quality Assurance	7
Testing	7
Issue Tracking	7
Product	8
References	9
External References	9
Internal References	9

Introduction

The Capstone Active Shooter project is, at its core, designed to assist staff and volunteers in high-risk locations 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.

Process

Requirements

The primary requirement of this project is to design and implement a workflow for configuring a training environment for active shooter situations. This must be built to use Unreal Engine or Unity for their integrated VR capabilities. It has been stressed by the sponsor that a major issue with implementations of this project surrounds immersion, or a lack thereof, and as such immersion itself is a priority for research - users too easily feel as if they're in a game.

Based on the primary requirement, we have built the following priority requirement list:

- Build upon the Unity Game Engine (group choice)
- Add provisions for a trainer to configure a training site:
 - Allow a trainer to add their site laser-scanned model
 - Allow a trainer to easily add a starting point for the trainee
 - Allow a trainer to easily add one or more exits for the trainee
 - Allow a trainer to choose an entrance for the shooter to start at
- Implement a computer-controlled shooter 'bot' which can navigate the environment

Further, we have a secondary requirement list to try to reach the goal of improved immersion. These are not necessary for a working product or to complete the workflow, but are ideal for a finished product:

- Implement sound and sight sensitivity for the shooter, so that it will react to nearby movement
- Implement logical movement for the shooter so that it navigates methodically
- Implement 'innocent' computer-controlled bots that will attempt to escape or hide:
 - Allow a trainer to choose if and how many innocent bots will be included in the simulation.
- Environment fidelity improvements
- Allow for the export of a configured project into an executable for ease of use

Infrastructure

The team has agreed to use a private Github repository for code management and issue tracking, and Github Projects as a project management tool due to the close integration between the two and the general team familiarity with Github as a tool.

As the project is to be contained and operated within the Unity ecosystem, there are no deployment requirements initially beyond the project's capability to be run on any system with a current or LTS version of Unity. There is, at the current time, no requirement for automation of any process.

Life Cycle Model & Project Plan

The team is and will continue to loosely use a scrum project management methodology. The majority of this project is implementation-based and there are clear goals to reach, which given a basis to work off, sub-teams or individuals can progress upon their own sections without requiring input or concurrent work from others.

