**AT03 Production Diary**

**1.1.1 A\* Algorithm Research**

A\* is one of the best path-finding algorithms for finding the most optimal path to a goal/destination. A\* uses the lowest F Cost to choose the next node to go to & repeats till it gets to the goal then returns the lowest F Cost path to be used.

**A\* Terminology Definitions**

**F Cost:**

This is the cost for moving to the node. It is the g + h costs.

g = the movement cost to move from the start to a given node that follows the generated path to get there.

h = the estimated movement cost to move from a given node to the goal.

**Heuristic:**

Is the estimated movement cost to move from a given node to the goal.

Euclidean Distance Example:

h = sqrt ( (current\_cell.x – goal.x)2 +

(current\_cell.y – goal.y)2 )

**Priority Queue:**

It is a queue that arranges the nodes by their priority values.

**1.2.1 Specifications and Software Analysis**

**3D Model Export Specifications:**

FBX file format & for the model to have an armature.

**Character Animation Specifications:**

The model should have an armature.

**Animation Features, Toolsets, and Capabilities in Unity:**

Unity animator & state machine

**1.2.2 Unity Scene Navigation**

**Panning the viewport camera:**

Left & right or A & D to pan. Left-click & drag work as well.

**Zooming the viewport camera in/out:**

Mouse Scroll wheel or Alt & right-click

**Rotating the viewport camera:**

Right-click & drag.

**1.2.3 Enhancing Workflow in Unity**

**Keyboard Hotkey/Shortcut Procedures:**

E for Rotate Tool, R for Scale Tool, Ctrl+Z for Undo

**Navigational Input Procedure:**

W,A,S,D,Q,E to move in 3D viewport. Arrow keys for menus.

**1.3.1 HCI Device Integration Research**

You would need to know the controller mapping.

**Potential HCI Devices:**

Mouse & Keyboard, Xbox Controller, PlayStation Controller

**Selected HCI Devices and Associated Control Schemes:**

Xbox Controller, Mouse & Keyboard

**1.3.2 Planned HCI Integration**

**C# Event System Summary:**

It is to handle methods or delegates within the event. In C# events are encapsulated delegates, which means it is dependent on the delegate that it encapsulates. The class that raises the event is called a Publisher, & the class that receives the notification is called a Subscriber. A single event can have multiple subscribers.

Event

Delegate

Event Handler

Subscribe: Event += Event\_Handler

Subscriber

Publisher

Subscribe Events

Fires Events

Event handler signature must match with event delegate signature

C# Event system will be used to determine what state the AI will be in:  
Wonder (is moving & not chasing the player)  
Chase (Player is in the detection area)  
Idle (is not moving & not chasing player)  
Stun (Player interacted with me)  
Force Chase (Player has all objectives)

**Unity GUI Library Reviews:**

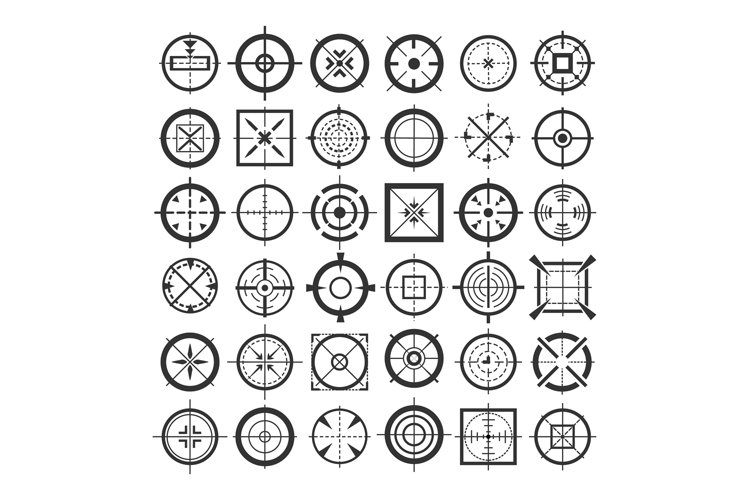
uGUI (Unity UI) is an in-game UI & IMGUI (Immediate Mode GUI) is for the Unity editor.

uGUI is for Technical Artists because it is an object-based UI system.

IMGUI is for Programmers as its main focus.

Reference: https://docs.unity3d.com/Manual/UI-system-compare.html

**1.3.3 UI Widget Example Overviews**

Crosshair examples [](https://www.google.com/url?sa=i&url=https%3A%2F%2Fdesignbundles.net%2Fvectortatu%2F1202554-target-crosshair-icons&psig=AOvVaw0Bs0pXyW8OGTTLanEsdCah&ust=1701312580916000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCKD13Pab6IIDFQAAAAAdAAAAABAE)

Objective examples[](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.gameuidatabase.com%2Findex.php%3F%26set%3D1%26tag%3D1%2C151&psig=AOvVaw2JXhH2OvTA5bwi1IxILt6K&ust=1701312966303000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCJDj-Oma6IIDFQAAAAAdAAAAABA1)

**2.1.1 AI Behaviour Chart**

*Insert your behaviour chart for the AI here.*

Idle

Chase

Wonder

Stun

**2.1.2 AI Design Review**

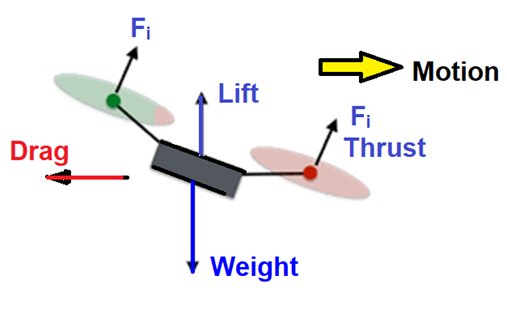
**Review of AI Design Feasibility:**

The AI is quite feasible as long as the transitions are smooth & work.

