Introduction:
Problem Environments & Intelligent Agents

CS3243: Introduction to Artificial Intelligence – Lecture 1

NUS | Computing
National University of Singapore

Contents

Administrative Matters

- Teaching Staff
- Topics & Schedule
- Assessments
- Support & Resources

Introduction to AI

- What is AI?
- Intelligent Agents
- Problem Environments
- Taxonomy of Agents

1 Administrative Matters

Teaching Staff

- Lecturer
  - Daren Ler : dler@comp.nus.edu.sg
  - https://www.comp.nus.edu.sg/cs/people/dler/

- Tutors
  - Long Tianqi : e0969329@u.nus.edu
  - Tam Keng Hong : e0257179@u.nus.edu

Topics

- Introduction: Problem Environments and Intelligence Agents
- Uninformed Search: Problem-solving Agents and Path Planning
- Informed Search: Incorporating Domain Knowledge
- Local Search: Efficient Goal (Versus Path) Search
- Constraint Satisfaction Problems: Generalising Goal Search
- Adversarial Search: Playing Games
- Logical Agents: Knowledge Representation
- Bayesian Networks: Representations within Uncertainty

Weekly Schedule

- Lectures
  - Wednesday, 1000-1200 hrs
  - Release of Lecture Slides: Tuesdays
  - Release of Lecture Video: Wednesdays (later in the afternoon)
- Diagnostic Quizzes (Optional, but Recommended)
  - Release: Wednesdays (after the lecture)
  - Deadline: None (attempt independently – review with peers/tutor)
- Tutorials
  - Begin Week 3
  - Release: Wednesdays (after lectures - from Lecture 2 onwards)
  - Tutorial Assignment Pre-Tutorial Deadline: Sunday (week before Tutorial Session)
  - Tutorial Assignment Post-Tutorial Deadline: Friday (same week of Tutorial Session)

About the Course & Lectures

Some positive comments

- Sem 2310: The content coverage of the course was well thought out and the order in which topics are taught was very logical and well structured. The assignments were also well designed to have a good coverage over practicing the newly learnt content.
- Sem 2320: He is very supportive and encourages students to ask questions to reinforce concepts in lectures, projects, etc.
- Sem 2320: He is very methodological in his teaching and assessment structure. I like how he explicitly tell us the structure of the assessment and his grading is quite transparent. This helps me systematically and confidently allocate my resources/time while studying for this module, making it enjoyable.

Some negative comments

- Sem 2310: No machine learning is involved. It would be interesting if we cover a little bit of learning and combine them with what is taught in this course.
- Sem 2320: I suggested more interactive lecture styles where we have time to practice thinking through questions in class and he has added them which I really enjoy ... though I can understand how this is difficult when many students do not want to participate in lecture ... I think getting students to answer questions during lectures might help identify areas where students have misunderstandings.
- Sem 2320: Stick to the initial way of teaching in the first half of the semester (i.e., without the random questions during lecture).

