

Contents

Introduction.....	2
MSci Project General Information	3
My project specification: Artificial Intelligence Planning.....	3
My project abstract	5
My project outcome	6
Grade Grid References	6
Interim Submission – Report (10%).....	8
Interim Submission – Programs and Technical Achievement (10%)	10
Interim Submission – Viva (5%)	10
Final Submission – Report (30%)	11
Final Submission – End Product (30%).....	15
Professionalism (5%)	18
Project Demonstration (5%)	19

Introduction

What is a project

An individual project is a piece of individual work done under the guidance of an academic supervisor. It should be about 300 hours of work.

Different projects may stress **theoretical** aspects of a problem, or **practical** (programming/implementation) aspects, or may be essentially of a **survey** type.

- You will have to submit a project plan, programs that you have written, an interim report for the December Review meeting, draft and final project reports.
- You will have to present your project and demonstrate the software that you have developed to staff and fellow students.
- You will have to keep a work diary up to date, attend meetings for which you have prepared in advance, and store all of your work on the GitLab account the department will provide to you.

Why do a project?

The Department **requires** all single honours Computer Science students to do a full unit individual project. Students on joint honours courses are also encouraged to do a full unit project, but other course commitments may mean that a half unit project is more appropriate.

A project is valuable to you because it is your opportunity to demonstrate your ability to work individually. *Success in your project will provide evidence of your skills to any future employer and students often use their project supervisors as referees after completing their degree.*

Your project may be the most enjoyable part of your undergraduate studies. You get to choose a topic of personal interest and to study it in depth.

How do you complete a project

You will be assigned to a supervisor. Your supervisor takes the role of your line manager in a company. **They are certainly not there to teach you the material for your project. The project eventually is your own, and you have to take individual responsibility for its success**

You will have a number of brief meetings with your supervisor and at these meetings they will give you feedback on the work you have done, suggest new targets and help with any particular questions that you have.

You are required to organise your own time, *to keep a work diary*, deliver reports and programs for assessment and comment, and to update your GitLab regularly.

Use your supervisor to good effect. Hand in regular reports, show them your code and keep them up to date on your progress towards your final deliverables. They can help assess your writing style, your coding style and your understanding of the relevant theory.

The success of your project relies on steady hard work throughout the time you have available.

MSci Project General Information

Aims

The aim of the individual project is to give students the opportunity to complete a substantial piece of work. This involves organising their own time, deadlines and deliverables, and delivering a completed piece of work in a professional manner.

Objectives

The following objectives will be achieved by completing a individual project.

The student will be able to:

- work independently on a significant piece of work, organising deadlines and deliverables;
- learn new skills and theory from diverse information sources;
- make technical decisions after consideration of appropriate evidence and act on those decisions;
- present and discuss a technical subject;
- compose and complete a technical report;
- work steadily under guidance for the duration of a project;
- understand what is required of a computing professional.

Assessment Criteria

Whilst there are several different assessments made there are only three marked components to an individual project: the project plan, the interim assessment, and the final assessment.

My project specification: Artificial Intelligence Planning

Aims: The goal of this project is to either effectively model an application as a planning problem or to improve existing algorithms for automatically finding plans.

Background:

Broadly speaking, planning is the art of thinking before acting. Automated planning is a crucial and well-established field of artificial intelligence and is concerned with the automatic synthesis of a course of action to achieve a set of goals based on a given description of the world. A planner is an instance of a general problem solver: it accepts an abstract description of a problem and returns a solution for it. No knowledge of the domain is provided to the planning, which must be able to adapt to any description of the problem. This description consists of a model of how the world functions, the initial state of the environment and a set of objectives to achieve. The solution of a planning problem, a plan, is a sequence of actions that accomplish the goals when executed. The choice of different ways to describe these fundamental elements, world model, initial state and objectives, has determined the developments of a number of distinct planning paradigms. They range from those addressing fully observable, deterministic, static and discrete environments to those dealing with partially observable or stochastic

environments. In all cases, the derivation of a course of action from a model is computational intractable. Hence, the central challenge in planning is to achieve both generality and scalability.

