Hunt the Wumpus: an Empirical Approach

Abstract

This paper discusses the use of Empirical Modelling in a Hunt the Wumpus game. Means of investigating the effectiveness of different inferences given different rules are discussed, as is the robustness of strategies if something unexpected occurs. Therefore, this is partly a study in Artificial Intelligence. The experimental approach taken with the model, however, suggests that it is also relevant to the area of Educational Technology, as the user of the model is encouraged to learn about the game and how to play it well by experimenting with strategies.

1 Introduction

In this paper, the area of Empirical Modelling is investigated in reference to a simple and classic game, Hunt the Wumpus. Some research has been done into modelling games from an EM perspective (**?**), and some models have been built that simulate games using EDEN, the Evaluator of Definitive Notations (**?**). The dependency-driven nature of EDEN allows us to define certain observables in a game in terms of other variables, instead of confining the game to a fixed number of set states. This means that the user of the model can perform operations on variables that might not have been contemplated by the programmer in order to investigate the results. This will be illustrated with a simple model of a Hunt the Wumpus game, a discussion of the dependencies the model contains, and the observables from which the player can infer useful facts.

2 Hunt the Wumpus

The game of Hunt the Wumpus was conceived by Gregory Yob circa 1973 (**?**). It is set in a cave of twenty interconnected rooms. Unlike other hide-and-seek games of its era, Wumpus (as the game is known) used a topological map — the vertices of a dodecahedron — as the play area, rather than a Cartesian grid. This means there is no concept of "up and down" or "north and south"; a room is simply adjacent to three others.

The player (who plays the role of an intrepid adventurer out to kill the wumpus) starts in a random room in the cave, and the wumpus starts in another. The adventurer has five arrows, and to win the game he must shoot one into the wumpus' room. If the adventurer enters the wumpus' room, he is eaten and loses the game. The adventurer also loses if he runs out of arrows without killing the wumpus.

The adventurer's task is further hampered by bottomless pits and giant bats. Walking into a room with a pit results in instant death. If the adventurer walks into a room containing a bat, then the bat picks up the adventurer and takes him to a random room, which may well contain either a pit or a wumpus, in which case the adventurer dies. The wumpus, having sucker feet and being too heavy for a bat to lift, is immune to both pits and bats.

The adventurer is given a number of clues about the locations of hazards. If the adventurer is within two moves of the wumpus, then the player is informed ("You smell a wumpus"). The adventurer is also told if any bats or pits are one move away ("You hear the rustling of bats" and "You feel a draught" respectively).

The wumpus does not move; it stays in the same room for the duration of the game. In some variations of Hunt the Wumpus, however, the creature does "wake up" and move into a random adjacent room (making lunch of any adventurer who happens to be standing in the way) if and when the player misses with an arrow. The player is thus discouraged from firing arrows speculatively. This variation will be explored in the paper.

3 Inferences

Many inferences can be made from what the player has experienced over the course of a game. These range from the obvious (the player has visited room 14 and is still alive, therefore the wumpus is not in room 14) to the more complex: say room 1 is adjacent to rooms 2, 5 and 6; the player has visited rooms

5 and 6, and is still alive; the player has visited room 1, and there is a pit nearby; therefore there is a pit in room 2. **?** discuss the AI aspect of Hunt the Wumpus in greater depth, and provide algorithms for calculating the probability of a certain hazard existing in a certain location. Here, we shall not be concerned with exact probability; all we are interested in is whether something is certain, impossible, or neither.

Room	Wumpus	Pit	Bat
1	maybe	maybe	maybe
2	no	no	yes
3	maybe	no	no
⋮	⋮	⋮	⋮
20	no	no	no

Table 1: The inference table.

Table 1 shows what has been inferred about the cave based on what the player has already seen. A "maybe" in a column indicates that no inference has been made about the presence or otherwise of the hazard for that room. Using this table, the user can make an informed decision about what move to make. Furthermore, by adding more definitions to the script, a more sophisticated set of rules can be used to give a stronger set of inferences. Therefore, what the model illustrates is a computer-assisted game of Hunt the Wumpus; one which can be used as a tool for teaching effective strategies in the game. **?** explores the idea of the computer as an artefact or instrument which can be exploited in this way. Such an idea could be applied to other strategy games such as chess, where the computer displays the set of possible moves along with an estimate of the likely result of each one, thus training the user to recognise which kinds of situations tend to be the most beneficial.

