IGB383 AI for Games

Assignment 3

Competitive Bot AI using Behaviour Trees

|  |  |
| --- | --- |
| **Assessment Overview** | |
| **ITEM:** | **Assignment 3: Competitive Bot AI using Behaviour Trees** |
| **WEIGHTING:** | 30% (25% for Part A and 5% for Part B) |
| **OBJECTIVES:** | Unit Objectives 1, 2, 3 & 4 from unit outline |
| **DUE DATES:** | **Friday June 10th 2022** |
| **DESCRIPTION:** | This is an individual assignment. Your deliverable for this assignment will be in two parts archived into a single .**zip** file and submitted via **Blackboard**:   1. A practical component. 2. A statement of completeness, issues and how you solved those issues.  Part A – Practical Component For this part, you are required touse Unity3D. You will expand upon the provided assignment base to setup Behaviour/Decision Trees for selecting complex behaviours and firing weapons according to the outline in this document and worked on during Workshops.  **Development Features**: You are required to implement two primary components for this assignment component, that being:   * 1. Behaviour Trees: Implement working decision tree constructs that are capable of selecting between major behavioural states (leaf nodes) using your own personalised logic. Behaviour trees will control both NPC actions and weapon firing behaviours.   2. NPC Behaviours: Tweak or improve upon the proved AI solution to create your own unique actions for behavioural selection. A degree of freedom is allowed here, however NPC actions achieved via cheating or overt exploitation will be viewed negatively.   Details as to how these models will translate appropriately into gameplay elements will be discussed during Lectures and Workshops.  **Important:** Your implementation should not exceed anything beyond the single script MechAIDecisions and associated BT text file. Changes or additions to the game beyond this will disqualify you from the Week 13 tournament and will be viewed critically upon further inspection. Part B – Written Component In this part of the assignment, you will each write a statement of completeness along with any issues that you encountered, your theories on these issues and how you solved the issues. |
| **CONTENT:** | You will submit via Blackboard in a .**zip** file the following items:   1. The Unity project executable contains your solution to Part A, including script files. 2. Part B – Your statement of completeness.   **All submissions completed via Blackboard.** |
| **FORMAT:** | Part B: Short report should have the following sections:   1. **Title page**: Your name, student number and unit code (some students are postgraduates) 2. **Statement of Completeness**: What was/was not completed in the assignment. 3. **Behaviour Trees**: A section where you discuss your implementation of your Behaviour Tree models, displaying your understanding of their design and utility. 4. **NPC Behaviours**: A section where you discuss your tweaking or implementation of customised NPC behaviours, including the use of any specialised code and logic necessary to undertake them. 5. **Conclusion**: A summary of what you have learned.   Hint: Diagrams in your discussions will go a long way in conveying your understanding in sections c and d. |
| **BOT AI TOURNAMENT:** | This assignment will conclude with a small Bot AI Tournament in Week 13 during the scheduled workshop times. There will be a prize for the best performing Bot AI. Cheating and exploitation will be viewed critically and result in disqualification. |

# Requirements

You are required to implement the following components within the provided assignment base:

1. Behaviour Trees – Branching decision making construct capable of making logical and intelligent decisions. Design and rationale of Behaviour Trees is the focus here, emulating player-like behaviour.
2. NPC Behaviours – Customised Bot AI actions, including supporting logic and inferences, denoting final bot activity. These NPC behaviours should be viewed as the leaf nodes in your Behaviour Tree.

**Note:** Your implementation should not exceed beyond additions made within the script MechAIDecisions and associated Behaviour Tree text file. Any additions or requirements beyond this will result in disqualification from the tournament. Tournament results have no impact on final grades.

