

# Features, Requirements, and Evaluation Criteria

*Gavin Scott, Kevin Patel, Jason Swanson, Spencer Mulligan*

## *The Super Mario Bros*

### Features

- An emulated Super Mario game environment that can be played either manually or through an autonomous agent
- The game passes a game state to the selected agent each frame and responds to its movement decision
- Three different autonomous agents in addition to a manual player
  - Reactive Agent
  - Mimicking Agent
  - Neural-Network Agent
- An automated procedural level generator for the game based on user-provided constraints

### Requirements

- AI vs Manual Control
  - Command-line argument specifies whether Mario is controlled by user input or by algorithm input.
  - When under AI control, the player will not be able to affect the gameplay in any way, and the AI will attempt to finish the level on its own.
  - If an agent finishes the level it will restart the level and continue
    - If the level was generated a new one will be generated with the same constraints.
  - If an agent fails to complete the level (dies or times out), it will restart the level and try again.
    - The neural network agent will be “killed” if it stands still or fails to increase their score for too long
  - The game-speed will be able to be increased to speed-up testing of agents
- The game state is taken at each frame and passed to the selected bot (if not under manual control)
  - Game state includes the position of all objects in frame, including the player control) to receive its movement decision
- Game state is cached frequently to be used as training data for the mimicking bot
- Automatic Level Generation
  - The user will be able to provide constraints to the level generator in terms of what game elements they want included in the level, as well as how many of each element.

- The requested elements will be randomly placed throughout the level, ensuring that there is a valid path and no elements overlap with one another
    - A generated level will also have by default a floor, Mario at the far left, and the finishing flag at the far right
  - The reactive agent, given a set of rules, will be able to perform well and be used as a baseline to compare to the other bots
  - An agent using a neural network will learn how to play based on the provided game states and a heuristic
    - The performance of the neural network agent will improve as it plays and learns how best to improve its score
  - A mimicking agent will use data from a human player to learn how to play a level on its own.
    - The human will be able to decide at any time whether they want to cede control to the agent.
    - The level(s) the agent plays will be different than the level(s) the human played when using the level generator

## Evaluation Criteria

- User input
  - The user can manually control the game.
  - The user can specify a level based on a set of constraints, which will be generated in a way that ensures the level is possible.
  - The user cannot influence the behavior of a bot once it has control
- Agents
  - All agents have been implemented
  - The all agents perform as expected
  - The mimicking agent acts accordingly to the strategy of the player it's learning from
  - The neural network bot progressively improves as it learns
- Logging
  - Logs of the game state can be traced back in order to discover trends in decision making.
- Miscellaneous
  - The game rarely crashes.
  - Level generation is implemented.