3.1 Some inference rules

We know that we detect the wumpus if it is at most two moves away. Furthermore, we know that we detect pits and bats if they are adjacent to our current location. Using these facts, we can construct a series of rules that can help us in the game. In EDEN, we can use triggered actions to recalculate the possibility of there being something dangerous in room n, in the event of some observable (such as the player's current location) changing.

3.1.1 Bats and pits

Pits can be detected from one room away, i.e. we can detect a pit from room x if and only if an adjacent room y contains a pit. The following simple rules apply:

1. Trivially, if we visit room x and find no pit there, that is evidence enough that there is no pit in room x. A similarly trivial rule applies if we find there is, but if that happens then the point is moot as we are too preoccupied with plummeting to the centre of the earth.

2. If room x is adjacent to a set of rooms A, and there exists at least one room y in A such that we cannot detect a pit from room y, then there is no pit in room x.

3. If x is adjacent to y, and we can detect a pit from room y, then if we know that every other room adjacent to y does *not* contain a pit (either because we have visited those rooms or by previous application of these inference rules), then x contains a pit.

4. If none of the above apply, then x may or may not contain a pit.

Since bats can also be detected from one room away, an identical set of rules applies to them as to pits, with the exception that we do not automatically lose the game if we visit a room that contains a bat.

3.1.2 Wumpi

In this paper, it will be assumed that there is only one wumpus. An extension of the model might investigate what effect two or more wumpi might have on the available inferences, but this model makes no such leap.

The rules for deducing the whereabouts of the wumpus are more complicated than those for pits and bats, as the wumpus can be detected from up to *two* moves away. For now, we will assume that the wumpus stays in the same room for the duration of the game. The model implements the following rules:

1. Trivially, if the player has visited room x and has lived to tell the tale, the wumpus is not in room x.

2. If there exists a room y which we know is within two moves of room x, and we cannot detect the wumpus from room y, then there is no wumpus in room x.

3. If the wumpus *can* be detected from room y, then every room x that we know is *not* reachable within two moves from y is wumpus-free.

4. If nineteen of the rooms definitely do not contain the wumpus, then the twentieth one does.

5. If none of the above apply, then the wumpus may or may not be in room x.

Note that at the start of the game, we may not know what the map layout is, so some exploration will be necessary before we can make the assumptions of rule 2 and rule 3 about rooms being reachable from other rooms in a certain number of moves. Indeed, rule 3 can only be used if we know we have complete knowledge of all the passages accessible within two moves from y.

4 Playing the game

Armed with our list of rules, and a model which simulates a game of Hunt the Wumpus and applies these rules based on the current situation and on what the player has experienced so far, we can experiment with the model to find out to what extent the inferences we can make from the rules help us. We can also find the limitations of certain rules, and means of altering the game so that they are no longer valid.

4.1 Dependencies and definitions

The Empirical Modelling tool EDEN provides a means of expressing dependencies, for example, the possibility of a wumpus being in room x depends on whether we have visited room x, whether we could detect a wumpus in any of the rooms near to x, and other observables. The text on the labels which show the inferences made, for example, depends directly on these observables; if the player moves to a new room from which no wumpus can be detected, the text on the labels is updated accordingly as an atomic action. This idea of "definitive programming" in which the text on a display is *defined* by a function of observables and experiences, is in contrast to traditional programming in which the responsibility for periodically checking that displayed information is up to date is placed upon the programmer. This means that the model user can perform operations that perhaps were not originally considered by the author, for instance, suddenly moving the wumpus to another room. If the user of the model moves the wumpus, then if the player notices this (i.e. because he can now smell a wumpus when previously he could not, or vice versa) then the inferences about the possibilities of the wumpus' location will change accordingly.

However, since the inferences discussed earlier rely not only on current observables but also on memory of what has already been experienced, performing such an action as moving the wumpus might cause some of the inferences to be wrong, as the inference rules assume that the wumpus will stay put.