## Behaviour Trees

Your scene should contain the following components related to the implementation of Behaviour Trees:

1. **Behaviour Tree for NPC Actions:** A primary behaviour tree for transitioning between primary leaf node states (e.g. Attack, Roam etc.).
   1. A working behaviour tree, converted from a provided FSM solution, will be given to you via workshop tutorial content (Panda BT). You are encouraged to tweak and enhance this baseline behaviour tree further. Some guidance in workshops will be given here.
   2. The behaviour tree will work in conjunction with logical decision tree operators and methods to garner information about the game world. Some of these are provided to you already, others you may create yourself.
   3. You are encouraged to tweak the baseline behaviour tree, to achieve results that are interesting and player-like. An example: you may have logic set up to fluctuate randomly between two states when ‘indecisive’, use heuristics to determine the sensibility of an action, or even perform multiple tasks simultaneously. Bonus marks will be given for ingenuity here.
2. **Behaviour Tree for Weapon Firing:** A secondary behaviour tree for controlling the operation of the NPCs weapon systems.
   1. The base Mech NPC avatar is set up with an array of weapons that use resources and have cooldowns. Additionally, certain weapons are better for specific ranges, or have properties such as locking-on or explosive AOE damage.
   2. The secondary behaviour tree should expand on and replace some very simple weapon firing logic provided within the assignment base. This provided code should act as a guide for firing behaviour, but you have some creative freedom here to do what you think is most player-like
   3. You should consider setting up this behaviour tree so that multiple firing activities can operate simultaneously, or to work in conjunction with the NPC Actions behaviour tree (i.e. action driven). As an example, one of the weapons is a short range laser beam that chews energy quickly but has high damage. You probably do not want to use this weapon at long range when you are fleeing.
   4. Careful consideration for ammunition/energy levels or even environmental factors should be considered here.

## NPC Behaviours

Your solution should aim to have the following:

1. **Customised NPC Behaviours**: You are provided initially with a FSM bot AI solution in the assignment base. This will be quickly converted to use a decision tree via practical content. However, you should aim to further tweak this solution for more realistic, player-like and higher performing behaviours.
   1. As an example, the Attack state/leaf node is adequate, but could be improved. It has some simple logic allowing it to ‘strafe’ randomly with a timer controlling the frequency. However, this mechanic operates on predictable movement vectors, is cumbersome, and does not account for what the enemy target may be doing. Circle strafing around the enemy target would be a much more interesting pattern, provided there was room to do so.
   2. Again, your implementation should not exceed anything beyond the single script MechAIDecisions. This includes additional public variables that would need to be set in Unity.
2. **Realistic Player-like Behaviour**: Your solution should aim to implement Bot AI behaviour that is difficult to distinguish from an actual player interacting with the game. Your implementation should consider the following:
   1. It is easy to program a bot to cheat or exploit mechanics, such as teleporting or being able to see through walls. Such implementations will be viewed critically, not just because they are simpler to implement, but because they are non-player-like.
   2. Specific AI methods have been set up and utilised in separate classes (e.g. MechAIAiming) within the default assignment base already. You are free to use and combine these methods together to ask intelligent questions regarding behaviour tree processes or within Action states themselves. These methods are designed to eliminate exploitation, by setting up a way to target the closest player that is within line of sight and within the field of view.
   3. Winning the Bot AI Tournament is not the focus of the Assignment. Focus on creating a realistic Bot AI first. No marks will be deducted for realistic, player-like bots that do not necessarily score well.

## Bot AI Tournament

This event is for fun and is not mandatory. Disqualification or non-participation in this event will not impact your final grades. During this event, all valid student designed bots will compete in a single-elimination, 4 player setting with advancing brackets. The winner of each bracket advances to the next until a final round and highest scoring bot AI is identified. The author of the best performing bot AI will win a prize, to be disclosed closer to the date.

Please note that unsatisfactory Bot AI performance will have no impact on your final grade. Arguably, the highest performing bot may in fact be the least appropriate for an actual game implementation. Further discussion around this topic will occur during the Lectures and Workshops preceding this event.

The use of scripting that could be construed as cheating or exploitative will be critically observed by both your assessor (scripts) and peers (observers) and may be flagged for further observation during the marking process (i.e. not player-like). Obvious exploitative behaviour will result in tournament disqualification.