About the Diagnostic Quizzes & Tutorial Assignments

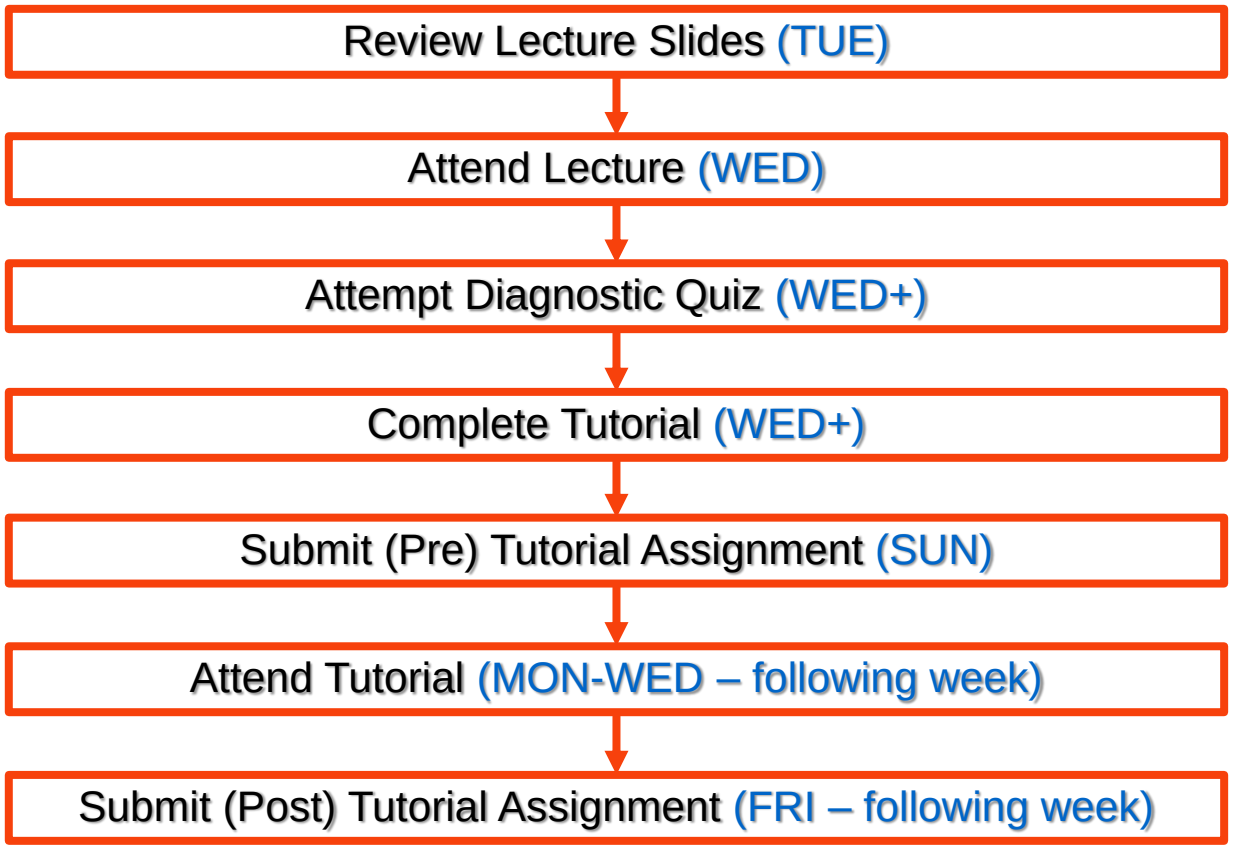
Some positive comments

- Sem 2220: We are given many resources like post lecture diagnostic quizzes, and I like that its optional.
- Sem 2220: I really liked how the DQs were provided but were not graded, as there were some higher-level thinking questions (which is good for learning), without stressing us students too much.
- Comments from TAs: ... DQs are important and should be mandatory so that students enter the tutorials with the correct pre-requisite knowledge.

Some negative comments

- Sem 2210: The course had the typical micromanager-y nature of assignments that CS modules tend to have, which I think lecturers should not bother, especially for a 3k module. If a student can't self-manage how much they learn from a module at a 3k level, they kind of deserve to fail.
- Sem 2220: ... tutorial homework submissions should be placed after tutorials as the tutors help to clarify some misconceptions on the topic (but it was too late to modify tutorial answers).
- Comments from TAs: ... tutorial homework should be submitted before tutorials so coverage can be in-depth.
- Sem 2320: Too many submissions for tutorials. Maybe 1 a week would be good.

Expected Learning Flow



- Lecture slides released the day before the lecture (TUE)
- Lectures recordings available for reference purposes – released after lecture (WED afternoon)
- Refer to the reference textbook for additional extended content
- Discuss on Telegram
- Discuss on Canvas > Discussions

Project & Examination Schedule

- Projects
  - Project 1.1: released Week 3 (MON) ; due Week 5 (SUN)
  - Project 1.2: released Week 3 (MON) ; due Week 6 (SUN)
  - Project 2.1: released Week 6 (MON) ; due Week 8 (SUN)
  - Project 2.2: released Week 6 (MON) ; due Week 9 (SUN)
  - Project 3: released Week 9 (MON) ; due Week 12 (SUN)
- Midterm Examination
  - 2 October, 1005-1135 hrs (Week 7 (WED) Lecture Session)
- Final Examination
  - 3 December, 0900-1100 hrs (check details on EduRec)

About the Projects

Some positive comments

- Sem 2220: The projects were really challenging ..., but in retrospect, they were well designed with each project nicely leading to the next.
- Sem 2310: The project component was very fun and helps with learning.
- Sem 2320: The projects. Although I'm dying, it's fun.
- Sem 2320: The Projects challenged me in a good way, and I was able to get a lot out from this course.
- Sem 2320: The course was very interesting. The projects allow us to really test the algorithms we studied.