Many problems can be formalised as planning tasks. Prototypical examples are single-player games ("puzzles") such as the Rubik's Cube or the Freecell card game. Real-world applications of automated planning are the generation of plans for robotic and virtual agents, logistic problems, airport ground traffic control, testing protocols for logical flaws and printer control.

Within the area of AI planning, a wide variety of projects are possible, ranging from highly theoretical to practical and implementation-oriented. Projects can focus either on modelling a problem as a planning task (e.g. search and rescue, urban traffic control, internet of things) and then solving it with existing state-of-the-art planners or on devising new algorithms for efficiently searching for plans (e.g. generating new heuristics for speeding up the search for plans).

Early Deliverables

1. Report: The modelling languages for automated planning.
2. Code: Examples of models that describe specific applications and implementations or proofs of validity.
3. Report: State-of-the-art Fast-Downward planners. Specifically this must include manually worked examples of their operation.
4. Report: Identification of a specific scenario/application to model as a planning task. In depth description of why planning is well suited to model such a scenario/application.
5. Code: Working prototypes of several alternative models of your chosen application and their validation.

Final Deliverables

1. Report/Code: A planning model in PDDL for the chosen application and an in depth report that describes and motivates the model.
2. Report/Code: Experimental evidence that the problem under consideration can be solved efficiently via planning.
3. Code of the algorithms implemented, experimental evidence that the algorithms work efficiently (both for optimal and sub-optimal search) and in-depth report on the theoretical approach behind the algorithms implemented.
4. Code: The algorithms must include:
 - An implementation of at least 2 domain-specific heuristics within the FastDownward planner. By domain, we mean the application.
 - A domain-specific heuristic search algorithm where you hard-code the domain and the following search algorithms:
 - A*
 - Weighted A*

- The same domain-specific heuristics

Reading

- Artificial Intelligence: A Modern Approach (Third edition) by Stuart Russell and Peter Norvig. Chapters 10 and 11.
- A Concise Introduction to Models and Methods for Automated Planning by H. Geffner and B. Bonet. Morgan and Claypool, 2013
- Automated Planning - Theory and Practice by Malik Ghallab, Dana S. Nau, Paolo Traverso. Elsevier, 2004
- Useful Website: [Fast Downward Planning](#)
- Useful Website: [Get started with PDDL](#)
- Useful Website: [Modelling in PDDL \(1\)](#)
- Useful Website: [Modelling in PDDL \(2\)](#)
- Useful Website: [Modelling in PDDL \(3\)](#)
- Useful Website: [Modelling in PDDL \(4\)](#)
- Useful Website: [PDDL Tooling Episode 1](#)
- Useful Website: [PDDL Tooling Episode 2](#)
- Useful Website: [CAPS 2019: Tutorial on PDDL](#)

Prerequisites: Strong interest towards Artificial Intelligence. Good programming skills, including the ability to modify C++ code. Positive attitude towards learning new modelling languages, algorithms and techniques.

My project abstract

Logistical operations in motorsport contribute significantly to the industry's environmental footprint. These operations are notoriously complex and resource-intensive, involving the global transportation of vehicles, equipment, and personnel under tight schedules. Motorsport organisations are working to reduce their environmental impact; Formula 1, for instance, set a net-zero target by 2030, with logistics contributing to 49% of F1's 2022 **carbon footprint** [7].

Artificial intelligence is transforming sustainability in the automotive industry. AI applications in logistics and automotive production contribute significantly to reducing greenhouse gas emissions by **optimising routes and fuel consumption** [8]. Automated planning, an AI technique, optimises complex logistical challenges and has been successfully applied in demanding fields [9]. This project focuses on minimising the environmental impact of motorsport logistics by developing and testing an **automated planning model** using PDDL.

The **inspiration** for this project stems from participating in the Google x Formula E sustainability hackathon, exploring how AI-driven strategies could positively impact motorsport logistics [10], [11]. Another source of inspiration comes from experience optimising aviation operations using AI in the SaaS industry.

