Lavish Gulati - 170101082
Vakul Gupta - 170101076

CS461 - Computer Graphics - Project Report

Abstract

This project will address the problem of cognitive character animation. We propose the use of finite state machines (FSMs) for the behavioural control of characters. Our approach rests on the idea that the cognitive character arises from the evolutionary computation embedded in the artificial life simulation, which in our case is implemented by the finite state machine.
This project to build virtual worlds is aimed at the graphic simulation of an arena, where small mobile robots can perform requested tasks while behaving according to their own motivation and reasoning. Each robot is an intelligent agent that perceives the virtual environment through a simulated vision system and reacts by moving away from or approaching the object it sees.

Technical Challenges

➔ **3D SIMULATION ENVIRONMENT** - Since we would have to build neural networks and also do image preprocessing using OpenCV (both of which have current implementations in Python language), we also need a 3D simulation environment tool which uses Python language to ensure single language support. But to the best of our knowledge, no such prominent tool exists and we would have to use primitive tools such as OpenGL or Java3D.

➔ **LINKING C++ AND PYTHON** - The visual sensor of the ROBOT viewport outputs the RGB image which needs to be inputted into the Python code for OpenCV. So we would have to bridge the links between C++ and Python for proper execution of the program.

Current Work

The current research paper - "An artificial life approach for the animation of cognitive characters" explains the current model - Click Here
In the current implementation, there is a 3D robot in a simulated environment (or arena) with sensing and mobility skills. The robot decides which action it should take at every moment. A finite state machine has been evolved from scratch for this purpose. The robot should look for energy sources while avoiding traps that make it weaker. Based on visual information, the robot plans its way to get closer to the energy sources, and a finite state machine that evolved through generations in fact conducts this planning.

Preliminary idea

The major parts of the implementation are as follows:

1. **Building a 3D simulation environment (ARENA):** One of the tasks that the ROBOT has to perform in this project is to be aware of nutrients (yellow pyramids) and hurting entities (red cubes) that are

present in the ARENA. The ARENA scenario and the ROBOT vision are simulated by means of a 3D simulation tool (preferably using OpenGL). The visual information is gathered from the 3D surrounding scene by projecting it to a viewport with 3 colour channels, namely R (red), G (green) and B (blue).

2. **Capturing the viewport and preprocessing to identify the coloured objects:** To differentiate and spatially locate these entities, two specialized neural networks interpret the visual information received by the robot, one of them targeting nutrients and the other targeting hurting entities. To help in this task, a simple filter for the colours is implemented as a pre-processing stage using OpenCV.
3. **Using neural networks to interpret the visual information:** The ROBOT is an intelligent agent with a simulated visual sensor to pick images of the environment from its point of observation. The two neural networks inside the agent analyze these images classifying the visual patterns. These network outputs are tokens fed into the agent control system, which is an FSM.
4. **Motion control based on the visual information using FSM:** Based on the above codes, the FSM chooses an action from its repertory. The motion control is performed by the FSM automaton, which is designed without specifying how the interstate transitions should occur. It is set initially with a random structure that is improved based on evolutionary computation concepts. Only the maximum number of allowed FSM states is specified. The FSM is represented by a string of bits coding its states, inputs and actions. This string is named the ROBOT chromosome. For each FSM state, there is a chromosome section.
5. **Selection of the fittest to retrain the model:** At the start of the evolutionary process, a population of ROBOTS is generated by sampling chromosomes at random from a uniform distribution. This population constitutes the first generation. Each member of the current generation is put into the ARENA, and it is assigned a performance value proportional to the duration of its life. After each generation of robots in the ARENA, the fittest ones are selected to be the parents of the next generation and their chromosomes are grouped in pairs. These chromosomes thus undergo a mutation followed by a crossover of each pair. This procedure follows, giving thus more generations until there is a stabilization of the fitness value of the population.

Improvements

➔ For the ARENA, we will try to introduce more complexity in the environment, concerning its size, the distribution of objects, and the presence of textures.

➔ For the ROBOT, we can build new and more sophisticated characters and the kind of tasks it can perform (like adding another factor MONEY which will act in accordance with HEALTH). So the ROBOT will now have to look for a balanced amount between MONEY and HEALTH to extend its viability.

➔ Concerning the actual stage of the simulations, the ROBOT can have some improvements, namely to allow many more states in the FSM (thus increasing the number of bits in the input of FSM) and to provide a memory for a finite number of previous actions.