Some negative comments

- Sem 2310: The project private test cases are hidden with no sense of directions on how to improve. The hints should be given earlier in the project rather than at the end.
- Sem 2320: The projects were extremely difficult for a non soc student that only had background in cs2040... steep learning curve and disadvantaged, but the support from the team made up for it. Overall enjoyable module!
- Sem 2320: The projects are too difficult (not due to the implementation but due to the efficiency)... Even if efficiency must be mentioned, I feel that it is sometimes a bit too complicated (especially because I never had these kind of concerns before). So, knowing which data structures we should use without any clue is definitely hard. And testing all of them is time consuming.

About the Examinations

Some positive comments

- Sem 2220: Examinations thus far were set as a reasonable difficulty and covered content discussed in lectures and tutorials.

Some negative comments

- Sem 2220: Exams (midterms) might benefit with more time allocated for students to attempt the questions, as I found myself struggling to ensure that I could give enough thought to every question within the given time period.
- Sem 2310: in exams ... modelling ones (questions), can be difficult in the sense that you either get it on the spot or you can never think of it in the 2-hour period.
- Sem 2310: The exams sometimes feel like a hit-and-miss ... questions often feel totally different from the tutorials.

Assessments & Course Weights

- Course Participation (11)
  - Total 5%
  - Tutorial Attendance
- Tutorial Assignments (9)
  - Total 5%
  - Each worth 1% (best 5 taken)
- Midterm Examination (1)
  - Total 20%
  - Closed Book + Cheat Sheet (1 x A4 Sheet)
  - In-person + Written
- Python Projects (5)
  - Total 30%
    - Projects 1.1 + 1.2: 10%
    - Projects 2.1 + 2.2: 10%
    - Project 3: 10%
  - Individual
  - Bonus marks available (covering projects)
- Final Examination (1)
  - Total 40%
  - Closed Book + Cheat Sheet (1 x A4 Sheet)
  - In-person + Written

Plagiarism & Copyright

- Plagiarism
  - Reported for disciplinary action
  - Simple rule for projects: discussions are ok, but DO NOT SHARE OR COPY CODE!
- Copyrights

NUS National University of Singapore

NUS Course Materials: Ethical Behaviour and Respecting Copyright

All course participants (including permitted guest students) who have access to the course materials on LumiNUS or any approved platforms by NUS for delivery of NUS modules are not allowed to re-distribute the contents in any forms to third parties without the explicit consent from the module instructors or authorized NUS officials

NUS National University of Singapore

Examples of Disallowed Things

No Posting on any websites (except for the materials explicitly allowed by your lecturer in the respective module)

No selling of material

No sharing of questions/answers which could lead to cheating/plagiarism

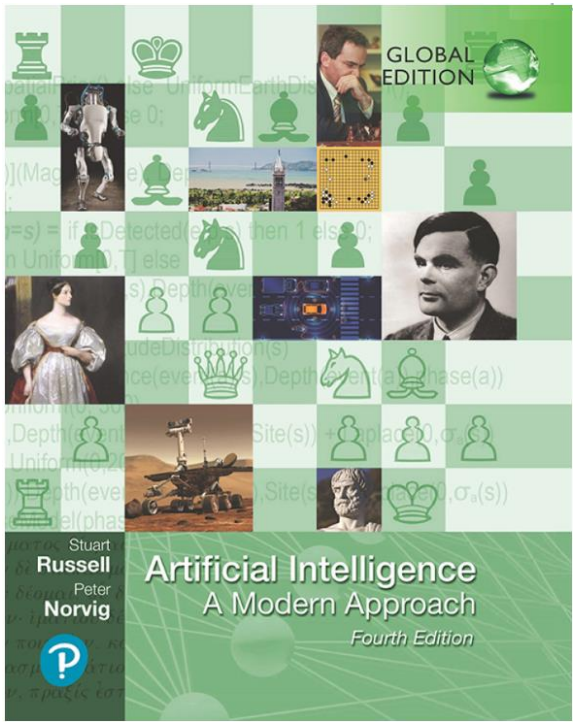
Lecture Protocol

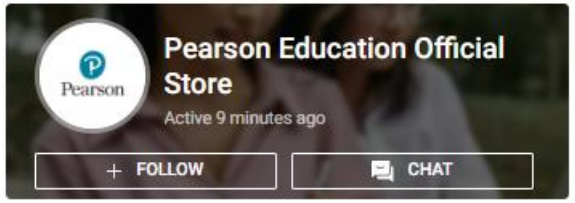
- Post questions anytime
  - Ask questions on Archipelago
- Answers given at selected times
  - After the break
  - Towards the end of the lecture
- Archipelago
  - Use the voting board to post/upvote questions
- Lecture recordings
  - Canvas > CS3243 > Videos/Panopto

Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/67243073190

Resources & Textbook

- All course material will be on Canvas (Files)
  - https://canvas.nus.edu.sg/courses/61583
- Textbook
  - Artificial Intelligence: A Modern Approach (4th Edition)
    - IBSN (Print) 9781292401133
    - IBSN (eBook) 9781292410074
  - NUS Library:
    - https://linc.nus.edu.sg/search~S16?/i=9780134610993/i9780134610993/1%2C1%2C2%2CB/frameset&FF=i9780134610993&1%2C%2C2





Shopee

https://shopee.sg/product/849371650/20991479774

Consultations & Other Academic Support

- Consultations
  - By appointment only
  - Exhaust other channels first
- Canvas Discussions
  - Will be answered in reasonable time
- Telegram groups
  - One Telegram group per tutorial class
  - Managed by your tutor
  - Created by end of Week 2

Questions about the Lecture?

- Was anything unclear?
- Do you need to clarify anything?
- Ask on Archipelago
  - Specify a question
  - Upvote someone else's question

Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/67243073190

2 What is Artificial Intelligence (AI)?

Artificial Intelligence (in a Nutshell)

- Intelligent mechanisms that solve problems to help humans
  - Programs concerned with human actions / thinking
  - Intelligence assessed based on the mechanism's generality and performance
- Generality
  - More dynamic solutions → able to deal with many cases
  - Example
    - Google DeepMind's AlphaGo, AlphaZero, and MuZero
      https://deepmind.com/research/case-studies/alphago-the-story-so-far (with movies)
- Performance
  - Perform at least as well as humans (and preferably better than humans)
  - Not necessarily in the same way as performed by humans (or in nature) – e.g., birds versus planes

Kinds of AI

- Strong AI
  - General problem solver
  - Very dynamic programs → able to solve many different problems
- Weak / Narrow AI
  - Less dynamic programs → solves fewer problems (typically just 1)
  - Corresponds to most AI work
- Usually, focus is on Narrow AI
  - Easier to formalise
  - More on this later …

Note that solutions are typically assumed to satisfy some performance threshold (e.g., as good as humans)

How would you categorise ChatGPT?
https://openai.com/blog/chatgpt/

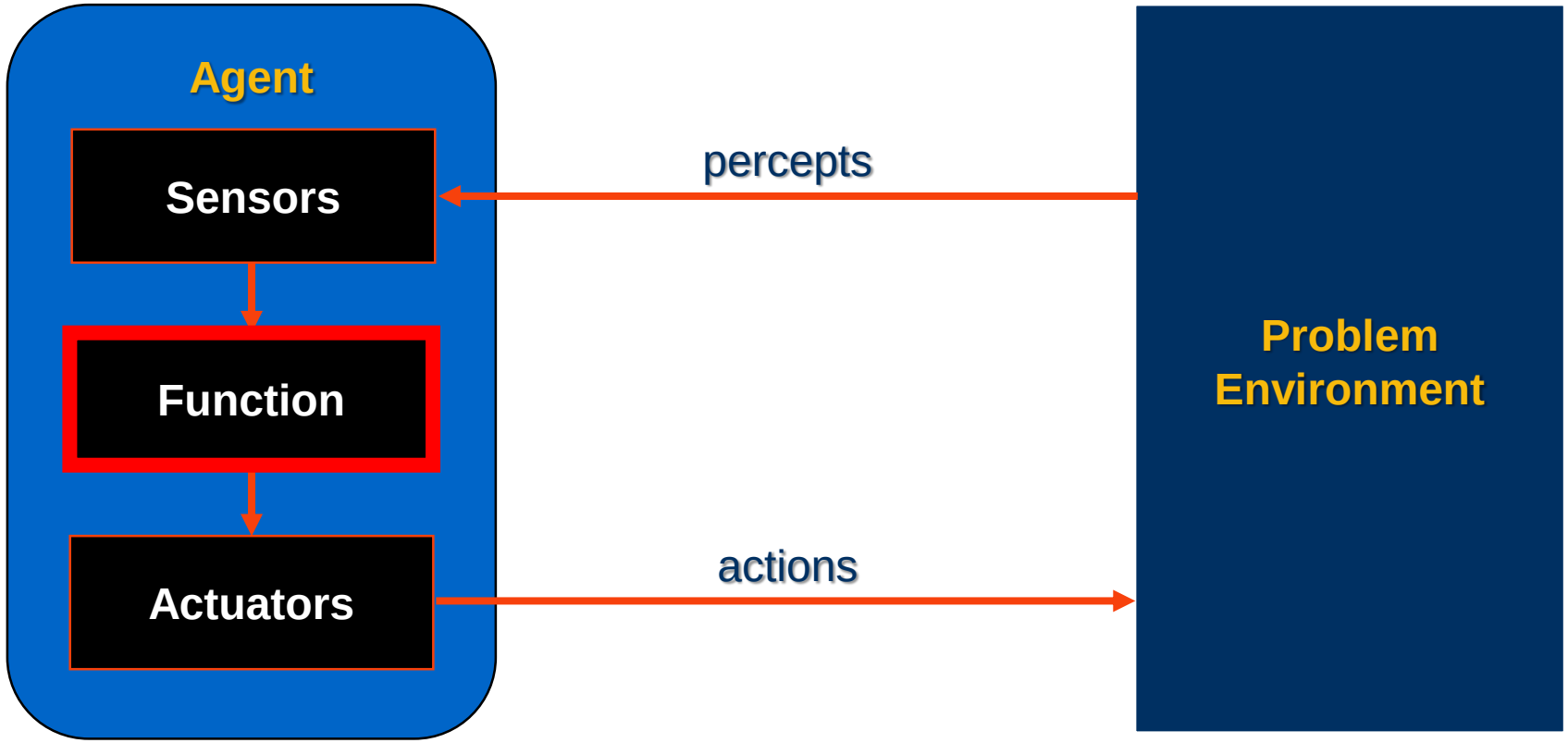
Stronger AI focusing on chat
(refer to the Turing Test for an interesting (but older) perspective on general AI)