4.2 Normal game

Consider a game which follows the classic rules as discussed in section 2. Consider also that the player is given a graphical view of the map and a tabular view of the inferences (as shown in table 1) being updated as necessary. From experimenting with the model in this way, it was found that the game is very easy to win. Starting in a room from which no wumpus can be detected instantly eliminates half of the rooms on the map as potential wumpus-harbourers. In a few moves, it is very easy for the inference rules to narrow it down to one room. It is then merely a matter of navigating to a room adjacent to that one and shooting into it to win the game.

4.2.1 Strategy

The "safest" strategy when playing the game as detailed above is to visit as many different rooms as possible, preferring to visit rooms which we have inferred have no hazards in them, until only one room can possibly contain the wumpus. This seems to work rather well as long as the player only visits rooms which definitely have no hazards in. This cannot always be done; if the player starts in a room that is adjacent to a pit, for example, the player has no knowledge of which of the adjacent rooms contains a pit, and so has to take a chance. On the whole, however, the experimentation with the model established that with both the map and inferences displayed, the game was very easy to win.

4.3 Starting with no map

One of the reasons why the game played with both a map and a table of inferences is so easy is that the entire graph is known. This means inferences based on the location of the wumpus can be made more easily, as there is always a definite "yes or no" answer to the question "is room x reachable in two moves from room y?" If we are not shown the map, and we make no assumptions about its layout other than that there are exactly twenty rooms, these inferences become more difficult to make. Yes, the player can smell the wumpus from room 4, but the player does not know whether room 8 is within two moves of there, and so cannot say for certain that the wumpus is or is not in room 8.

Playing the game without starting with the full map makes for a more realistic game, and also makes the problem more interesting from an Empirical Modelling perspective. We can now construe a "player" agent who has found himself in a cave of twenty rooms, and must build up the map as he goes along, noting down where bats, pits and wumpi were detected. Of course, the player is still helped substantially by the computer, which does most of the legwork in working out possible hazard locations based on the information available.

4.3.1 Strategy

The most effective strategy in this situation seems to be the same as when the player has a full map. Exploring as many rooms as possible that the inference table asserts are hazard-free is without risk and narrows down the number of possible wumpus-concealing atria.

4.4 Playing with no map or inferences

Technically, the presence or otherwise of a table of inferences does not affect the model's behaviour, or a player's ability to decide which rooms are safe to enter, because all the inferences the model makes are based on observables which are given to the human. However, if these inferences are not readily calculated and displayed for the player's convenience, then the player must make his own decisions about where the hazards might be, thus giving a more challenging game. An accomplished Wumpus player could try the game without the inferences in order to test his own ability to reason about the game.

5 Making the inferences fail

The inferences detailed earlier make a number of assumptions about the map and the capabilities of the wumpus. For example, it is optimistically assumed that bats, pits and the wumpus all stay in the same room for the duration of the game. The assumption that pits are static is, given a conventional construal of the problem, a valid one. However, we could follow the lead of some of the more adventurous variations of Hunt the Wumpus and introduce random earthquakes which change pit locations. Similarly, some versions of the game feature bat migrations, whereas in the conventional rules of the game, bats stay where they are. Many variations of the game, however, involve a wumpus which can under limited circumstances move to another room. In the event that things happen that are not permitted by the game's rules (and therefore that the inference rules assume will not happen) the strategy of "if possible, only enter rooms known to be safe" is less effective, as the inference rules might conclude a room is safe when it is not.

5.1 Moving the wumpus

One of the advantages of a dependency-driven language such as EDEN is that state such as the location of the wumpus, and the locations of bats and pits, can be changed by the user at will. Any variables that depend on these (such as being able to detect a pit) will be updated without the need for programmer intervention. Therefore, the user of the model can decide to move the wumpus from room 1 to room 20, and anything that depends on the location of the wumpus will automatically change. The inference table, however, relies on what has already been experienced as well as what is currently being experienced; for example, "the last time I was in room x, I could smell a wumpus". That is to say, it has a state. If we are in room y, and the wumpus moves such that it can no longer be detected from the distant room x, the inference table would still believe that the wumpus is within two moves of x. Therefore, while it is perfectly possible to move the wumpus to another room, and for all the observables that depend on the location of the wumpus to be updated seamlessly, we can make the inference table come to incorrect conclusions as it does not know that the wumpus can move.