Percentage Description

50%-69%	Adequate Description and motivation of the project
70%-85%	Describes and motivates the project. Gives well thought out individual project goals
86%-100%	Clearly describes and motivates the project. The abstract, especially project goals, is convincing and professional

Timeline: 40 marks**Percentage Description**

<40%	Milestones missing, poorly thought through or just copied from project list
40%-49%	Milestones inadequate to give structure to an MSci project
50%-69%	Milestones are adequate or good, each with dates but some have insufficient motivation
70%-85%	Milestones are very good, each with dates. they are well motivated and clearly structure the work required
86%-100%	Good list of well explained milestones

Bibliography: 15 marks**Percentage Description**

<40%	Bibliography missing, unusable or just includes web pages
40%-49%	Bibliography lists too few sources or in some other way is inadequate for an MSci project
50%-69%	Clear use of more than one source. Not just web sites. Nice evidence of good research
70%-85%	Clear use of more than one source. Not just web sites. Strong evidence of good appropriate research
86%-100%	Bibliography items are very good. Each has a short relevant discussion of its value to the project

Risk Assessment: 15 marks**Percentage Description**

<40%	just general risks such as "may fall behind schedule"
------	---

Percentage Description

40%-49%	Risks assessed but thought process in assessment fails to pass at MSci level
50%-69%	Risks associated with deliverables and show clear thought. Ideally each (or some) risks are provided with mitigations that seem reasonable
70%-85%	Risks associated with deliverables and show clear thought. Good mitigation provided. Clear evidence of careful thought
86%-100%	A good case is made for understanding how the project might fail to proceed and what should be done about it. Some risks have likelihoods and importance

Interim Submission – Report (10%)

Quality of Writing: 20 marks

Percentage Description

<40%	Poor writing or structure or too little material
40%-49%	Writing or structure weak or too little material to pass at MSci level
50%-69%	Well structured and good use of English
70%-85%	Very well structured. Well written
86%-100%	Organised and collated towards a final project report

Background Theory: 35 marks

Percentage Description

<40%	Very little theory or poor evidence of understanding
40%-49%	Evidence of understanding theory below the level expected for an MSci
50%-69%	Evidence of understanding of background theory, ideally demonstrated by explanation of appropriate proof of concept programs
70%-85%	Good evidence of understanding of background theory. Nice proof of concept programs. Evidence of a structured process
86%-100%	Clear evidence of independent thought using background theory imaginatively

Project Diary: 10 marks

Percentage Description

<40%	No diary or just a list of achievements
------	---

Percentage Description

40%-49%	Diary or just a list of achievements weak
50%-69%	Diary of achievements well related to original plan
70%-85%	Diary of achievements well related to original plan, showing clear project structure
86%-100%	Reflective diary showing strong evidence of analysis of process

Bibliography: 10 marks**Percentage Description**

<40%	No bibliography and poor references to the literature
40%-49%	Some bibliography and references to the literature, but inadequate
50%-69%	Interesting and appropriate bibliography or many appropriate well understood references
70%-85%	Interesting and appropriate bibliography. Use in report shows a level of competence in the area
86%-100%	Excellent evidence of background reading and appropriate use of references in the report

Software Engineering: 15 marks**Percentage Description**

<40%	Poor evidence of use of SE tools and methodology
40%-49%	Weak evidence of use of SE tools and methodology
50%-69%	Clear evidence of an engineering approach to developing software
70%-85%	Very clear evidence of an engineering approach to developing software, possibly including a good test strategy, requirement analysis, and UML diagrams
86%-100%	Excellent SE. Clear evidence of all appropriate processes including the use of a version control system and a good test strategy

Demo Video: 10 marks**Percentage Description**

< 40%	No video or poor explanation.
40%-69%	Good video provides some basic information of the project.

Percentage Description

70%-100% Excellent Video. Clear demonstration of expected deliverables interim submission.

