

Solo Man’s Quest for the Volcano Idol

23.10.2023

Ehren Holgersson

Changelog

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 0.1.0 | 05/10/2023 | Initial Setup |
| 1.0.0 | 23/10/2023 | Completed |

# Contents

[Changelog 2](#_Toc1)

[Contents 3](#_Toc2)

[Introduction 4](#_Toc3)

[Rationale 4](#_Toc4)

[Background 4](#_Toc5)

[Terminology 4](#_Toc6)

[Proposed Design 4](#_Toc7)

[Non-Goals 4](#_Toc8)

[Software and Hardware Requirements 4](#_Toc9)

[System Architecture 5](#_Toc10)

[Data types 5](#_Toc11)

[Interface/API/Namespaces Definitions 5](#_Toc12)

[Which namespaces (Includes) did you include in your project? 5](#_Toc13)

[What functionality did each namespace provide to your code? 5](#_Toc14)

[Risks 5](#_Toc15)

[Alternatives 5](#_Toc16)

[Pseudocode 5](#_Toc17)

[System Pseudocode 5](#_Toc18)

[Evaluation 5](#_Toc19)

[Reflection 5](#_Toc20)

# 

# Introduction

## Rationale

I am trying to demonstrate my ability to utilize the AI NavMesh components in Unity to make multiple AI agents to navigate a maze in unique ways.

## Background

This document and the related project have been created primarily for the purpose of meeting the requirements outlined for AI skills assessment 1 for TAFE Certificate 4 in ICT Game Programming.

## Terminology

Level – A 3D environment the Agents will need to navigate.

Agent – Used to describe a unity object which will navigate (move) around the level

## Proposed Design

I intend for one AI agent to take the place normally filled by the player as the “Hero” character. This hero will have to navigate to the end of the level or maze whilst being chased by a boss “Ogre” agent, also avoiding other Ai agents patrolling the area (these will be spiders).

## Non-Goals

I would like the project to result in a completed game loop, despite there being no actual player input. One of the Ai agents will stand in for the player and complete the level.

## Software and Hardware Requirements

Hardware requirements should be fairly minimal. I would expect not much higher than unity3D requirements of windows 7 with x64 and Sse2 capable CPU and DirectX10 GPU.

# System Architecture

## Data types

## **enum PlatState { Top , Bottom, TravelUp, TravelDown }**

This is used to outline the state of the moving platform at the end of the level, and also used in some doors within the maze.

## **Vector3**

Used extensively to describe positions in space, for agent destinations as well as for agents to determine their positions relative to each other. See description Below.

**RayCastHit**

Stores result of raycast calculation, used in function to determine line of sight between agents.

## Interface/API/Namespaces Definitions

### Unity components/Classes in Use

**NavMeshAgent**

This component moves an object within the scene along a path based on the NavMesh. Refer to <https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html>

**Animator**

Controls the state of the animation system on an object. See <https://docs.unity3d.com/ScriptReference/Animator.html> and <https://docs.unity3d.com/Manual/class-Animator.html>

**GameObject**

GameObject is the base class for creating custom scripts which attach to unity GameObjects as components withing the editor. <https://docs.unity3d.com/ScriptReference/GameObject.html>

**Vector3**

Used to describe a position or direction in three dimensional space. <https://docs.unity3d.com/ScriptReference/Vector3.html>

**Camera**

The camera component defines the reference point used to draw the world, either directly to the screen or to a texture to be used for other purposes. <https://docs.unity3d.com/ScriptReference/Camera.html>

**Image**

A visual element that displays a texture. <https://docs.unity3d.com/ScriptReference/UIElements.Image.html>

**TextMeshProUGUI**

This is the Canvas/UI version of a TextMeshPro class. This displays text using Unity’s Text Mesh Pro system which replaces the older text systems. <https://docs.unity3d.com/Manual/com.unity.textmeshpro.html>

### Which namespaces (Includes) did you include in your project and what functionality did each namespace provide to your code?

**UnityEngine**

Adds basic unity classes such as GameObject, ScriptableObject and the various components.

**UnityEngine.AI**

Adds Navmesh agent class, used to control AI agents.

**UnityEngine.UI**

Adds canvas functionality used to display UI elements in game.

**System.Threading.Tasks**

Used to create asynchronous method or function (alternative to coroutine)

**System.Collections**

Adds IEnumerator which is required for coroutine. Both asynchronous methods and coroutines used in different parts of project.

## Risks

I am fairly confident in my ability to complete this project and know it can be done with the unity NavMesh/Ai tools. The primary risk is likely to be the ability to complete the work in the time available.

## Alternatives

N/A

# Pseudocode

## System Pseudocode

**Agent Logic:**

**Hero Class:**

If Enemy is visible

Increase fear

Agent speed = running speed

Reduce stamina

Else

Reduce Fear

If fear is greater than 50%

Agent speed = running speed

Reduce stamina

Else

Agent Speed = walking speed

Increase Stamina

If Area contains required collectable

Move to next required collectable

Else

If area contains optional collectable

If fear > 50%

Remove next optional collectable from list

Else

Move to next optional collectable

Else

Move to Area Goal

**Ogre Class:**

If Player is in sight

Move Directly to Player

Set uncertainty to zero

Else

Increase uncertainty

Pick a random spot within distance of player determined by uncertainty and move there

If Player is in attack range

Play attack animation

Reduce player health

**Spider Class:**

If Player is in sight

Move towards Player

Else

Move to random spot within patrol area

If Player is in attack range and Random Number is greater than some amount

Play attack animation

Reduce player health

# Evaluation

## Reflection

I am fairly happy with the final version. I feel that it meets all the requirements for this project. However if I were to revisit this in the future there are a few additions and changes I may make.

Animation state of the characters moves between running and idle based on movement, but the speed of the animations is not linked to the movement speeds. This results in agent’s feet appearing to slide on the ground somewhat.

I would also move the Hero character’s fear and stamina logic from the AI coroutine to Update or FixedUpdate as I feel it would be more appropriate there.