VR: Urban Movement Design Document

Intern: Lim Wan Ning
Date: 30 July 2021

1. Versioning

Unity Version: 2020.3.11f1

2. Dependencies

Packages	Version
SteamVR	2.7.3
Animation Rigging	1.0.3
TextMeshPro	3.0.6
OpenVR XR Plugin	1.1.4
XR Plugin Management	4.0.1

3. Scenes

No.	Scene Name
1	Tutorial
2	FireMovement

4. Scripts

Name	GameObject Tagged To	Description
AllyMovement	AllyManager	Controls the movements of the 3 allies, including filtering and executing the voice commands.
ChangeScene	TUTORIAL: SceneControl	Change scene on button tap
Grading	Reticles	Prints the success message when all rooms are cleared.
NextCanvas	TUTORIAL: CanvasManager	Loops through the tutorial screens from 0 to 10 when menu buttons are pressed on the controllers.

PhysicsButton	TUTORIAL: Push	Mimics the physics property of the “Go to Game” button.
PlayerController	Player	Denotes the move and speed input components of the player controller movement.
Reticle1Trigger	Reticle 1	Denotes the properties of the reticles on trigger. Note that the Trigger component on the Box Colliders of these triggers in Unity GUI needs to be checked.
Reticle2Trigger	Reticle 2	
Reticle3Trigger	Reticle 3	
Reticle4Trigger	Reticle 4	
Reticle5Trigger	Reticle 5	
ShoulderTap	Shoulder	Controls the movement of the 3rd Ally when his shoulder is tapped. Moves him towards the door to open it.
ShowOneReticle	Reticles	Only show one reticle at any point in time. These reticles are checkpoint markers; show the next reticle only when room count has decreased.
Timer	TimerText (TMP)	Controls the timer’s properties.
TimerTrigger	ReticleMainDoor	Starts the count of the timer when the first checkpoint is active.
TriggerDoorController	Shoulder, L-Bomb Door, L-Door 1, L-Toilet Door, L-Door 2, 01_low (main door)	Automatically opens the door when the door is triggered by the 3rd Ally.
VRAnimatorController	PlayerCharacter	Controls the movement of the player controller’s body.
VRFootIK	PlayerCharacter	
VRRig	PlayerCharacter	

5. Game Concept

5.1. Introduction

This document details the Fire Movement VR training tool developed for HTC Vive consoles. This tool can be used to provide an insight into, and supplement the training of Section Commanders in an urban environment, specifically in voice communication.

Purpose
It aims to allow (1) soldiers to safely experience how to respond to threats in urban environments and (2) familiarise themselves with new combat environments.

Game Engine
The tool will be built using the Unity game engine.

5.2. Key Points

Console: HTC Vive
Genre: Action/Communication
Target audience: Sergeants/Officers of the Singapore Army.
Style: Simulates the environment of real-world army training.

Other glossary:
Each section consists of 7 men (1 commander, 6 men)

6. Game Mechanics

In the interest of time, the game mechanics will only focus on the core functionality of the game.

6.1. Core Gameplay

- The player will be playing as a commander, along with 6 other people (AI reacts to voice commands) in the section. The player is expected to make use of voice commands to guide and command the section during an urban fire movement situation.

The use of VR can enhance the learning experience through better preparing commanders to take charge of the section.

6.2. Game Flow

- **Tutorial Screen:** Player will be able to view the tutorial screens and press the right and left menu buttons on the controllers to control the screen number. Players will click on the “Go to Game" button when ready.
- **Mission:** The mission entails a main objective that the player has to complete. Metrics such as time taken, commands made are taken into consideration here, and displayed as a summary screen after each round.
- **Grading/Success Screen:** Mission success screen will be shown when all rooms are cleared.

6.3. Characters

The user will be playing as the commander in this game.

6.4. Enemies

Keeping to a 1:3 enemy ratio, there will be around 3 enemies deployed each round.

6.5. Player Controls

The player will be able to take the following actions:

- **Move:** Movement about the environment will be done through the controller's trackpad. The speed of movement is controlled through the Vector1 value of the trigger (0 - 1).
- **Crouch:** Physical player crouch movement.
- **Speech*:** Speech and keyword recognition will be used here. The player will speak into HTC Vive's built-in microphone to give commands to his fellow men (responds to commands).

