St Mary's University Twickenham London

CPS5002 – Artificial Intelligence – Assessment Brief

Module Code:	CPS5002
Module Title:	Artificial Intelligence
Module Convenor:	Prins Butt
Module Level:	5

Assessment Number:	2
Assessment Title:	Techburg
Assessment Weight:	60%
Assessment Individual/Group:	Individual
Assessment Type:	Software Artefact with Report
Assessment Time/Word Count Restrictions:	1 software artefact 2400 words report
Assessment Time/Word Count Limit Consequences:	It is essential that assignments keep within the time/word count limit stated above. Any work beyond the maximum time/word length permitted will be disregarded and not accounted for in the final grade.

Issue Date:	4th November 2024
Hand in Date:	17th January 2024
Planned Feedback Date:	Within 3 working weeks
Mode of Submission:	Online via Moodle
Number of copies to be submitted:	1 copy of each of the following: • a report in pdf format • a zip file containing your software artefact

Author:	Prins Butt
Internal Moderator:	Harshil Joshi
Moderation Date:	28th October 2024

Table of Contents

Introduction....3
Requirements....4
Tasks....6
Environment and Tools....8
Submission....9
Assessment Criteria....10
Learning Outcomes....11
Regulations, Policies, and Guidelines....13

Introduction

In the forsaken city of Techburg, remnants of a once-thriving human civilisation, only autonomous machines roam. With humans long gone, a small collective of survivor bots is left to scavenge spare parts vital to their survival. These survivor bots must navigate the eerie landscape of derelict buildings, carefully avoiding malfunctioning security drones and scavenger swarms that now dominate the city. The drones, no longer under human control, react erratically to movement, pursuing any entity they detect, while the scavenger swarms consume electronic devices they encounter. To endure, the bots must gather as many parts as possible and deliver them to recharge stations hidden within the city. Will the bots succeed in navigating the dangers of Techburg, or will the city's hostile inhabitants thwart their mission? The desolate streets await.

Requirements

You have been assigned the task of developing a software system to simulate the activities within Techburg. The system must meet the following requirements:

(a) Techburg City:

The city should be represented as a 2D grid, where each cell may be empty or contain spare parts, a bot, a recharge station, a malfunctioning drone, or a scavenger swarm. The 2D grid should wrap around at the edges.

(b) Spare Parts:

In the Techburg grid, spare parts are randomly scattered and come in three sizes: small, medium, and large. Each size offers different enhancements for survivor bots, with small parts boosting abilities by 3%, medium parts by 5%, and large parts by 7%. However, these spare parts corrode over time, losing 0.1% of their enhancement value each simulation step. Once placed in a recharge station, a spare part stops corroding and gradually recharges back to its maximum enhancement value.

Scavenger swarms, composed of autonomous nanobots, pose a constant threat by consuming spare parts to boost their collective energy. Small parts increase energy by 1%, medium by 2%, and large by 3%. Once consumed, these parts are removed from the simulation.

(c) Survivor Bots:

Survivor bots are resilient machines designed to navigate the Techburg grid in search of spare parts. Each bot can move to any adjacent cell, detect nearby parts, collect them, and transport them back to the nearest recharge station. They can carry only one part at a time and will prioritise larger parts that pose a lower risk when making choices. Survivor bots have limited energy, depleting by 5% with each movement. They consume parts at recharge stations to restore energy, unless their energy is critically low (5% or less), in which case they will immediately consume a part. For example, consuming a small part restores 1% of energy per simulation step until fully consumed. If a survivor bot's energy reaches zero, it becomes inactive and will be removed from the grid after a few simulation steps unless recharged.

Collaboration is key among survivor bots, as they can share parts and work together to gather more efficiently. While at recharge stations, they can rest to regenerate energy at a rate of 1% per simulation step without consuming parts. Survivor bots can be classified as either repair bots or gatherer bots. When both types are present at a recharge station, there is a 20% chance of creating an additional gatherer bot

(costing 30% energy from each) and a 5% chance of creating an additional repair bot (costing 50% energy from each).