Interim Submission – Programs and Technical Achievement (10%)

Level	Description	Value
<i>Inadequate</i>	Only poor quality code or none submitted. No new programming or algorithmic concepts/algorithms/data structures/use of libraries beyond second year level.	Fewer than 25 marks
<i>Poor</i>	Some working code without good documentation or poorly written. Weak evidence of engaging with the programming challenges of the project.	26-39 marks
<i>Basic</i>	Working code, well written or adequately documented. Clear sense of purpose in programs.	40-59 marks
<i>Good</i>	As <i>Basic</i> . Interesting or complex algorithms coded, or perhaps the use of complex library. Clear evidence of a design process.	60-69 marks
<i>Very Good</i>	As <i>Good</i> but with clear focus on covering a wide range of topics necessary to complete the final programs. Also final program initial design begun.	70-85 marks
<i>Excellent</i>	As <i>Very Good</i> but also completing advanced targets from the project specification or other significant extensions outside of the original project specification. A nearly complete good quality project.	More than 85 marks

Interim Submission – Viva (5%)

Level	Description	Value
<i>Inadequate</i>	Does not understand the aims of the project let alone anything done towards achieving them.	Fewer than 25 marks
<i>Poor</i>	Does not understand basic theories relating to any part of the project and cannot defend any of it.	26-39 marks
<i>Basic</i>	Understood much of what they have done. May be very hesitant on background theory.	40-59 marks
<i>Good</i>	As <i>Basic</i> . Could not necessarily defend all decisions and maybe struggled if the conversation went beyond the scope of the minimum requirements for the project.	60-69 marks
<i>Very Good</i>	As <i>Good</i> . The student clearly understood and defended nearly all aspects of the project and its background. Clear evidence of commitment to excellent performance on the project.	70-85 marks
<i>Excellent</i>	As <i>Very Good</i> . The student clearly knew more than experts in the department about (some aspects of) the project and its context.	More than 85 marks

Final Submission – Report (30%)

A final project report is approximately 15,000 words and must include a word count. *It is acceptable to have other material in appendixes.*

The section on **professional issues** should be about **1,000 words**.

Technical Accuracy: 10 marks

Percentage Description

0-39	Inaccurate content.
40-49	Superficial technical description of content.
50-59	Technical content moderately accurate, with some inaccuracies.
60-69	Overall, the technical content is accurate.
70-85	As Above.
86-100	Excellent and accurate technical content, well described.

Aims and Objectives: 5 marks

Percentage Description

0-39	Aims and objectives not met.
40-49	Aims and objectives poorly met.
50-59	Aims and objectives partially met.
60-69	Aims and objectives mostly met.
70-85	As Above.
86-100	Aims and objectives fully met.

Project Challenge: 5 marks

Percentage Description

0-39	Not challenging. Very easy.
40-49	Challenging in a limited way. Fairly easy.
50-59	Moderately challenging.
60-69	Fairly challenging.
70-85	As Above.

Percentage Description

86-100 Very challenging.

Clarity of Writing: 5 marks**Percentage Description**

0-39 Inappropriate use of English.
40-49 Poor literacy style and deficiencies in presentation.
50-59 Generally clear literacy style and mainly appropriate presentation.
60-69 Correct, clear English and writing style.
70-85 As Above.
86-100 Lucid style of expression in English.

Depth of Content: 10 marks**Percentage Description**

0-39 No depth of content.
40-49 Limited exploration of content.
50-59 Some content covered appropriately.
60-69 Good in-depth coverage of content.
70-85 As Above.
86-100 Thorough in-depth coverage of content.

Evaluation and Analysis: 10 marks**Percentage Description**

0-39 No critical evaluation. Weak and unacceptable analysis.
40-49 Superficial critical evaluation. Limited analysis.
50-59 Appropriate but limited evaluation and analysis.
60-69 Evidence of ability to evaluate results. Competent analysis.
70-85 As Above.
86-100 Clear evaluation and conclusions. High level of critical analysis.

Report Layout: 5 marks**Percentage Description**

0-39	Unacceptable layout in terms of organisation and structure.
40-49	Poor layout in terms of organisation and structure.
50-59	Generally good layout.
60-69	Consistent layout and clear overall organisation and structure.
70-85	As Above.
86-100	Excellent layout and very clear organisation and structure.

Professional Issues: 10 marks**Percentage Description**

0-39	No real attempt to describe any professional issues.
40-49	Professional issues content poorly thought out and/or not related to the project.
50-59	Professional issues discussed that are shown to be relevant to the project.
60-69	Discussion of professional issues. Clearly related to project material.
70-85	Good, clear discussion of professional issues. Well written and related to project material.
86-100	Thoughtful discussion of professional issues and how they have affected the project process.

