CS 5100: Foundations of Artificial Intelligence

By Aakash Mahesha, Samanvya Tripathi, and Jack Dewey

Project Proposal

1. Project Description

For our final project, we propose to develop a computer vision program that interprets 2D 'minimap' information to perform simple navigation tasks in a 3D environment. We intend to try this idea on two video games, first a simple 2D atari game similar to those used in the OpenAI gym, and then a 3D role-playing game with on-screen graphical user interface (GUI) elements designed to help human players navigate the world–most prominently an on-screen minimap and player heading indicators. We will use these 2D maps to guide the AI as it moves around the map following pre-existing pathways in the environment, as in source [1]. This will involve using simple vision techniques to extract and interpret the GUI elements and logic to map these pathways onto gameplay commands directing player movement.

2. Approach / Algorithms

We plan to extend this basic functionality by incorporating several concepts we have learned this semester–a combination of heuristic search, reinforcement learning (deep and Q learning) [3,4,5], and CNN-guided object detection [6] to allow for the play to recognize goals on the GUI map and then pursue those goals in the corresponding 3D world. We will not attempt to attain any more complicated behavior than moving toward goals and away from obstacles / enemies, and will not attempt to direct the agent to engage in combat.
[note: we will change this if it ends up being doable to add more, but I think this would be much more complicated to try to attempt.]

3. Planned Comparisons

Similar to the programming assignments for this course, once we have implemented the functionality to detect and interpret the on-screen GUI elements, we will be able to use that information to direct player navigation in a manner analogous to a PacMan navigating a 2D maze. As such we will be able to use map markers indicating the presence of enemies and / or target locations to guide player movement in more or less optimal ways. We will also be able to augment this functionality by using vision techniques to integrate information visible in the game world but not directly accessible via the onscreen map–for example, improving avoidance of obstacles and enemies.

4. Allocation of Work

We will collaborate on designing the basic algorithms and heuristics
Samanvya Tripathi - Gameplay algorithm (policy, eval)
Jack Dewey - Parsing the minimap(s)
Aakash Mahesha - Object recognition for 3D scene

5. Resources:

[1] “Cyber Python 2077 - Using computer vision to read and walk from Cyberpunk 2077 map” Youtube, uploaded by sentdex, 15 Jan 2021, https://www.youtube.com/watch?v=dUU6ZsJIZKQ

[2] Shafaei, Alireza & Little, J.J. & Schmidt, Mark. (2016). Play and Learn: Using Video Games to Train Computer Vision Models. 26.1-26.13. 10.5244/C.30.26.

[3] M. Kempka, M. Wydmuch, G. Runc, J. Toczek & W. Jaśkowski, ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning, IEEE Conference on Computational Intelligence and Games, pp. 341-348, Santorini, Greece, 2016 (arXiv:1605.02097)

[4] Stable Baselines [source code], https://github.com/hill-a/stable-baselines

[5] Open AI Gym API [source code], https://github.com/openai/gym

[6] Yilin Wang, Jiayi Ye: An Overview of 3D Object Detection, 29 Oct 2020, https://arxiv.org/abs/2010.15614