This document provides some additional information to help students select what kind of experimental data they should collect to evaluate AI techniques appropriately.

* **Importantly, the data gathered must be informative in regards to the effectiveness or usefulness of the AI in light of a specific and explicitly stated goal.**
* **In addition, the AI should be evaluated under different configurations and/or conditions to provide a full analysis of the technical implementation.**

The type of data you gather will depend on the specific goals and implementation you have attempted. The following provides a list of ideas to get you started thinking about how you might evaluate your AI. You do not need to do everything mentioned here and you might have other ideas that are more suitable to your application. You are strongly encouraged to discuss your AI idea and evaluation strategy with module tutors in class to make sure they meet assessment requirements.

# General Experiment Considerations and Measuring Effectiveness

Consider what “effectiveness” or “usefulness” means for your AI. The experiment you construct and the type of data you gather will be directly tied to your goals. The following examples provide some appropriate examples.

**AI Car Following a Track:**

Design different race tracks with different path shapes or ordering of obstacles, then test your AI on all tracks and see what track properties are easier or more difficult for the AI. Effectiveness can be measured by number of collisions, proximity to an “ideal” path (such as the middle of the track), or others.

**AI Game Opponent:**

In this scenario, e.g. checkers, you can pit the AI against itself under different game conditions, e.g. different board sizes. You might change some parameters of your AI (see next section) to create slightly different versions of it. You can then setup a tournament between the versions. The data you gather can be how many wins, losses, and ties each AI version got. Other stats might include specific game metrics like how many game pieces were lost, how much of the game field was controlled by the AI, what strategic behaviours were preferred by the AI, etc.

**AI Content Generation and Others:**

If you have an AI that generates some game content, whether it be level design, dialogue interactions, visuals, or others, you have to be more creative and resourceful in how you evaluate it. Consider unit testing, where you might create a systematic experiment that runs the AI under all possible conditions and edge cases. Depending on your AI goals, effectiveness might be measured in terms of:

* Variety and distinctiveness of the generated content or behaviour (not looking the same all the time). Ideally, create an objective, quantitative measure of this. This could be counts in different categories, descriptive stats like means and standard deviations, or some kind of similarity measurement.
* Time series evolution of characteristics that change or evolve over time. For example, if you have a learning algorithm that evolves aspects of the game’s world or behaviour, you can track how certain metrics change over time.
* User feedback survey data on how the AI was perceived by people. However, this usually requires gaining ethical approval first as it involves people other than yourself, so you must have enough time to conduct this. This approach is generally discouraged in this module due to the time frame available.
* Performance considerations, such as frame rates, loading times, memory usage, CPU/GPU utilisations, algorithm time and space complexity, parallelisation opportunity, etc.

# Technique-specific Considerations

Depending on your chosen AI technique there would be different configurations to consider testing against. The following is not an exhaustive list but should get you started on what properties to consider manipulating for your experiments in order to find the best values for your AI.

**Fuzzy Inference System**

* Number of fuzzy variables to consider
* Number, shape and range of membership functions
* Number and complexity of fuzzy rules
* Defuzzification methods

**Genetic Algorithms**

* Chromosome size and representation
* Crossover methods
* Mutation factors
* Population size
* Fitness calculation and selection criteria
* Number of generations and stopping rules

**Reinforcement Learning**

* Reward shaping
* Exploration policies
* Learning rate and discount factor
* Initial conditions

**Neural Networks**

* Input dimensions
* Number of layers
* Number of neurons per layer
* Learning rate
* Loss function
* Optimiser type