Try asking ChatGPT the following

- *What questions are you unable to answer?*
- *Can you learn?*
- *Can you plan?*
- *Can you reason?*
- *Can you teach?*
- *Can you create?*
- *Can you feel?*

3 Intelligence Agents

Agent Framework



Agent Components

- Sensors and actuators
  - Sensors: what can/should be captured about the environment?
    - Percept data at time step t, p_t
    - Percept sequence, $P = \{ p_1, \ldots, p_t \}$
  - Actuators: how will the agent affect change in the environment?
    - Choosing an action to take from a set of possible actions, A
- Focus is on the agent function
  - Specify a function, f
    - Such that $f : P \rightarrow a_t$
    - Where $a_t \in A$ is the selected action given P

CS3243 focuses on
- Representations for P and A
- Algorithms that determine f

Rationality & Performance

- Desire a program that works well
  - At least better than humans; ideally optimal
  - Implies a quantifiable objective → performance measure
    - Are the objectives and performance measure aligned?
- Rational agent (function), $f : P \rightarrow a_t$
  - Given:
    - Percept sequence
    - Prior knowledge
    - Set of actions
    - Performance measure

    } available data
    - Rational agent optimises performance measure

Note:
do ***not*** assume agent is omniscient

Why do we find more Narrow AI?
Easier to define the performance measure and thus a rational agent to solve that problem

AI as Search: A First Look

- Goal in AI → determine agent function `f`
  - `f : P → a`
  - `a ∈ A`

Recall the agent framework
- Agent gets percepts
- Agent function determines action
- Agent enacts action
- Repeat

- Key idea → AI as graph search
  - Each percept corresponds to a state in the problem (state → vertex)
  - Define the desired states → goals
  - After each action, we arrive at a new state (action → edge)
  - Construct a search space (graph)