Examples of obvious exploitative bot actions include the following:

* Increasing damage, health, speed, ammunition etc.
* Teleportation, flying, invincibility etc.
* Direct manipulation of enemy NPCs
* Direct manipulation of environmental elements (e.g. patrol points)

Additionally, any player scripts that simply do not compile or integrate with the tournament build will be omitted from the tournament.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IGB383 Assignment 2: MARKING CRITERIA**  The assignment is designed to assess certain aspects of the objectives for this subject  Performance standards and criteria for each objective appear below | | | | | | |
| **Criteria** | **Performance Standards** | | | | | |
|  | **High Distinction** | **Distinction** | **Credit** | **Pass** | **Fail** | **Mark** |
| **Specifications** | Adhered to every requirement, and did some additional research. | Did some research but missed one minor requirement Or  Did no additional research but adhered to every requirement | Missed several minor requirements.  Or  Did some research but missed one major requirement | Missed a major requirement. | Missed many major requirements. |  |
| 15 – 13 marks | 12.5 – 11.5 marks | 11 - 8 marks | 8 – 6.5 marks | 6 – 0 marks |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IGB383 Assignment 2: MARKING CRITERIA**  The assignment is designed to assess certain aspects of the objectives for this subject  Performance standards and criteria for each objective appear below | | | | | | |
| **Criteria** | **Performance Standards** | | | | | |
|  | **High Distinction** | **Distinction** | **Credit** | **Pass** | **Fail** | **Mark** |
| **Visual Affect** | Player-like Bot AI is observable with very good use of behaviour trees, resulting in interesting and varied NPC behaviour. | Player-like Bot AI is observable with good use of behaviour trees, resulting in somewhat interesting and mostly varied NPC behaviour. | Player-like Bot AI is observable. Decent use of behaviour trees, resulting in adequate NPC behaviour. | Player-like Bot AI is mostly observable. Some use of behaviour trees, resulting in tolerable NPC behaviour. | Player-like Bot AI is not observable. Improper or no use of behaviour trees. NPC behaviour is exploitative or uninteresting. Not suitable for production. |  |
| 10 – 9 marks | 8 marks | 7 marks | 6 – 4 marks | 3 – 0 marks |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IGB383 Assignment 2: MARKING CRITERIA**  The assignment is designed to assess certain aspects of the objectives for this subject  Performance standards and criteria for each objective appear below | | | | | | |
| **Criteria** | **Performance Standards** | | | | | |
|  | **High Distinction** | **Distinction** | **Credit** | **Pass** | **Fail** | **Marks** |
| **Communication**  relates to:  Outcomes 1, 2 & 3 in the unit outline | Gave an excellent few paragraphs on what was learned from this exercise by stating the problems, theories and solutions. | Gave a very good few paragraphs on what was learned from this exercise by stating the problems, theories and most of the solutions. | Gave a good few paragraphs on what was learned from this exercise by stating the problems, theories but not quite getting all of the solutions. | Gave a few paragraphs on what was learned from this exercise by stating the problems, theories but no real solutions. | Gave one or zero paragraphs on what was learned from this exercise. State the problems but did not have any theories. |  |
| 5 – 4.5 marks | 4 marks | 3.5 marks | 3 – 2 marks | 1.5 – 0 marks |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **IGB383 Assignment 2: MARKING CRITERIA**  The assignment is designed to assess certain aspects of the objectives for this subject  Performance standards and criteria for each objective appear below | | | | | | | |
| **Criteria** | **Performance Standards** | | | | | | |
|  | **High Distinction** | **Distinction** | **Credit** | **Pass** | | **Fail** | **Marks** |
| **Software**  relates to:  Outcomes 3 & 4 in the unit outline | All scripts compile. Clean programming style. Correct & efficient code. No logical errors. | All scripts compile. Good programming style. Correct code. No logical errors. | All scripts compile. Reasonable programming style. Correct code. Non-invasive logical errors. | 80% of scripts compile. Minor programming style issues. Correct code. Some minor logical errors. | | Less than 50% of scripts compile. Major programming style issues. Incorrect code. Major logical errors. |  |
| 10 – 9 marks | 8 - 7 marks | 6 - 5 marks | 4 - 3 marks | | 2 – 0 marks |  |
| **Comments:** | | | | | Total Mark Awarded | |  |