Survivor bots can upgrade their abilities by consuming spare parts at recharge stations. These enhancements improve their efficiency, enabling increased movement speed, detection range, or energy capacity. Typically, survivor bots move at a medium pace, acting every second simulation step. However, if they achieve a speed enhancement of 51-100%, they will move every simulation step. The maximum speed enhancement a survivor bot can attain is 100%.

Survivor bots can also detect neighbouring cells. With a vision enhancement of 51-100%, they can detect parts up to two cells away, while an enhancement of 101-150% allows detection up to three cells away. The maximum detection range enhancement is 150%. Ultimately, the aspect of speed, vision, or energy that survivor bots choose to enhance is up to their strategic preferences and immediate needs.

(d) Recharge Stations:

Recharge stations are fixed locations within the Techburg grid. Each station can hold up to five survivor bots at a time. Survivor bots can store collected parts in the recharge station and consume stored parts to restore energy and enhance abilities. When multiple survivor bots occupy the same station, they share information about known locations of parts, recharge stations, drones, and scavenger swarms.

(e) Malfunctioning Drones:

Techburg is patrolled by malfunctioning security drones that autonomously roam the grid. Drones will detect any survivor bot within three cells and may pursue it. A pursuit depletes a drone's energy by 20%, so drones can only chase for short distances. If a drone catches a survivor bot, it can decide how to attack the survivor bot. It may deliver a shock, reducing the survivor bot's energy by 5% and causing it to drop any carried spare part, or it may try to disable the survivor bot, reducing its energy by 20% and causing it to drop the spare part. Depending on the type of survivor bot, the drone may perceive it to be a greater threat and destroy it completely. The survivor bot remembers the location of any dropped spare part and may retrieve it if it survives the encounter. Drones enter hibernation mode when their energy is at 20% or less, recharging at a rate of 10% per simulation step until they are fully recharged before continuing.

(f) Scavenger Swarms

Scavenger swarms are clusters of nanobots designed to consume decaying electronic and mechanical materials. Originally created to clear waste, these swarms now autonomously roam the city, dismantling any inactive bots or parts they encounter. They emit a decay field that causes nearby bots and drones to lose 3% energy per simulation step when within one cell of the swarm, due to its electromagnetic influence. When scavenger swarms come close to each other, they merge to form larger swarms, which can consume resources more rapidly. Additionally, if a swarm gathers enough material, it can self-replicate, creating new swarms to further enhance its scavenging capabilities.

(f) Simulation:

The simulation should begin with a Techburg grid of at least 30 by 30 cells, populated with recharge stations, survivor bots, spare parts, drones, and scavenger swarms. Each agent should act according to its attributes and behaviours on each simulation step. The goal of the simulation is to maximise the parts collected and enhancements gained by bots, with minimal survivor bot losses. The simulation ends when all spare parts are either collected or corroded, or all survivor bots have been eliminated without the possibility of replication.

Tasks

You are required to develop a software system that meets the requirements stated above.

For a pass grade (40 – 49):

1. Basic Application:

Software Artefact Criteria:

- The basic simulation should consist of recharge stations, survivor bots, and spare parts. The survivor bots should collect the spare parts and deposit them to recharge stations.
- The solution should be implemented using object-oriented programming.
- Version control should be used to track and evidence the development of the software artefact.

Report Criteria

- Provide a detailed summary of your implementation with some suitable justifications and supporting evidence including relevant diagrams.
- Include evidence of version control e.g., a detailed commit history.

For a 2:2 grade (50 – 59):

2. Comprehensive Solution:

Software Artefact Criteria

- The simulation should provide the implementation for recharge stations, survivor bots, and spare parts and make a strong attempt at implementing the drones.
- The solution should demonstrate the application of good object-oriented principles and techniques.
- Appropriate version control techniques (e.g., feature branches) should be used to track and evidence the development of the software artefact.