7. Level Overview

HDB 4-Room Layout (Urban Environment)

The first environment modelled will be a typical HDB 4-Room Layout. In addition to it being a commander training tool, this is also useful for soldiers to visualise before they conduct their operations in a HDB environment.

Scenarios:

1. Commander will practice the basics of entering and clearing a room/apartment (movement technique) and conducting a breach (employing hand grenades and guns). Some breach techniques can include door breach, window breach and wall breach.

   There will be 3 enemies placed on the scene.

   The stage starts with the commander (player) and 3 assisting troopers. The troopers will execute the commands if a command is given.

   This scenario will be a basic simulation of how a typical room clearing situation will look like. As mentioned above, in future developments, we hope to see players either taking on multiple roles in urban warfare, or rather have a multiplayer feature that will ensure the whole section knows how to work together.

   The appropriate approach:

   Outside the apartment

   1. "Third man cover!" -> Third man goes into cover fire position.
   2. "First man scan!" -> First man will scan the door and report back how to break in the door. Replies "No I.E.D., no booby trap, door swing inwards".
   3. Section commander **taps** the 3rd man. The 3rd man moves to open the door.
   4. The 1st and 2nd man take position to clear the room. Section commander takes over the 3rd man's previous covering position to cover the group.

5. The operation starts when the section commander says “Clear room!”.
6. After the 2 men clear the room, the section commander rushes into the room and shouts “Room clear!”

There will also be a timer on the screen to record the time spent on the mission. The time starts when the user moves to the vicinity of the door. The splash screen will show when the room has successfully been cleared (success).

8. Interface

The interfaces required for this tool include: start screen, tutorial screen, success screen (with grading system) and a failure screen (with grading system). The breakdown of performance includes time taken to clear the mission, correct commands given, etc.).

Tutorial Screens

1) Use your left and right controller’s menu buttons to navigate through the instructions.
2) Welcome to the tutorial! There are a few things you need to know before you get started.
3) We will first start off with the basics of apartment clearing.
4) You will play as a section commander. The objective of this scenario is to clear the whole apartment using basic room clearing techniques.
5) Outside the Apartment: First, scout the apartment's surroundings. Figure out the kind of breach method to use. For this scenario, we will use a door breach method.
6) Outside the Apartment: At the main door, the commander (you) says "Third man scan!". The third man covers and scans the surroundings.
7) Outside the Apartment: Next, you say "First man scan!". The first man scans the door and reports the door's specifications. => “NO IED, NO BOOBY TRAP, DOOR SWING INWARDS.”
8) Outside the Apartment: You will have to tap the 3rd man for him to open the door. The operation starts when you shout "CLEAR ROOM!".
9) Clearing the Apartment/Room: The first man will choose a direction to head into. The second man will move in the opposite direction.
10) Clearing the Apartment/Room: After the room has been secured, the section commander (you) will rush into the room and shout "Room Clear!".
11) Clearing the Apartment/Room: For this scenario, clear all rooms in the apartment. The scenario will end when you finish clearing all the rooms.

9. Speech Recognition

For the speech recognition portion, the API library UnityEngine.Windows.Speech was used. The KeywordRecognizer API was used to recognise only certain words:

1) First (man)
2) Second (man)
3) Third (man)
4) Cover
5) Scan

6) Clear Room
7) Room Clear

In code,

```
private Dictionary<string, Action> keywordActions
        = new Dictionary<string, Action>();
private KeywordRecognizer keywordRecognizer;

keywordActions.Add("first", AllyChange); > etc.
```

10. Grading System

The current program's grading system is based on the number of commands received. This can be found in the AllyMovement.cs file.

```
void AssignGrades()
{
    if (commandCount > 20)
        grade = "A";
    else if (commandCount > 18)
        grade = "B";
    else if (commandCount > 10)
        grade = "C";
    else
        grade = "F";

    gradeText.text = String.Format("{0}", grade);
}
```

For future improvements, the grading system should work based on the completeness of commands; meaning whether the AI/allies understand the commands and correctly execute the steps.

11. Sound

No.	Name	Location
1	bell.mp4	Reticles
2	leftclear.mp4	Ally 1
3	rightclear.mp4	Ally 2
4	scan.mp4	Ally 1