**AI Production Timeline and Cost Estimates:**

Time to make AI: 1 hour  
Time to make NPC: 3 hours  
Hourly pay = $32.90  
Total cost = $32.90 \* (1+3) = $131.6

**2.2.1 Animation Reference Materials**

[Image Reference](https://www.google.com/imgres?imgurl=https%3A%2F%2Fcfdflowengineering.com%2Fwp-content%2Fuploads%2F2020%2F09%2Fword-image-37.png&imgrefurl=https%3A%2F%2Fmappingmemories.ca%2Fpyrcezvac1042827&tbnid=q3m7n8upKgTwSM&vet=12ahUKEwiut-Ce9Yb5AhW1yaACHdxWAVsQxiAoAXoECAAQHQ..i&docid=QyNhBe7gWqKXZM&w=507&h=311&itg=1&q=how%20drones%20fly&ved=2ahUKEwiut-Ce9Yb5AhW1yaACHdxWAVsQxiAoAXoECAAQHQ)

<https://youtu.be/4kjA473swIo?t=205> <https://youtu.be/4kjA473swIo?t=328>

**2.3.1 UI Widget Paper Prototype**

*Insert the paper prototypes for the required UI widget here.*

Clicking Play would send you to the game

Clicking Help would bring up the help menu

Play

Help UI

Clicking Quit will close the game.

Help

Quit

In-game UI

Objective

The crosshair will change colour when you're looking at something you can interact with & if it is in range.

**3.1.2/3.3.2 Testing Log**

*Please add rows as required.*

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Description | Expected Results | Actual Results | Success? |
| I press down the W key &or move right joystick up on gamepad | For the player character to move forward | Player character moved forward | Yes |
| I press down the S key &or move right joystick down on gamepad | For the player character to move backward | Player character moved backward | Yes |
| I press space bar &or A button on Xbox controller | For the player character to jump | Player character Jumped | Yes |
| Pressing the E key &or the Y button on Xbox controller when able to interact with something | For the game to register the interaction | It registered the interaction | Yes |
| When looking at an interactable the crosshair changes colour. | For the crosshair to change colour when able to interact | The crosshair changed colour when the player character looked at an interactable in-range | Yes |

**3.2.2 Animation Troubleshooting**

**Error Scenario Summary:**

Animation transitions aren’t smooth.

The animations don’t change & are stuck in endless loops.

**Relevant Official Unity Documentation:**

<https://docs.unity3d.com/2020.3/Documentation/Manual/StateMachineTransitions.html>

<https://docs.unity3d.com/2020.3/Documentation/Manual/class-Transition.html>

**Relevant Unity User Feedback:**

<https://forum.unity.com/threads/problems-with-states-and-transitions-in-the-animator.1255572/>

**3.3.2 Web Browser Testing**

**Web Build Running in Mozilla Firefox:**

Yes, it does & it works correctly.

**Web Build Running in Google Chrome:**

Yes, it does & it works correctly.

**4.1 Required Amendments**

Increase Player & enemy Speed.

Confirm controller mapping.

**4.2 Final Review**

**AI Evaluation:**

The AI works like it should from the design brief. I made it seem like it cursed when it gets stunned.

**Final Checks:**

|  |  |
| --- | --- |
| **Final Checks** | **Confirmed** |
| * Camera movement responds to relevant inputs | ✔ |
| * Player movement responds to relevant inputs | ✔ |
| * Player interaction responds to relevant inputs | ✔ |
| * AI path finding (using the A\* algorithm) has been successfully integrated | ✔ |
| * The AI behaviour is defined by four states – idle, chase, stun, and either patrol or wander. | ✔ |
| * The logic of all AI behaviour states aligns with the game brief | ✔ |
| * AI successfully and appropriately transitions between behaviour states | ✔ |
| * Chase state becomes the default state for the AI when the objective item is interacted with | ✔ |
| * AI transitions to stun state in response to player interaction | ✔ |
| * AI 3-D model and animations have been successfully integrated for each behaviour state | ✔ |
| * AI audio clips have been successfully integrated for each behaviour state | ✔ |
| * Game win conditions have been successfully implemented | ✔ |
| * Game loss conditions have been successfully implemented | ✔ |
| * UI prompt displays appropriate message based on game loss or victory | ✔ |
| * UI widget responds to relevant keyboard inputs | ✔ |
| * UI widget responds to relevant mouse inputs | ✔ |
| * UI widget responds to relevant controller inputs | ✔ |
| * UI set to scale with a full HD resolution (1920x1080) | ✔ |
| * UI objective text updates appropriately | ✔ |
| * UI ‘new game’ button successfully loads game scene | ✔ |
| * UI ‘info’ button successfully displays a window with some game information | ✔ |
| * UI ‘quit’ button successfully closes the application | ✔ |
| * Appropriately compatible with Windows | ✔ |
| * Appropriately compatible with Google Chrome web browser | ✔ |
| * Appropriately compatible with Mozilla Firefox web browser | ✔ |

**4.3 Final Client Sign-Off**

*Insert a screenshot of your email communications with the client, providing evidence of their endorsement to finish the production of the project.*

Observation Checklist



**Repository Link:**

Itch: <https://stampard0.itch.io/at03>

Password: Drone

Repository: <https://github.com/Stampard0/AT03_Indie_Game>