Assignment 2b: Genetic Programming Search

In this assignment, you will implement a GP search to find high-performance Pac-Man controllers in GPac. You are asked to complete the Jupyter notebook `2b_notebook.ipynb` and several other Python files as directed by the notebook.

You need at minimum to implement support for the following EA configurations, as described in the notebook:

Representation Parse tree

Initialization Ramped half-and-half

Parent Selection Fitness Proportional Selection, k-Tournament Selection with replacement, uniform random selection

Recombination Sub-Tree Crossover

Mutation Sub-Tree Mutation or Point Mutation

Survival Selection Truncation, k-Tournament Selection without replacement

Bloat Control Parsimony Pressure

Termination Number of fitness evaluations

Your submission should also contain a report to document the findings of a 10-run experiment as well as files containing the game log and parse tree from the controller with the highest base fitness from all runs. In your report, include the following:

- A table of every EA parameter used in your experiment.
- A histogram showing the distribution of scores across all runs in the experiment.
- An evals-vs-metrics plot showing the progress of evolution averaged over 10 runs (like assignments 1b-1d).
- Statistical analysis comparing the best fitness obtained by each run to data generated by the random search algorithm you implemented during Assignment 2a. This should include the sample size, sample means, standard deviations, the test's p-value, α, and a brief discussion interpreting the results of the statistical test.
- An informal comparison of the behavior of the best Assignment 2a agent and the best agent from this experiment, by analyzing the visualization of the highest-score game from each algorithm, and by comparing and contrasting their respective highest-scoring parse trees.

The deliverables of this assignment are:

GREEN 1 Your source code and completed notebook

GREEN 2 A PDF document headed by your name, AU E-mail address, and the string "COMP x660 Fall 2025 Assignment 2b", where x needs to reflect the section you are enrolled in, containing your report, including statistical analysis and plot(s)

GREEN 3 Files containing any data you analyzed to write your report or generate your plot(s) saved to the `data` directory of your repository

YELLOW 1 Up to 20% (bonus for COMP 5660 students, required for COMP 6660 students) can be earned by investigating the impact of casting parsimony pressure as a second objective. That is, instead of using a parsimony pressure penalty function, treat this as a multi-objective problem where your GP search is trying to maximize game score and minimize number of nodes. Combine all Pareto fronts from the final generation of each run into one set of solutions, then use the Pareto front of this set for plotting and behavioral analysis. Plot this Pareto front similarly to Assignment 1d, and conduct an informal analysis of the highest-scoring tree, the smallest tree, and one tree from somewhere in the middle of the Pareto front. You also must conduct a control experiment, which keeps track of the same metrics and produces the same artifacts in your report, but uses the same parsimony pressure mechanism as your GREEN experiment. Finally, you should conduct statistical analysis comparing the global best score found during these two experiments. That is, compare the global best scores, ignoring the parsimony pressure mechanisms from each experiment. Include these results in a separate dedicated section of your report.

RED 1 Up to 25% bonus can be earned by implementing fitness sharing and investigating its impact on population diversity, as measured by phenotypic distance. A distance metric will be provided for you based on the different components of the GPac score function. This experiment should include all the same components as your GREEN experiment, with the following changes: **1)** the evals-vs-metrics plot must *also* visualize the population diversity (calculated as the population's average pairwise distance in phenotype space, as described in the notebook) at each generation, averaged over 10 runs, **2)** statistical analysis and behavioral comparison should be conducted against your GREEN experiment, rather than your random search, and **3)** a second, additional statistical comparison should conducted against your GREEN experiment based on population diversity from the final generation of each run. Include these results in a separate dedicated section of your report.

RED 2 Up to 5% bonus can be earned by investigating having multiple simultaneous Pac-Man agents all employing *identical* controllers, where they all have to die for the game to end, and they share the same score (i.e., there's no competition between the Pac-Man agents). You must add a new primitive, M, returning the distance to the nearest OTHER Pac-Man agent. This experiment should include a similar plot and informal agent analysis as the GREEN experiment (but does not require statistical analysis). Include these results in a separate dedicated section of your report.

RED 3 *If you have completed RED 2*, up to 15% additional bonus can be earned by investigating having multiple simultaneous Pac-Man agents all employing *different* controllers, where they all have to die for the game to end, and they share the same score (i.e., there's no competition between the Pac-Man agents). In addition to the M primitive implemented in RED 2, you must implement an alternative version of the `play_GPac` function called `play_GPac_multicontroller` that accepts multiple Pac-Man controllers (in addition to the typical parameters) and uses each controller to determine moves for a particular Pac-Man agent. Evolution should utilize a single population of controllers, and fitness should be re-assessed each generation with stochastic controller pairing such that all individuals (including adults!) play at least one game per generation, no individual plays more than one game more than any other individual, and the base fitness of an individual is determined by averaging the scores obtained in each game the individual played in during the current generation. Each game played should be counted as one fitness evaluation. This experiment should include a similar plot and informal agent analysis as the GREEN experiment (but does not require statistical analysis). Make sure you analyze all of the parse trees that participated in the highest-scoring game, and compare them to what you observed during the RED 2 experiment. Include these results in a separate dedicated section of your report.

RED 4 Up to 10% bonus points can be earned by investigating the evolution of a controller that controls all ghosts and plays against the default Pac-Man strategy. This requires adding a new primitive (M, which returns the distance to Pac-Man) and modifying the G primitive to return the distance to the nearest OTHER ghost. These will be required for Assignment 2c, so consider this an opportunity to

get a head start. This investigation should use negative game score as the base fitness metric. This experiment should include a similar plot and informal agent analysis as the GREEN experiment (but does not require statistical analysis). Include these results in a separate dedicated section of your report.

RED 5 *If you have completed RED 4*, up to 15% additional bonus can be earned by investigating having multiple simultaneous ghost agents all employing *different* controllers, against the default Pac-Man strategy. All ghosts share the same fitness for a particular game (i.e., there's no competition between the ghost agents). In addition to the components implemented in RED 4, you must implement an alternative version of the `play_GPac` function called `play_GPac_multicontroller` that accepts multiple ghost controllers (in addition to the typical parameters) and uses each controller to determine moves for a particular ghost agent. Evolution should utilize a single population of controllers, and fitness should be re-assessed each generation with stochastic controller pairing such that all individuals (including adults!) play at least one game per generation, no individual plays more than one game more than any other individual, and the base fitness of an individual is determined by averaging the scores obtained in each game the individual played in during the current generation. Each game played should be counted as one fitness evaluation. This experiment should include a similar plot and informal agent analysis as the GREEN experiment (but does not require statistical analysis). Make sure you analyze all of the parse trees that participated in the highest-scoring game, and compare them to what you observed during the RED 4 experiment. Include these results in a separate dedicated section of your report.

Submit all files via GitHub, by *pushing* your latest commit to the default branch, including readyToSubmit.txt. The due date for this assignment is 10:00 PM on Sunday November 16, 2025.

Grading
The point distribution is as follows:

Assessment Rubric \ Deliverable Category	Green	Yellow	Red 1	Red 2-5
Algorithmic	50%	55%	50%	60%
Tuning	10%	0%	0%	0%
Programming practices, readability, and implementation	20%	25%	25%	20%
Report and plot(s)	15%	15%	20%	20%
Statistical analysis	5%	5%	5%	0%