Use of References: 5 marks**Percentage Description**

0-39	No bibliography.
40-49	Bibliography present but poorly formatted.
50-59	Well-written bibliography. Some citations.
60-69	Well-formatted bibliography. Adequate citations at correct point.
70-85	As Above.
86-100	Correct reference to sources and inclusion of full bibliography.

Sufficiency of Content: 10 marks**Percentage Description**

0-39	Inadequate or poor coverage of relevant content.
40-49	Incomplete coverage of content.
50-59	Sufficient coverage of relevant content.
60-69	Good coverage of relevant content.
70-85	As Above.
86-100	Excellent and exhaustive coverage of relevant content.

Independent Thought: 5 marks**Percentage Description**

0-39	No evidence of independence of thought.
40-49	Limited evidence of independence of thought.
50-59	Appropriate evidence of independence of thought. Limited novelty.
60-69	Evidence of independence of thought. Some novelty.
70-85	As Above.
86-100	Clear evidence of independence of thought. High level of novelty.

Understanding: 15 marks**Percentage Description**

0-39	No evidence of understanding of the project topic.
40-49	Little evidence of understanding of the project topic.
50-59	Evidence of understanding with clear explanations.
60-69	Good knowledgeable account of project topic.
70-85	As Above.
86-100	Excellent understanding and insight.

Demo Video: 5 marks

Percentage Description

< 40% No video or poor explanation.

40%-69% Good video provides some basic information of the project.

70%-100% Excellent Video. Clear demonstration of expected deliverables interim submission.

Final Submission – End Product (30%)

Marks awarded by the project supervisor and second marker for the *quality* of the end product and the candidate's total technical achievement.

- Does it work? Is it stable?
- Is the software usable? Is the interface appropriate for the application (a compiler might require more technical skills to run than an e-commerce site)?
- Does the code and system structure follow the design?
- How complete is the functionality with respect to the requirements?
- Is the coding clean and well documented?
- Does the GitLab repository reflect the use of good software engineering principles including appropriate use of branches/tags?
- Does the product require techniques, tools, theory or concepts which are clearly above second year level?
- Does the product use complex new technologies/platforms or require the study of a significant body of literature from disparate sources to complete?

Code Quality: 10 marks

Percentage Description

0-39 Code may not even be readable

40-49 Code was submitted, but code organisation or structure insufficient to pass at the MSci level

50-69 Code meets a clean standard - possibly Checkstyle. Code nicely organised

70-85 Lovely code that clearly shows the maturity expected at MSci level. Well organised and clean.

86-100 Beautiful code, nicely commented, with clear helpful structure

Complexity: 20 marks

Percentage Description

0-39	Code or results are very simple or do not represent what would normally be expected for this project
40-49	Code or results are simple or do not represent what would normally be expected to pass at MSci level
50-69	The project as delivered required learning of a significant amount of material, or the use of complex technologies
70-85	The project as delivered demonstrated learning of a significant amount of material, and the use of complex technologies
86-100	The project involved a deep understanding of several new areas of work and this is demonstrated in a successful implementation or clear and professional presentation of theory

Functionality: 10 marks**Percentage Description**

0-39	Code may not even work as expected or not possible to see code running using supplied instructions, or in a submitted video.
40-49	Code seems to work as expected but the supplied instructions or video fail to achieve MSci pass level.
50-69	Code has all functionality as specified in the original project specification
70-85	Code exceeds the basic functionality as specified in the original project specification. Nice videos submitted that show all aspects of the code
86-100	Functionality significantly exceeds original specification. Clear instructions, both for usage and installation. Good submitted video.

Stability: 10 marks**Percentage Description**

0-39	Code has no documentation or overall structure is very weak
40-49	Code has documentation but the readability or overall structure below the level expected for an MSci pass
50-69	Code has good documentation and includes clear use of regression testing
70-85	Code has very good documentation. Clear and explicit use of Software Engineering techniques like regression testing, or good VCS history

Percentage Description

86-100	Excellent SE. Clear evidence of all appropriate processes including the use of a revision control system and a good test strategy
--------	---

Structure and Design: 15 marks**Percentage Description**

0-39	Code may not even be broken up according to functionality
40-49	Design process not clearly apparent. Code structure development is clear, but has failings so does not achieve MSci pass level.
50-69	Clear evidence of a design process. UML and design patterns used.
70-85	Clear evidence of a design process. UML and design patterns used. May use appropriate data structures or design patterns
86-100	Design process shows refactoring. Evidence of careful use of appropriate data structures.