(1) Define performance measure and search space

  - Design and apply a graph search algorithm

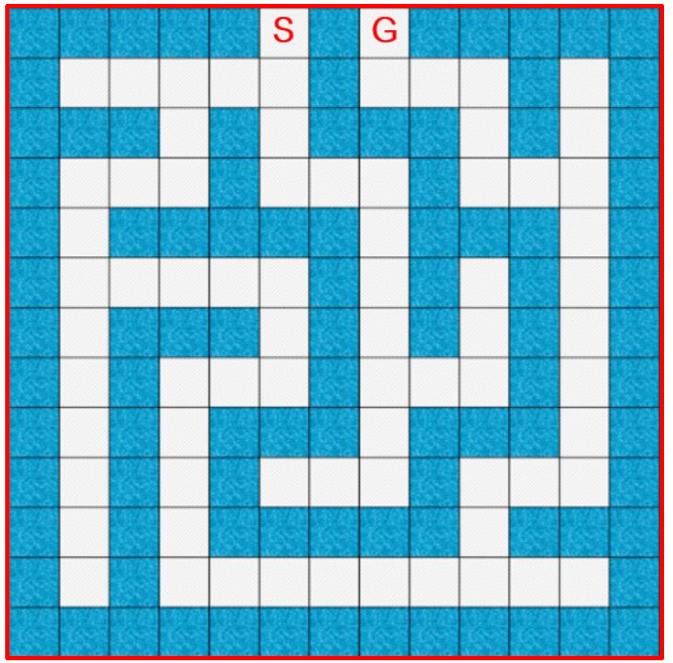
(2) Design search algorithm

This is the first type of problem we will look at in CS3243 (next lecture)

– other topics will expand on this idea

An Example Agent

- Problem environment
  - 2-dimensional maze navigation agent
  - p_t : (row, column)
  - A : { ←,↑,→,↓ }
- Agent function f ?
  - Assume map always the same
    - Function: series of if statements
  - Assume map is different each time but remains static during game
    - Function: determined by path planning algorithm (e.g., Dijkstra's)
  - What other possible assumptions?
    - We review this in the next part of the lecture



A Note on Value-Alignment

- Toy problems
  - Objective is clear
  - Performance measure is explicitly given and specific
- Real-world problems
  - Objectives not always clear or specific enough → uncertainty in objectives
    - Example: an agent that chooses the best drink for you…
      - Taste versus health?
    - Agent may have to learn objectives (e.g., through observation)
  - Not enough time to compute optimal solution? → limited rationality

Little to no value-alignment focus in CS3243

Questions about the Lecture?

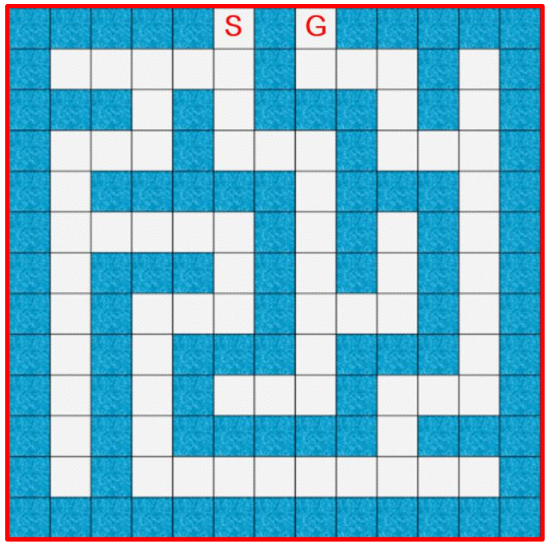
- Was anything unclear?
- Do you need to clarify anything?
- Ask on Archipelago
  - Specify a question
  - Upvote someone else's question

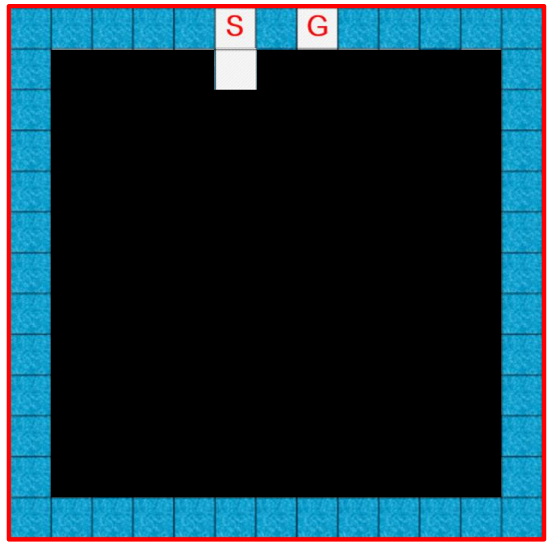
Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/67243073190

4 Problem Environments

Environment Properties

- Fully observable versus partially observable
  - Agent cannot access all information as some cannot be sensed
  - Requires inference (and/or, handling uncertainty)





