Donald Nute's VWorld

Bumble 2.0

Author:
James P. Spencer
Computer Science (Games)
Student Number: 08809130

Author:
Thomas J. Taylor
Computer Science (Games)
Student Number: 08813043

School of Computing, Engineering and Mathematics
University of Brighton

Documentation

April, 2012

Project Aim

You will be working on Donald Nute's V-World throughout the majority of the workshop programme, and will have soon completed workshop exercises designed to experiment with the world, its actors and its main (adventurer) agent "Bumble." This assignment is simple and extremely open-ended:

Ideas

Interesting behaviour could be characterised in many ways, but the following is indicative:

modify aspects of the program to make the behaviour of the actors (including Bumble) more interesting than it is at the end of the formal workshops

- The (demonstrated) ability of Bumble to deliberate about its situation
- The (demonstrated) ability of Bumble to react to changes in the world
- The ability of Bumble to demonstrate intelligent decisions about actions
- Actors in the world to demonstrate an enhanced level of intelligent behaviouretc.
- Proof of the success of your modifications could be:
- Behaviour based on a given set of scenarios - comparison with Bumble 1.1 & 1.2 say
- Performance measured against V-World's number of moves before (RIP) etc.

Deliverables

To be submitted: Thursday 10 May, 2012 by 8:45am

(a) The program - submission details to be arranged
(b) Documentation - submitted both in written & electronic form

This should be structured in the normal way for such documents, e.g. introduction, strategy, modifications made (with relevant code snippets), test plan, outcomes, further work, conclusions. It should support the program and be around 3000 words

Marking Criteria

- The system is functional according to the specifications, i.e. new behaviour demonstrated - it works: 20%
- Level of sophistication in the new behaviour(s): 30%
- The code is well-structured, documented and annotated (it is clear and accessible): 10%
- The documentation as specified: 40%

Introduction

At the beginning of the module, we were presented with Donald Knute's VWorld: a simple 2D sandbox game environment written in the Prolog logic programming language. The game world itself is represented as a 2D grid structure, and can consist of a number of different levels (or rooms) interconnected with doors. Each level is filled with a number of different objects, ranging from impassable walls, to collectibles such as health power ups (fruit) and weapons (a sword), to computer-controlled non-player characters (NPCs) such as snails and hornets. The main protagonist of the game is Bumble, an AI character who is dropped into the mysterious and very dangerous realm of VWorld. Bumble has a few basic stats which are updated as he roams the level: strength, which decreases as Bumble moves around the level and can be replenished with tasty apples, and damage, which is incremented if Bumble is attacked. Bumble dies if either his strength reaches 0, which means he is unable to move, or if his damage reaches 100%.

[list of game objects??]

There is no real aim to the game, or rather the aim is left for the programmer to determine for themselves. At the most basic level, the Bumble's main aim could be to merely traverse the world, and survive for as long as possible. We were given a number of test level maps, which are designed in such a way as to encourage certain behaviour to be implemented. For example, some of the test levels included the 'bird' character, who must be caught. Catching the bird requires that the player collects 'birdseed' which is located somewhere in the level, and so it can easily be seen that some sort of objectives may required for Bumble to succeed in such levels. Similarly, some levels have locked doors which require specific keys to enter.

To start us off, we were given a very basic Bumble, which we made improvements to as we progressed through the set tutorial sessions. At the end of the tutorials, Bumble's behaviour was functional, if very simplistic from an 'intelligence' point of view. By the end of the tutorials, Bumble was able to move around the game world, and build a 'map' of where certain objects were located as they were found.

Bumble also had a few basic rules to help with his survival in the dangerous world. These rules are used to ascertain whether Bumble is hungry or hurt based on his current strength and damage levels. This information is then used to point Bumble in the direction of any power-ups as appropriate. This behaviour was quite naive, as Bumble would only move towards power-ups in his immediate vicinity (i.e. less than 2 squares away). In addition to this, Bumble also collected any 'interesting' or 'desirable' objects that he was standing next to.

Aims and Objectives

The aim of this project was very open-ended:

> to modify aspects of [*VWorld*] to make the behaviour of the actors (including Bumble) more 'interesting' than it is at the end of the formal workshops

There were a number of different angles from which we could approach this problem. As it was, Bumble's behaviour was still very naive: he was only able to react to objects he was close to, despite the fact that a map was dynamically generated as Bumble explored the level. There was also no real 'method' behind Bumble's navigation, he simple wandered around randomly. Bumble also had no sense of any 'objectives' in the level. For example, he had no idea that in order to catch the bird, he first needed some birdseed. There were also a number of small bugs in the existing system which hindered his progress.

Our initial goal with this project was to simply increase Bumble's survival rate, as we felt that this was needed before any other aspects of the game could be improved. Moving on from this, we wanted to implement some sort of search algorithm which Bumble could use in combination with his dynamically generated map to find objects he had previously seen.

We also used the test levels provided to create a basic 'specification' of the kind of behaviour we wanted Bumble to exhibit:

- Some form of search
-

Modifications

There were a lot of small modifications which were made to improve the basic version of Bumble.

New Features

Search

Some random words...

Fuzzy-Inspired Reasoning

Didn't bother with the probability component, as it caused unpredictable and unrealistic behaviour.

Testing and Results

We have utilised the set of test maps supplied with bumble to evaluate the effectiveness of our solution. In this experimentation we work with the assumption that if bumble survives more than 5000 moves he will live infinitely.

???3 or 5 tests ?

Map	Test 1	Test 2	Test 3	Mean
Test 1	X	Y	Z	X+Y+Z/3

Table 1: This table shows some data

Analysis

What we found out from the results

Conclusions

Further Work

Machine Learning

We have both completed final year projects in this area, and this is something of great interest, we feel that this is very interesting area of AI...

Q-learning

Reinforcement Learning

Genetic Algorithms

Search Improvements

At current the search in the system is simple in nature, it was desired to have implemented A* but this has unfortunately proven to be problematic.

The current search could be improved by searching for the same goal once a search is erased .

Collision avoidance.

Naturalistic search.

Mapping Improvements

The mapping algorithm is simple in nature and could be improved by the use of?

Boundary Detection (IE) can i see a wall ? is the cell adjacent a wall ? therefore i do not need to visit that cell because it won't reveal anymore information to me.

The problem of mapping is strongly linked to Maze exploration, there are several interesting maze exploration algorithms such as (Azkaban algorithm, Dead-end filling, Wall follower) and we feel that the use of one of these would make the Bumble's discovery of areas more effectively.

Out of these of we feel that the wall follower algorithm is well suited to bumbles world would....

Talk about wall follower.

Appendix X: jfkdsgkjfdksljgklfdlkgjlkfd