Technical Expertise: 30 marks**Percentage Description**

0-39	Project as delivered requires no final year expertise
40-49	Project as delivered requires does not show the level of expertise expected at MSci level
50-69	Project requires techniques, tools, theory or concepts which are clearly above third year level
70-85	Project shows good use of techniques, tools, theory or concepts which are clearly above third year level
86-100	Some tools, theory or techniques employed show evidence of mastery at a high level

Usability: 5 marks**Percentage Description**

0-39	User interface unclear missing, or inconsistent.
40-49	User interface unclear missing, inconsistent. or too simple to achieve a pass at MSci level

Percentage Description

50-69	User interface effective and appropriate. Code works well in its designed environment
70-85	User interface effective and appropriate. Code works cleanly and is clearly ready to be delivered and fit for purpose
86-100	Clear evidence of user interface design and thought about the user experience. Some attention on accessibility, or other enhancements

Professionalism (5%)

Marks awarded by the supervisor for the ability of the student to plan and organise their project as a professional. This score reflects the students ability to recognise and follow appropriate professional behaviour.

If there is no evidence of frequent, reflective updates to your work diary then a mark of zero will be awarded for your professionalism.

You are evaluated on:

- the ability to attend supervisory meetings;
- the ability to alter their project plan and identify priorities as they arise;
- the ability to keep an organised project diary;
- the ability to maintain a consistently high level of effort throughout the project;
- an assessment of whether the student can work independently or requires constant supervisory intervention;
- communicate well, understanding relevant professional issues, and maintain a good professional working attitude towards the project

Level	Description	Value
<i>Inadequate</i>	Student did not undertake any planning, and little to no effort was spent on the project.	Fewer than 25 marks
<i>Poor</i>	The student showed little if any attempt at organisation. Despite prompting by supervisor the student's effort was poor.	26-39 marks
<i>Basic</i>	The student showed some organisational skills, and the level of effort in the project was generally adequate. Marks in the lower range are awarded if the student required regular prompting.	40-59 marks
<i>Good</i>	The project was well organised and a significant amount of effort was put into the project by the student at during the project.	60-69 marks

Level	Description	Value
<i>Very Good</i>	The project was well organised. The behaviour of the student achieved a high standard of professionalism.	70-85 marks
<i>Excellent</i>	The project was very clearly organised, taking into account wider professional issues, and the student behaved throughout in a professional manner.	More than 85 marks

Project Demonstration (5%)

Marks awarded by the second marker for the ability of the student to defend the work.

The purpose of the project demonstration is to explore whether the student can:

- explain the aims and objectives clearly;
- demonstrate the working code;
- explain the background/relevance/importance of the project and set it in the wider context;
- briefly explain the theory underpinning the individual parts of the project (for example how algorithms work or which architectural options existed including their benefits/drawbacks);
- communicate well;
- support their work with a clear simple A3 poster.

Level	Description	Value
<i>Inadequate</i>	Does not understand the aims of the project let alone anything done towards achieving them.	Fewer than 25 marks
<i>Poor</i>	Does not understand basic theories relating to any part of the project and cannot defend any of it.	26-39 marks
<i>Basic</i>	Understood much of what they have done. The product works well enough. May be very hesitant on background theory.	40-59 marks
<i>Good</i>	Could not necessarily defend all decisions and maybe struggled if the conversation went outside the scope of the project. Simple Poster on display.	60-69 marks
<i>Very Good</i>	The student clearly understood and defended nearly all aspects of the project and its background. Clear evidence of potential.	70-85 marks
<i>Excellent</i>	The student clearly knew more than experts in the department about (some aspects of) the project and its context. Project demonstration flawless. Good Poster.	More than 85 marks