Report Criteria

- Detail the implementation with supporting evidence and rationale.

- Detailed evidence of version-control demonstrating strong version-control practice.

For a 2:1 grade (60 – 69):

2. Comprehensive Solution:

Software Artefact Criteria

- All aspects of the assessment must have been attempted.
- The solution should demonstrate the application of strong object-oriented principles and techniques.
- Strong version control techniques should be used to track and evidence the development of the software artefact.

Report Criteria

- Detail the implementation with supporting evidence and rationale.
- Extensive evidence of version-control and other evidence demonstrating ownership and development of the software artefact.

For a 1st class grade (70 – 100):

3. Advanced Application:

Software Artefact Criteria

- Must address all the requirements from a) – f).
- Demonstrate significant implementation of artificial intelligence. This may involve utilising multiple different techniques to enhance the agents or the simulation.
- Demonstrate strong object-oriented design and development.
- Demonstrate extensive utilisation of version-control and other professional development techniques.

Report Criteria

- Provide a comprehensive explanation and evaluation of your solution with strong supporting evidence.
- Evidence excellent project management.
- Strong consideration of professional practice and ethics.
- Extensive evidence of version-control showing professional development practice.

Environment and Tools

You are required to use the following tools:

- PyCharm: as your integrated development environment.
- Python 3.11+: as the standard python library.
- Draw.io (or equivalent tool): To create diagrams.
- Microsoft Word (or equivalent tool): To author your report and export as a PDF.
- Adobe PDF Reader: To view your final report which is saved as a PDF file.
- Git Tools and GitHub: for version control.

Additionally, the following libraries/modules may be imported and utilised:

- enum: to add enumerations
- math: for mathematical functions
- os: to retrieve or check file paths
- pytest: to test your implementation
- random: to generate random numbers
- sklearn: for machine learning related algorithms and constructs
- tkinter: to create a graphical and interactive interface
- typing: to add type checking
- textblob: for processing textual data and performing natural-language processing.
- unittest: to construct and run tests
- any standard python data structures

No other python libraries or modules should be used without written permission of the module convenor.

Submission

The assessment must be completed individually. You must not share, in part or whole, your assessment with another party other than the module convenor and for the purpose of submission to the university. You must ensure that the University's academic misconduct guidelines are followed in their entirety.

You should use the assessment submission link on the module's Moodle page to submit the following files:

- A **PDF** file for your report. This should not be included in the zip file but instead submitted as a separate file. Failure to do so may result in zero being awarded.
- A **Zip** file of your software artefact. This should contain your software solution and any relevant files to open and execute your solution.

You should ensure that you make a timely submission by the deadline stated at the start of this assessment brief.

Assessment Criteria

Your assessment will be graded according to the following criteria:

Grading criteria	Functionality	Documentation	Professional Practice
Mark band			
80-100 Pass (1st)	Displays exceptional skills with comprehensive implementation and design considerations including strong considerations and implementation for artificial intelligence.	Exceptional analysis and evaluation provided, offering deep insights into the software artefact's strengths and weaknesses, supported by evidence and critical reflection. Report is exceptionally well-structured with use of referencing and citations, with thorough and accurate citation of relevant sources to support arguments and analysis.	Demonstrates exceptional adherence to industry and ethical practice with strong usage of version control with a well-structured repository, regular commits, detailed commit messages, and evidence of branching and merging strategies.
70-79 Pass (1st)	Exhibits advanced skills with comprehensive AI and object-oriented development, and in-depth design considerations. Implements significant artificial intelligence features.	Advanced communication in explaining complex software development concepts. Well-structured report with appropriate citations and captions where relevant.	Advanced adherence to code conventions, ethics, and relevant industry practice with usage of version control with a well-structured repository, regular commits, detailed commit messages, and evidence of branching/merging strategies.
60-69 Pass (2.1)	Good, consistent knowledge and understanding of the material, main concepts, key theories, and practice at this level. Exhibits advanced skills in AI and object-oriented application development.	Strong communication in explaining complex programming and problem-solving concepts.	Competent adherence to ethics, code conventions and other relevant industry practice with usage of version control with regular commits and meaningful commit messages that provide clear context for changes.
50-59 Pass (2.2)	Sound, routine knowledge and understanding of the material, main concepts and key theories. Some flaws may be evident. Shows improved skills with refined implementation.	Improved communication skills seen in summarising design and implementation.	Improved adherence to code conventions and other relevant industry practice. Some ethical considerations. Usage of version control with regular commits. Commit messages provide some context but may lack detail or consistency.
40-49 Pass (3rd) (*Threshold*)	Demonstrates basic skills in developing a functional application	Basic communication evident in summary of implemented solution.	Basic adherence to code conventions and other relevant industry practice with evidence of version control usage. Commits are sporadic and lack meaningful commit messages.
30-39 Fail	Displays limited proficiency, with significant deficiencies in the programming.	Demonstrates inadequate communication skills, struggling to articulate ideas and concepts.	Inadequate adherence to code conventions and other relevant industry practice. Minimal evidence of version control.

Learning Outcomes

This assessment will enable students to demonstrate the following learning outcomes as stated in the module outline:

[Module Learning Outcome 2]

- Analyse and assess the societal impact of different AI applications, recognising their potential benefits and risks on individuals and communities, and consider their ethical implications.

How is this learning outcome addressed?

The assessment brief meets this learning outcome by requiring students to evaluate their solution. Students must consider the professional and ethical implications of their solution.

[Module Learning Outcome 3]

- Analyse and evaluate problem-solving techniques and algorithms used in AI, ethically applying them to diverse AI scenarios and identifying their strengths and limitations.

How is this learning outcome addressed?

The assessment brief aligns with this learning outcome by tasking students with creating, designing, and thoroughly testing a software solution to meet specific a AI scenario. It emphasises the utilisation of algorithms, advanced problem-solving skills, analytical thinking, and consideration of ethical and professional practices throughout the development process.

[Module Learning Outcome 4]

- Apply problem-solving techniques and algorithms to design, develop, and optimise AI systems, demonstrating proficiency in addressing AI-related challenges.

How is this learning outcome addressed?

This assessment addresses the learning outcome by requiring students to employ various algorithms and design patterns in their software solution. Students are

required to plan, develop, and deliver a suitable solution for the given problem scenario.

[Module Learning Outcome 5]

- Acquire practical AI programming skills and confidently apply them to create AI systems, leveraging popular AI frameworks, libraries, and tools to develop functional applications.

How is this learning outcome addressed?

The assessment brief supports this learning outcome by specifying the use of an Integrated Development Environment (IDE), an object-oriented programming language (Python), and AI algorithms and libraries for software development. It emphasises adherence to best practices in code design, readability, maintainability, and AI design reflecting the principles of AI development through the implementation.

Regulations, Policies, and Guidelines

Guidance for online submissions
https://www.stmarys.ac.uk/policies/online-submissions.aspx

Academic Misconduct
Any submission must be students' own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. Please find a link to the academic misconduct policy below:
https://www.stmarys.ac.uk/policies/academic-regulations.aspx

Ethics Policy
The work being carried out by students must be in compliance with the Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then students will need ethical approval prior to the start of the project. Please find a link to the ethics policy below:
https://www.stmarys.ac.uk/research/students/ethical-review-process.aspx

Extenuating Circumstances
The University's Extenuating Circumstances procedure helps students facing challenges in assessment submission. To request an extension or deferment, submit an EC application with evidence. Approved cases will not incur academic penalties. For longer-term issues, contact Student Services. Please find a link to the EC policy below:
https://www.stmarys.ac.uk/policies/extenuating-circumstances.aspx