



# Games AI

## Lecture 10.1

Discussion and Performance

- Writing and presenting the report
- Performance



## Writing the Report

Kupo  
"I sorry. I enter  
you in Moogle Diary!"



- 30 Pages A4 **maximum**
  - Do not write 30 pages of text
  - That's to give plenty of space for illustrations / screenshots / graphs / diagrams
  - Screenshots to illustrate, for example:
    - The game
    - The AI in the game
    - Specific behaviours (series of screenshots/overlaid diagram)
    - Specific contexts/levels/scenerios (e.g. "it glitches on certain kinds of level geometry")
    - Test scenarios
    - Examples of content generated
    - Editing/content creation tools (if any). How it looks in the Unity inspector.

- Part 1.1 Introduction (20%)
  - Clear explanation of the game that was selected (25%)
  - Clear description of the intended behaviour of the AI system (25%)
  - Clear explanation of the techniques selected for use in developing the AI system for this game (25%)
  - Justifiable description of why these techniques were selected (25%)

- Clear explanation of the game that was selected
  - You're designing an AI system **for a game**
    - What is the game (intended to be) like?
    - It's fine if you haven't implemented everything!
  - This will let you talk about your AI **in context**
    - Lets you justify your decisions for your game
    - What AI is appropriate depends on the game

- Clear description of the intended behaviour of the AI system
  - The **goals** you have for your AI System
    - **These should relate to the game**
    - It's fine not to achieve them all!
  - Think about
    - What would the e.g. agents do? (Anything it should avoid doing?)
    - How would this impact the game and the player?
    - Does your game put any special requirements on the AI?
      - performs well (mobile?)
      - is customisable/reusable (lots of content creation?)
      - is controllable (authorial intent?)
      - is unpredictable (replay value?)

- Clear explanation of the techniques selected for use in developing the AI system for this game
  - What techniques did you use?
    - Introduce them in general terms (save implementation details for later)
    - Show that you understand the principles/theory involved
  - Think of this a bit like a Literature Review - introduce us to the concepts we will need **to understand what you have done or the decisions you've made**
    - Refer back to this when talking about implementation and design decisions



- Justifiable description of why these techniques were selected
  - Why did you use these techniques?
    - Why were these suitable for achieving your goals?
    - Make sure to contextualise it in your (intended) game
  - Were there any other factors influencing their selection?
    - Implementation time/effort? (What is it about these techniques that makes them easier to implement?)
    - Technical/library/game engine limitations?
    - Build requirements? (available compute to train)
    - Runtime requirements? (memory/CPU footprint)

- Part 1.2 Implementation Report (40%)
  - Clear description of the AI system architecture using the terminology taught during the course (30%)
  - Clear justification of major design decisions taken during the development of the AI system with reference to the desired behaviour, game experience feasibility of implementation, and runtime requirements of the system (30%)
  - Clear evidence of significant and complex artificial intelligence coding and development work (40%)

- Clear description of the AI system architecture using the terminology taught during the course
  - Imagine you've hired a programmer. Explain everything they need to know to keep developing your AI
    - Describe the implementation of techniques introduced earlier
    - This will be technical
    - Use technical terminology (states, response curves, fitness landscapes, classes, interfaces, delegates, ...)
    - Use diagrams/flowcharts/tables - a picture is worth 1000 words
  - Show you understand any third-party AI (e.g. Unity NavMeshAgent), but don't claim credit
  - Describe the game code to the extent that it's relevant to the AI
    - How does the AI get its data?
    - What behaviours are there? How do they work?

- Clear justification of major design decisions taken during the development of the AI system with reference to the desired behaviour, game experience feasibility of implementation, and runtime requirements of the system
  - This implementation was the result of your decisions - what were they and why did you make them?
    - Was technique X not working? (Why?)
    - Did changing to Y improve something (What?)
    - Did your timeline change? (How? How did your implementation plan change?)
    - Did you compare A and B in some metric? (What? Why? Which was better?)
  - Here you can technically justify your implementation
    - Was X the most performant implementation?
    - Did Y minimise garbage collection / draw calls / cache misses?
    - Was Z done to follow best coding practices / achieve modular AI?

- Clear evidence of significant and complex artificial intelligence coding and development work
  - Do something technically sophisticated
  - Describe it in a way that shows off its technical sophistication
    - I'm going to assume you've told me everything worth marks in the report
      - Make sure I don't overlook something
      - I won't go digging through your code to find it - it must be evidenced!
  - Use UML diagrams/code snippets/pseudocode to describe your code

- Part 1.3 Discussion (40%)
  - Clear understanding how the selected techniques and the implementation effected the behaviour of the system and a judgment of how successful this is at achieving the desired behaviour and/or desired game experience (50%)
  - Clear understanding of how the approach taken influenced the feasibility of implementation and the runtime requirements of the system (25%)
  - Clear understanding of alternative approaches that could be taken to solving the same problem and discussion of the pros and cons of these techniques (25%)



- Clear understanding how the selected techniques and the implementation effected the behaviour of the system and a judgment of how successful this is at achieving the desired behaviour and/or desired game experience
  - Describe how what you've implemented actually works
    - Clearly and objectively describe behaviour
    - Consider different contexts (e.g many/few opponents; levels size; level geometry) particularly those relevant to your game (intention)
    - Give examples/screenshots/diagrams
  - Evaluate it
    - Call back to your goals
    - Be honest - your might have tried something that failed, that's okay.
  - Quantify, if possible, in ways meaningful to your goals
    - Win rate? FPS? Average play time? Performance on mobile? User feedback?

