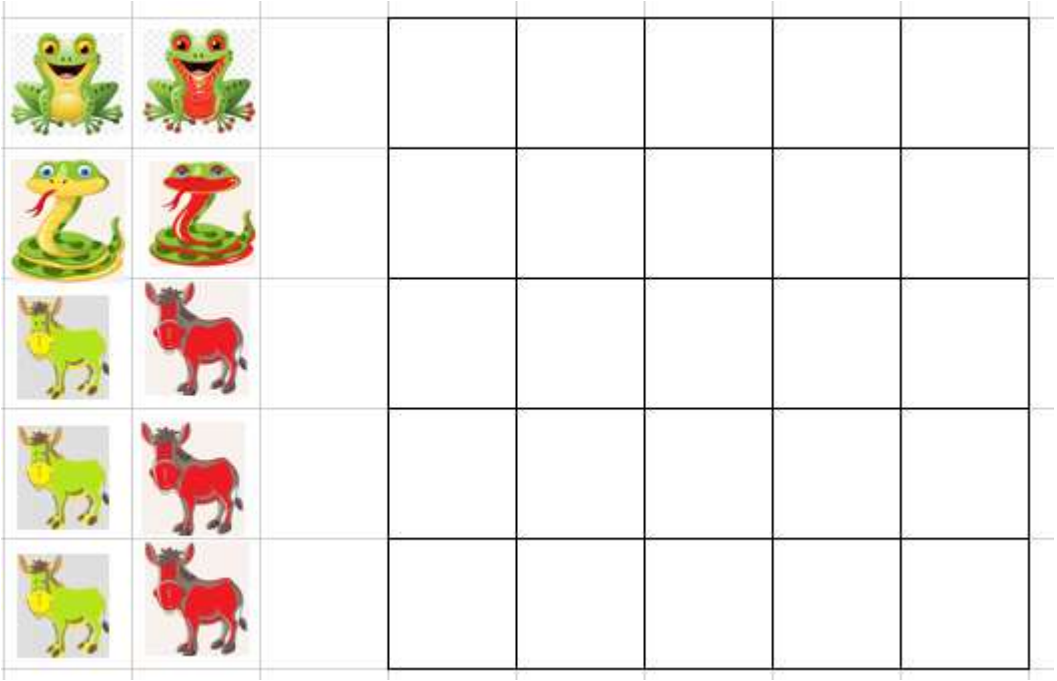
Lecturer: Dr Oisín Cawley

Continuous Assessment 1
Value: 20% of overall mark
Due Date: Fri 5th Dec 2025

The Fourth Protocol

You must produce a digital version of the game The Fourth Protocol. **The aim of the game is to simply gets 4 pieces in a row!** The rules are simple enough ☺.

The Game
It is a two-player, turn-based, perfect-information, zero-sum game. The board is a 5x5 grid. Each player has 5 pieces. 1 Frog, 1 Snake, and 3 Donkeys.



One player starts (by which ever means you want to decide that). Each player alternates placing one piece in any unoccupied place on the board until all pieces are placed (or the game is over because a player achieved 4 in a row straight away).
Once all pieces are placed, players can move pieces around in order to try and get 4 of their colour pieces in a row. However, pieces can only move as follows:

- Donkey can only move one space in any direction but NOT diagonally.
- Snake can only move one space in any direction INCLUDING diagonally.
- Frog can normally only move one space in any direction INCLUDING diagonally. However, in addition, it may also jump over any other pieces to a free spot as long as it is in a horizonal, vertical or diagonal direction on the board. It can't just jump to any random spot on the board.

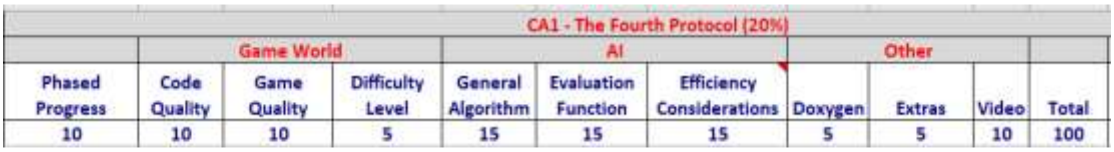
You are also to implement an AI that can play the game against a human utilising the minimax algorithm.

Submit a zipped project to Blackboard.

Project Details

Game to be produced in **C++** with **SFML 3** and documented using **Doxygen**. Please use an environment variable **$SFML_SDK** to point to your SFML install location.

Overall marks for each component as specified in the table:



CA1 - The Fourth Protocol (20%)										
	Game World			AI			Other			
Phased Progress	Code Quality	Game Quality	Difficulty Level	General Algorithm	Evaluation Function	Efficiency Considerations	Doxygen	Extras	Video	Total
10	10	10	5	15	15	15	5	5	10	100

10% for evidence of continuous project progress. This is to be **demonstrated in the lab each Friday**.

Phase 1 – Core game logic and basic AI (Nov 14th)
Phase 2 – Minimax implementation with basic evaluation function (Nov 21st)
Phase 3 – Advanced minimax and evaluation function (Nov 28th)
Phase 4 – Polished Game frontend and hand-up (Dec 5th)

45% of the marks going for the AI. The "General Algorithm" refers to your implementation of Minimax. "Efficiency considerations" refers to how you handle the large search space, so for example, implementing Alpha-Beta pruning.

5% for the "Difficulty Level" is to allow the game to be more/less challenging.

5% bonus available for extras (make sure it is obvious or that I know about it). Wouldn’t it be cool if the AI could also be set as the player, then the AI could play itself! Or how about a visual AI analyser which shows the AI's decision-making process: "Considered X moves, selected Y with a score of Z."

This is obviously based on an existing game but with some different rules. There are lots of implementations of 4 in a row online so I advise that you do your own. Note the **10%** for Video. This should be a max **5-minute video** explaining your code in detail. Make sure the code is large enough to see in the video.

Ensure that any specific technical requirements of the game are clearly highlighted. If you use external libraries, for example Thor, make sure they are part of your project. I don’t want to have to go looking for and installing stuff.
The best way to provide the above information is to submit an accompanying **Readme.txt** file with any points you think are important.

Team or Individual
This is a team (of 2) project. You may do it on your own if you prefer. You must email me the names of the people in your team asap. Project submission must include the % of each component that each student contributed to. However, you might simple agree the effort was 50/50 therefore you would both get equal marks. Sometimes this is not the case, and so the team must therefore agree on these %s when submitting (the total must add up to 100).