CMPSC 310
Artificial Intelligence
Fall 2018
Janyl Jumadinova

Lab 5 Part 2
29 October, 2018
Due: 12 November by 2:30 pm
This is an individual lab.

Objectives

To extend the practical knowledge of the Unity video game development platform. To create a video game that utilizes AI planning within the Unity platform. To correctly complete a simple implementation of the game using the concept of finite state machines. To extend the finite state machine implementation to utilize probabilities for transitions from state to state and utilities for the agent.

Lab 5 Structure and Reading

Last week you completed a tutorial to familiarize yourself with the Unity game development platform and to develop a Survival Shooter game. In the second portion of lab 5 you will incorporate some aspects of AI planning into the game development.

To learn more about Unity, please consult its documentation at https://docs.unity3d.com/Manual/. To review material on planning for the second part of the lab, please read Chapter 9 in the FCA online book we are using in this course. For details on C# language, you can utilize one of the resources listed below:

- https://unity3d.com/learn/tutorials/topics/scripting/coding-unity-absolute-beginner
- https://en.wikibooks.org/wiki/C_Sharp_Programming

Also, as you write your reflection, please refer to the relevant "GitHub Guides", available at https://guides.github.com/, that explain how to use many of the features that GitHub provides.

Tasks to Complete:

1. Finite State Machine Representation in Video Games

Study the Survival Shooter game and design a diagram that represents the states and the rules of transitions from state to state by the autonomous agent (NPC) in this game.

2. Probabilistic State Transitions in Video Game NPC

Now, construct a Markov Decision Process representation of the game based on the finite state machine from the previous portion. You will need to introduce transition probabilities and rewards for the agent taking an action and transitioning from one state to another state.

- Draw a diagram to represent your MDP and specify what different components of your MDP are in the context of the game.
- Devise an algorithm for how the agent selects actions and transitions from state to state (or you may utilize one of the algorithms described in the book). Implement the changes in the game. You have freedom in this portion in the specific details of the algorithm you decide to implement, as long as it incorporates states, transition probabilities and rewards, you will satisfy this requirement.

3. Changes to the Environment

Make at least **three** visual changes to the scene of the game. This could include changing the environmental scene, music and/or making changes to the objects in the environment.

4. Experiments

Conduct user experiments, where you collect data from at least 20 plays of the game by multiple players (use your friends). The collected data should, at the minimum, contain information on:

- The number of total state transitions.
- The number of times each state was visited.
- The reward values over time.
- Final game results (score, win/loss).

Display your collected results in graphs or tables, and briefly comment on the performance of your NPC.

Required Deliverables

This assignment invites you to submit electronic versions of the following deliverables through your lab repository. You are to use the same repository you created in the first part of the assignment.

1. A completed and properly documented source code of your extended game implementation, stored in the `src/` directory. Make sure you clearly identify this project as a separate from the game project you completed for part 1.

2. An extended `writing/gameModification.md` document that adds the Game Modification details to the document you created last week. Specifically, as you decide on and implement your extended game, you should add the following information to the `writing/gameModification.md` document:

    - Finite State Machine diagram for the original game as outlined in task 1. Please see https://guides.github.com/features/mastering-markdown/ on how to embed images in the Markdown document.
    - What is the new objective of the game (if changed)?
    - Describe how you improved the game. Include (1) Markov Decision Process diagram and its explanation with all of the parts of MDP specified for task 2 of this lab, and (2) your implemented algorithm, described in general English terms or in a pseudocode form.
    - Describe the visual changes made to the environment. Incorporate the screenshot of your visual extensions.
    - List three reasons why your ideas may not have been implemented by the original designers.
    - List three reasons why you think players would like your extensions.
    - Present the results of your experiments and their analysis from completion of step 4.

3. Extended `writing/reflection.md` document that contains an additional paragraph outlining the steps you followed to complete the second portion of the lab and describing any challenges you have encountered.

Extra Credit (30 points)

For extra credit, consider utilizing another game in Unity (other than Survival Shooter), enhancing it to Finite State Machine and then MDP representation, and completing tasks 3 and 4, as they are described above.