- Clear understanding of how the approach taken influenced the feasibility of implementation and the runtime requirements of the system
  - A chance to talk about practicalities
    - Did it take longer to develop than you expected? Why?
    - Was the goal too big/small? Was the approach taken suited to implementing this within the time constraints?
    - Was it more resource-intensive than hoped? Why?
    - Did it take too long to train?
    - Did the game engine/programming language make your approach/technique more difficult? (e.g. was Unity suited to the style of game/AI?)

- Clear understanding of alternative approaches that could be taken to solving the same problem and discussion of the pros and cons of these techniques
  - You should have already discussed specific technical decisions (e.g. why A\* instead of BFS). The section should have a broader perspective.
  - Look further afield, show you understand the wider AI landscape
    - Could Machine Learning (e.g. neural networks) have been used?
    - What about whole-game search?
    - Could hierarchical state machines / utility AI / behaviour trees have been used instead?
    - Could content have been hand authored instead of generated or vice versa?
    - Could automated game testing have been used?
    - Could evolutionary algorithms have been used?
  - Give me an idea of how these could potentially be used. Give pros and cons.

- Playable Build (Potential Mark Adjustment)
  - Your submission should include a playable game with your AI system included.
    - If the game is not submitted – or cannot be run with reasonable effort – then your mark will be adjusted.
    - Remember the CD/102 computers are not very powerful, and they run Linux
    - I run Linux too, build for Linux
  - Instructions to run and play in README.md

- Source Code (Potential Mark Adjustment)
  - Everything I need to build the playable build you submitted
  - README.md
    - Instructions to build. You don't need to go into detail if it is a standard Unity build
  - LICENSE.md
    - Software license
    - Copyright statement

## Performance and Measurement

Lap					Session	
Name	Time	Delta	Offset			
Brands Hatch Indy						
Aslon Martin V8 Vantage GT4						
19. 4. 2015 23:07 - Practice						
Lap 1	0:52.6					
Lap 2	0:52.1					
Lap 3	0:53.289	0:00.976			0,00	
Lap 4	0:51.378	-0:00.935			0,00	
Lap 5	0:50.883	-0:01.430			0,00	
Lap 6	0:52.091	-0:00.222			0,00	
Lap 7	0:52.479	0:00.166			0,00	
Lap 8	0:01.799	0:00.000			0,00	
Hockenheim Grand Prix						
Ford Zakspeed Capri Group 5						
19. 4. 2015 23:14 - Time Trial						
Lap 1	2:00.329	0:00.000			0,00	
Lap 2	2:08.440	0:00.000			0,00	
19. 4. 2015 23:18 - Time Trial						
19. 4. 2015 23:19 - Time Trial						
Lap 1	1:57.544	0:07.373			0,00	
Lap 2	1:50.808	0:06.638			0,00	
Lap 3	1:50.171	0:00.000			0,00	
Lap 4	1:55.899	0:05.729			0,00	
Lap 5	0:52.560	0:00.000			0,00	



The screenshot shows the iRacing software interface. On the left, the 'Channels' panel is open, displaying data for three laps. The third lap is selected. The 'Track' panel at the bottom shows a track map with a red car icon indicating the current position.

**Channels Panel Data:**

	Master	Overlay	Delta
<b>Lap</b>			
Lap	3	4	-0:05
Time	1:50.171	1:55.899	-0:00
S1	0:24.747	0:25.572	-0:00
S2	0:50.754	0:52.155	-0:01
S3	0:34.669	0:38.172	-0:03
Distance (km)	4.55	4.54	0.01
Fuel Used [l]	0.00	0.00	0.00
<b>Cursor</b>			
Time	0:53.917	0:55.650	-0:01
Distance (m)	2397	2397	0
<b>General</b>			
Speed (km/h)	180,6	180,5	0,2
Engine RPM [rpm]	9202	9174	27
Lat Accel [G]	-0,61	0,25	-0,86
Lng Accel [G]	0,35	0,37	-0,02
Fuel Level [l]	5,00	5,00	0,00
<b>Input</b>			
Gear	3	3	0
Throttle [%]	95,1	96,5	-1,2
Brake [%]	0,0	0,0	0,0
Steering [%]	0,0	-2,2	2,2
Clutch [%]	0,0	0,0	0,0
<b>Engine</b>			
Fuel Pres (MPa)	59	59	0

**Track Panel:**

Track:



- Measuremnt
  - Time.realtimeSinceStartup()
  - FPS Counter

- Profiling Performance
  - Unity Profiler