Melissa Zhang
CSCI 1710 Logic for Systems

Modelling and Conducting Property-Based Testing on “Among Us” in Forge

Context
Forge is a declarative language used to model and verify software systems. Forge can specify complex structural constraints, define behavior in a system, and test for unexpected or unwanted behavior. In this project we used Forge to build a model of the popular quarantine game “Among Us”, then conducted PBT (Property-Based Testing) to learn about properties of the system.

Among us is a multiplayer game with two teams: “imposters” and “crewmates”. The crewmates’ goal is to travel around rooms in the gamespace and complete tasks, while the imposters’ goal is to kill all crewmates before they can complete all the tasks. After each death, players vote as a group to “eject” a player they believe is the imposter. If crewmates are able to eject all imposters, or complete all tasks, they win. If imposters are able to kill all players before all tasks are completed, imposters win.

Motivations
In real games, players often use different strategies to approach the game to increase their chances of winning. For instance, some Imposter players may choose to continuously follow a Crew, and some Crew players may choose to move in a large group to ensure safety. Moreover, the number of Crewmates and Imposters in a game may vary, changing the effectiveness of strategies. How do different combinations of strategies and system constraints on the number of players affect the game’s outcome? What can we learn about the game’s properties?

Model & Simplifications

Our forge model simplifies parts of the game mechanics and encodes several behaviour strategies that might be employed by Crewmates and Imposters. We decided to focus on game actions inside rooms, by excluding hallways (or edges between rooms) as valid locations for killing, and excluding "travel" as a state. We also decided that players would be allowed to do nothing (remain in place) between states, not only to make sure our traces in temporal Forge could be infinite but also because this is valid in a real game and can introduce a factor of randomness to our otherwise prescribed movement strategies for each player.

We also implemented a suspicion-based voting system in place of the conversational group vote in the original game. In real games, after each killing, players discuss as a group who is suspicious and then collectively vote on someone to eject. In our model, each player receives a “suspicion score”: each Crewmate is suspicious of every player they were not in the same room as when the kill occurred; each Imposter is suspicious of (and adds +1 to the suspicion score of) every Crewmate. A group vote is automatically triggered after each death, and a player with the highest suspicion score is voted out.

Our model includes several behavioral strategies that might be employed by Crewmates and Imposters, which we later used in combinations to conduct PBT testing.

Crew Movement Strategies

- **moveCrewRandom**: Move to new rooms randomly
- **moveCrewTasks**: Prioritize rooms with uncompleted tasks
- **moveCrewPairs**: Move in pairs
- **moveCrewGroup**: Move together with a group of 3+
- **moveCrewDivideAndConquer**: Prioritize completing tasks as fast as possible by splitting up and dividing tasks

Imposter Movement Strategies

- **moveImposterRandom**: Move to new rooms randomly
- **moveImposterFollow**: Follow crewmates and prioritize rooms with potential victims

Findings

Through writing extensive PBT tests, we were able to learn and verify a variety of properties in the Among Us game system. Our most interesting takeaways are:

- In a game with exactly 1 Crew, Imposters always win.
- In a game with exactly 1 Imposter, the crewMoveInPairs strategy guarantees a Crew win, while the crewMoveInGroups (of 3+) guarantees a Crew win, as well as the constraint that no player ever dies. The “no one ever dies” predicate is not implied by any other movement strategy.
- In the standard game with exactly 7 Rooms, 3 Crew, and 1 Imposter, if Crewmates prioritize completing tasks with speed (using the divideAndConquer strategy), and Imposters prioritize following Crew, it is impossible for Crew to complete all tasks before the game ends. However, it is still possible for Crew to win the game overall by voting out the Imposter.