Environment Properties

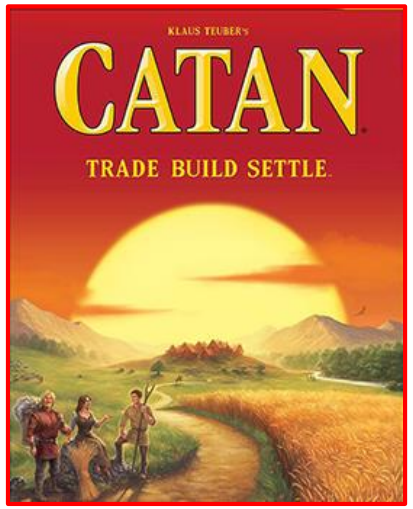
- Deterministic verses stochastic
  - Stochastic → intermediate state cannot be determined based on action taken at a given state
  - Handling uncertainty typically required

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Images taken from Wikipedia

Environment Properties

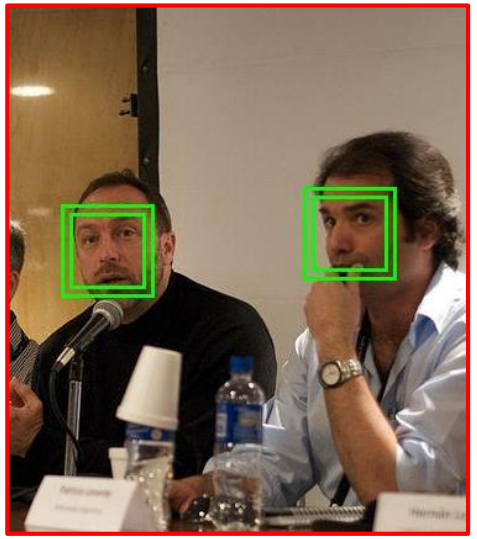
- Stochastic → partially observable?
  - May be fully observable (sense all) but still have randomness with action outcome

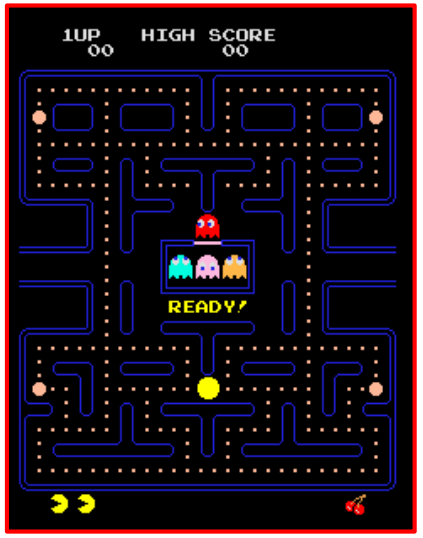


Images taken from Wikipedia

Environment Properties

- Episodic versus sequential
  - Episodic → actions only impact the current state (not those beyond)
  - Sequential → an action may impact all future decisions