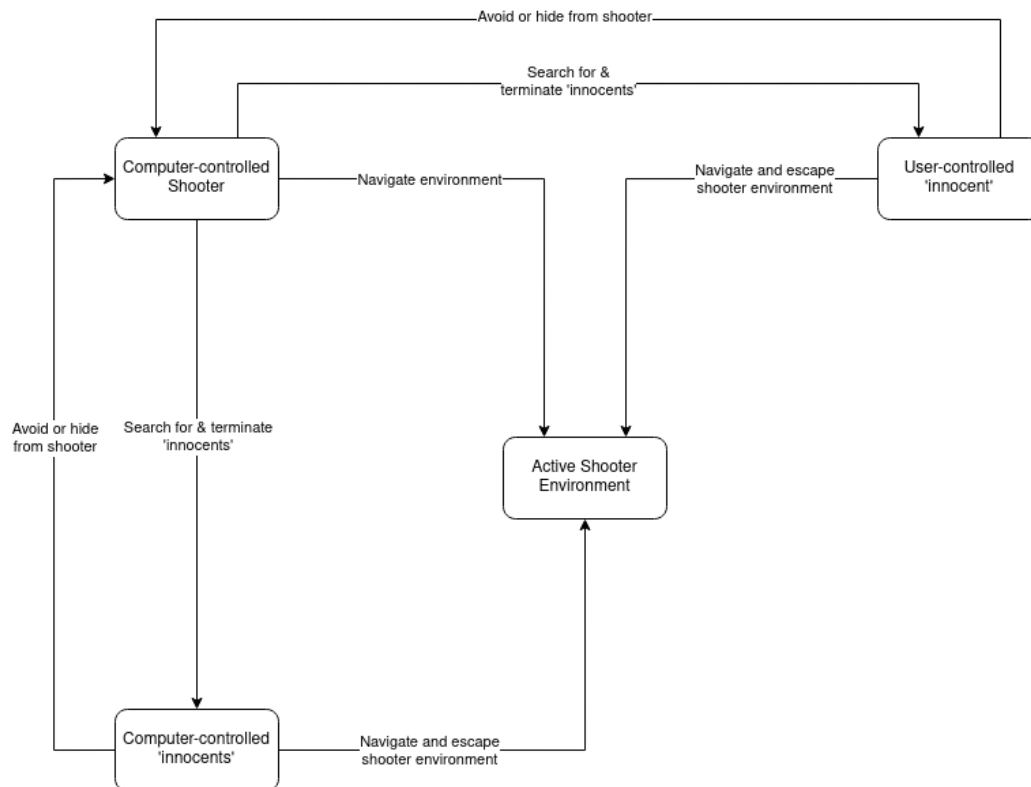
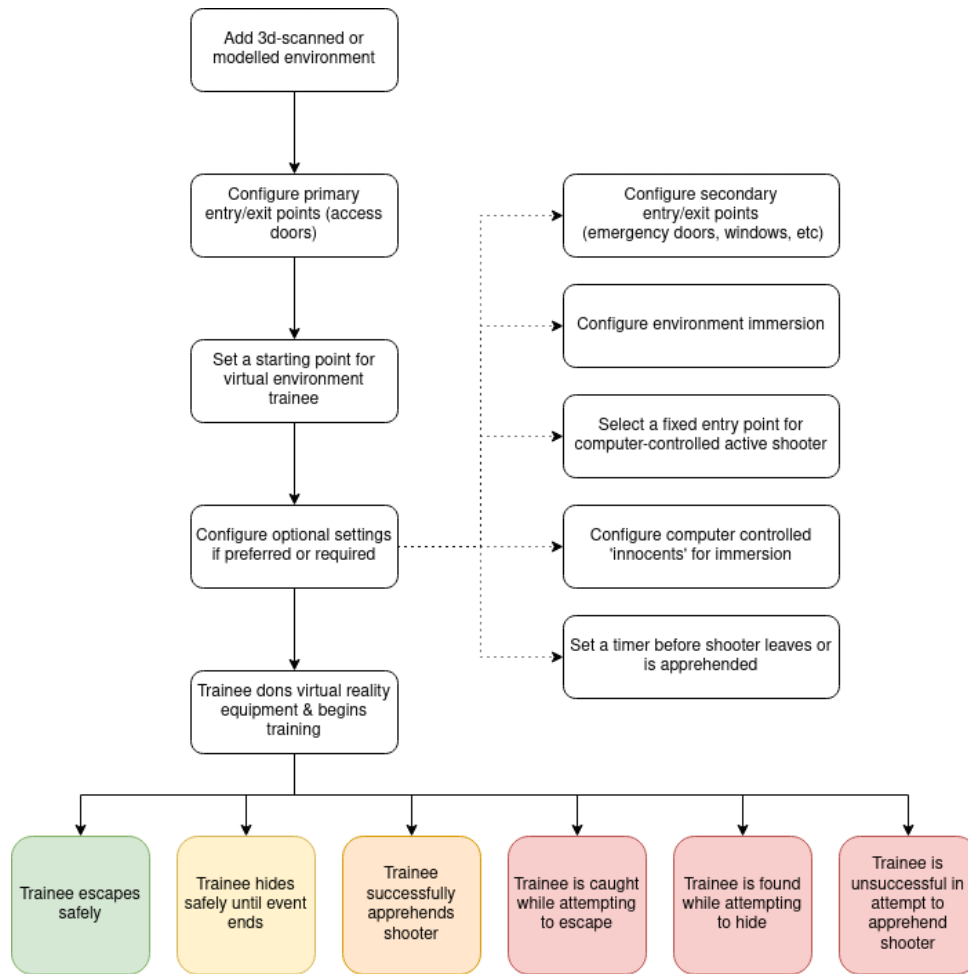
We will ideally keep this flexible, however, as priorities from the project sponsor are expected to change as the project continues and develops.

Technology

This project will be created entirely within the Unity 3D game engine, with no external technologies required. Some free assets from the Unity Store may be utilised for ease and speed of implementation. Unity 3D uses C# as a programming language but includes a visual editor and does not require coded implementations for all modules.

Architecture/Design

The project is primarily aimed at producing and implementing a workflow for active shooter simulations. However, it proved pertinent to generate two workflow diagrams - the first is an implementation workflow which indicates the required and optional steps, 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.



Risk Management

This project is relatively low risk, with no major concerns being raised. The potential risks would be implementation time for secondary requirements (which are not necessary for the 'base stage' completion of this project) and potential complexity of computer-controlled users. Neither of these should be a concern for an initial implementation.

Quality Assurance

Testing

The active shooter workflow & implementation can not reasonably be 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
- Virtual reality works as expected
 - Test using multiple VR setups if possible

Issue Tracking

Issues will be created and tracked via the Github project's Issues tab and can follow the standard flow process for an issue:

1. Issue is raised
2. Issue is discussed and allocated
3. Allocated team member(s) assess issue for correction
4. Allocated team member(s) correct issue
5. Allocated team member(s) close issue and push fix into relevant branch

Features can be raised and allocated through the Github Project and will follow the issue workflow from there (Github Projects directly links into Github Issues).

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.

As such, the current state of our product deliverable is the following:

- Project infrastructure (Jordan)
- A basic 3D Unity project (Jordan)
- Relevant assets from previous implementation (Bradley)
- Navigational mesh candidate for user and AI navigation (Bradley)
- Assets as a basis for first-person virtual reality controls (Isaac)
- Assets as a basis for a computer-controlled active shooter (Isaac)
- Script drag-and-drop assets for starting and finishing points (Melissa)

This will, of course, expand as the project progresses.

References

External References

- Unity Game Engine - <https://unity.com/>

Internal References

This project will heavily refer to content delivered to us directly by Regan as the final product of a previous implementation. This product was created in Unreal Engine but did not implement a full workflow as this one is intended to and as such the primary intention is to use that project for its assets.