

ABSTRACT

This is a technical document for approval for AI for games. The project is a mini stealth game with patrolling guards, and the player must find their way through the scene and to the end.

Name Withheld

AI for Games

STEALTH AI TECH DOCUMENT

AIE AI Assignment 2017

Introduction

Game introduction

The project I am putting forward is a mini stealth game. The idea of the game is the player will have to find their way from one side of the scene to the other. The player will not be able to attack and if the guard catches the player the game is over.

The main inspiration for this game is the scene in Zelda Ocarina of time when link is making his way through the temple gardens.

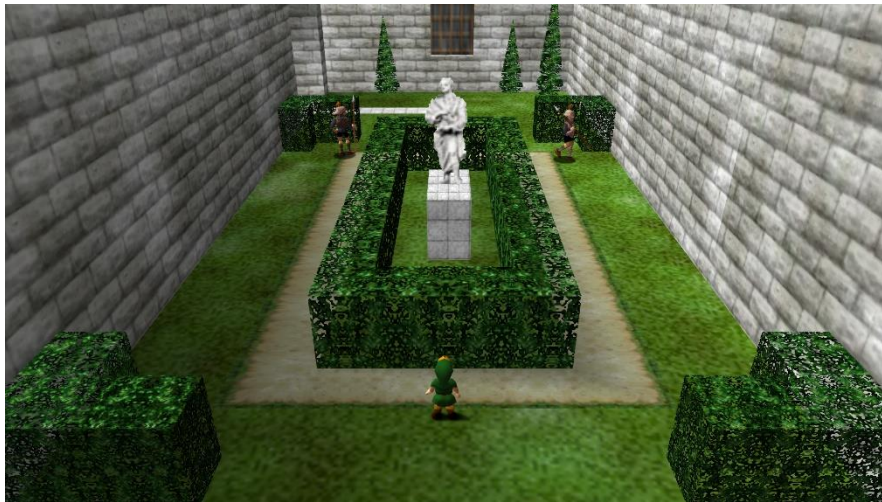


Figure 1: Scene from Zelda Ocarina of time

AI Elements

The game will have the following elements of AI behaviour.

- Keyboard control
- Patrol
- Search
- Chase
- Collision detect
- A* Pathfinding

Actors

There will be the following actors in the Scene.

- Player
 - The player will be controlled by WASD on the keyboard.



Figure 2: Player Sprite

- Guard
 - Guard will be patrolling
 - If guard hears a noise, will investigate the position he heard the noise
 - Guard will search that area for a few second if nothing is found when he gets to that area.
 - If guard sees the player he will Chase the player
 - If he catches the player then the game is over



○

Figure 3: Guard Sprite

Map

The map will be a castle like theme. Below is a Draft image this is not final.

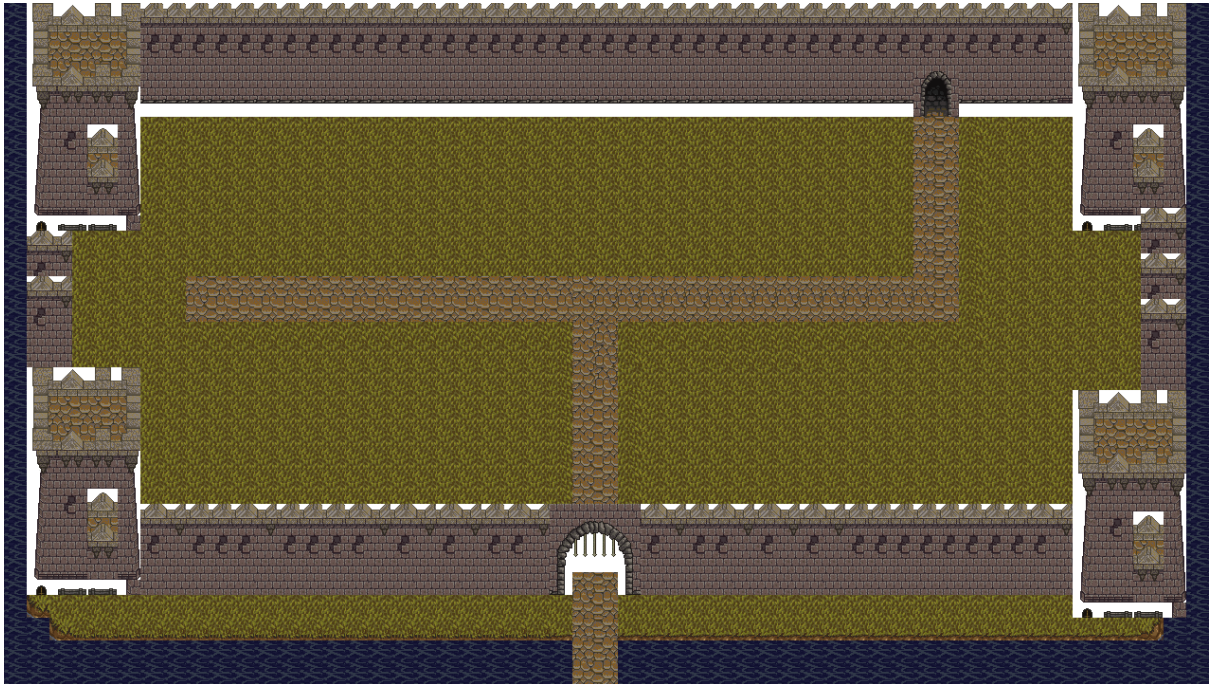


Figure 4: Map Draft

Technical Information

Technical Information

Will be coded using Visual Studio 17.

Language used is C++.

The map will be Orthographic 2D.

Key AI Mechanics

Keyboard Control

This will be a simple keyboard control that will move the character up down left right.

Patrol

This the default state for the guards. The actor will simply move between a set of Waypoints. If the guard sees the player while patrolling, then they will Chase the player.

Chase

If the guard can see the player, the guard will chase the player. Using ray casting if the ray detects a player then they are in view or if the player get too close. The guard will chase the player until vision has been broken for a few seconds. There will be a mix of pathfinding and collision detection while the chase is happening. If vision has been broken then, the guard will search the last known position of the player.

Search

After the guard has lost the player, if nothing is there he will hold the position for a few seconds. After a few seconds if nothing is found, then the guard will head back to his patrol path and begin patrolling again. There will be some pathfinding required to move back into the patrol position. If the player again gets too close or is spotted again then the guard will chase once again.

Collision Detection

All objects will need some sort of collision detection. All guard actors will have a Raycast that points in the forward direction. All the walls or impassable objects will have an Axis Aligned Bounding Box. If the raycast detects the AABB then the guard will not be able to pass through.

Pathfinding

The guards will need path finding to avoid objects when patrolling, searching or chasing the player. I will be using A* as the pathfinding algorithm. While patrolling and searching, the guard will need to find paths to fixed points so that is fine. However, when chasing the player, the pathfinding will need to be updated as the player runs away. This will be achieved by constantly updating the guards path as the player runs away.

Guard behaviour tree