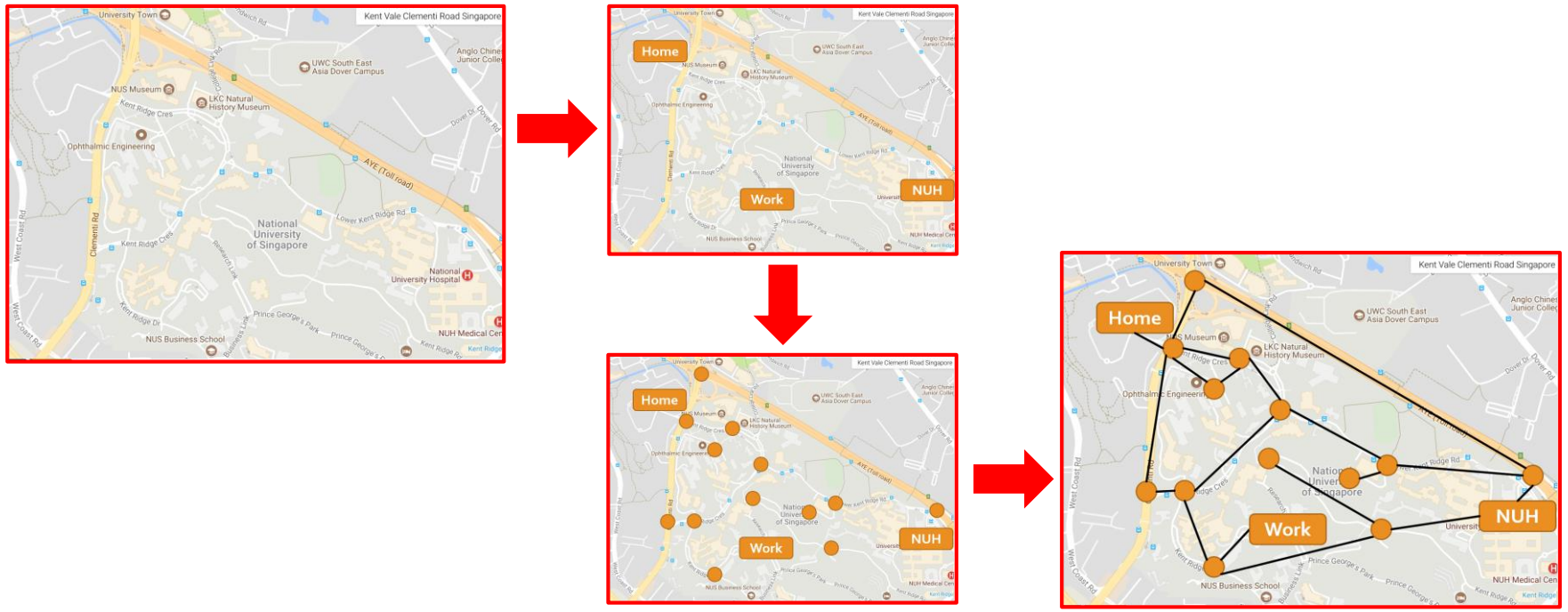


Images taken from Wikipedia

Note that it is possible to model a sequential environment as an episodic problem (more on this in the next lecture)

Environment Properties

- Discrete versus continuous
  - Refers to state information, time, percepts, actions



Environment Properties

- Single vs multi-agent
  - Do other entities exist in within the environment that are themselves agents whose actions directly influence the performance of this agent?
    - Chess → opponent is a competitive agent
    - Automated vehicles → other vehicles are cooperating agents
- Known versus unknown
  - Refers to knowledge of the agent / designer (not environment itself)
  - Includes performance measure
- Static versus dynamic
  - Will the environment change while the agent is deciding an action?

Environment Properties

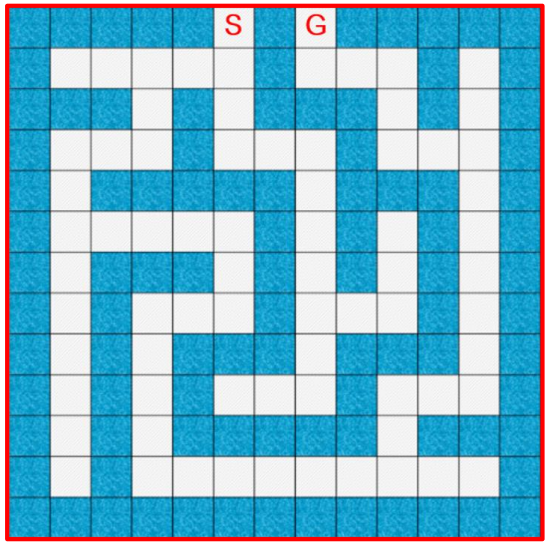
Property	CS3243	Notes
Fully / Partially Observable	Both	Latter in Logical Agents
Deterministic / Stochastic	Both	Latter in Bayesian Networks
Episodic / Sequential	Both	
Discrete / Continuous	Both	Mostly discrete
Single / Multi-agent	Both	Latter in Adversarial Search
Known / Unknown	Known	
Static / Dynamic	Static	

5 Taxonomy of Agents

Types of Agents

- Reflex agent & model-based reflex agent
  - Uses rules in the form of if-statements to make decisions
  - Direct mapping of percepts to actions
  - Mostly domain specific
  - Impractical with large search spaces

```
if at (0,5): ↓
if at (1,5): ←
if at (1,4): ←
if at (1,3): ↓
if at (1,2): →
if at (1,1): →
...
```



Types of Agents

- Goal-based and utility-based agents
  - Given
    - State and action representations
    - Definition of goals or utility
  - Determines
    - Sequence of actions necessary to reach goals or maximise utility
    - Or state that satisfies goal conditions or maximises utility
- Learning agents
  - Agents that learn how to optimise performance

Types of Agents

Agent	CS3243	Notes
Reflex Agents	Yes	
Model-Based Reflex Agents	Yes	• Logical Agents • Bayesian Networks
Goal-Based and Utility-Based Agents	Yes	• Uninformed / Informed Search • Local Search • Constraint Satisfaction Problems • Adversarial Search
Learning Agents	No	

Questions about the Lecture?

- Was anything unclear?
- Do you need to clarify anything?
- Ask on Archipelago
  - Specify a question
  - Upvote someone else's question

Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/67243073190