The wumpus, while it is usually dormant, does in many variations of the game move from room to room in a restricted fashion. A popular implementation of an "agile wumpus" involves keeping the wumpus in the same room until the player fires an arrow. If this arrow was fired into the wrong room, then the wumpus "wakes up" and moves into an adjacent room. This, of course, would render useless all our inferences about the wumpus we have made, since our memory of where we could detect the wumpus refers to where it was *before* it moved.

The model includes an option to play this version of the game, in which the wumpus moves whenever the player misses. When playing this version of the game on the model, leaving the inference rules exactly as they were, interesting results are produced when the wumpus moves. For example, the inference table can conclude that there is no wumpus on the map. This is all well and good until the player saunters nonchalantly into room 12 and is eaten by something that doesn't exist.

To prevent such a calamity, let us construe the model in such a way that the player *knows* that the wumpus will move to another room whenever he fires an arrow and misses. We can then have the model forget about all the wumpus-related inferences it has made so far when an arrow misses. An possible extension to this would be to alter the model so that it only forgets those inferences that could have changed, given that the wumpus only moves to an adjacent room.

The Empirical Modelling perspective on this issue shows us what assumptions our inference rules make, and investigates the relationships between what the player knows about the wumpus' capabilities, and how our inferences can become invalid if the wumpus does things the player does not believe is consistent with the rules of the game. The model illustrates the difference between the game's rules permitting the wumpus to move, and the player *knowing* that the game's rules permit the wumpus to move.

5.2 Moving bats and pits

We can perform similar operations on the locations of pits and bats. Again, the inference rules given earlier assume that bats and pits do not move[1]. As with the wumpus, the model allows us to arbitrarily redefine the locations of the pits and bats; their locations are stored as two lists of room numbers. As before, observables that depend on the locations of these hazards are indivisibly linked to these variables by dependency definitions and triggered actions.

The earthquakes and bat migrations discussed earlier could be modelled by the user changing the locations of pits and bats. The inference rules assume that pits and bats are static, so the player could come to incorrect conclusions by relying on such inferences. To model a player who "knows" that pits and bats may change location, a triggered action could be added to the definitive script which forgets all previous inferences about pit locations when, for example, a pit is suddenly detected where there was no pit before. Of course, in a realistic model, this action would not simply be triggered by the alteration of the locations of the pits, since the player could be over the other side of the map and in no position to know that a pit has moved. More likely, the action would be triggered by unexpected changes in whether or not the player can detect a pit.

[1]a common misconception is that bats move when they take the player to another room. In fact, the bat returns to its original room as soon as it has dropped the player, so as far as the inference rules are concerned, the bat always stays in one place.

6 Conclusion

The uses of the Empirical Modelling concepts of definitive programming and dependency have been discussed in the context of a game of Hunt the Wumpus. We have met a set of inference rules that apply in such a game, from which the player can make informed decisions about which move (or shot) to make next. We have also seen how these inference rules can be broken if the agents in the game perform unexpected actions, and we have discussed means of fixing this problem by adding more inference rules that "inform" the player that these unexpected actions might take place.

What is contributed, then, is a study of how the rules for a rudimentary artificial intelligence can be constructed using dependencies and triggered actions in EDEN, and how such EM concepts are useful in constructing the game. We have also examined how the model of the game represents a player agent interacting with the world, and forming inferences based on his experiences.

References

W. M. Beynon. Empirical Modelling for Educational Technology. *Proc. Cognitive Technology '97*, pages 54–68, 1997.

W. M. Beynon and M. S. Joy. Computer Programming for Noughts-and-Crosses: New Frontiers. *Proc. PPIG'94*, January 1994.

B. P. Carr, J. L. Stansfield, and I. P. Goldstein. Wumpus Advisor 1: A First Implementation Program that Tutors Logical and Probabilistic Reasoning Skills. Technical Report AIM-381, MIT Artificial Intelligence Laboratory, October 1976.

G. Yob. Hunt the Wumpus: The Genesis of Wumpus. *The Best of Creative Computing*, 1:247–250, 1976. URL `http://www.atariarchives.org/bcc1/showpage.php?page=247`.

E. Yung. Eden: An Engine for Definitive Notations, September 1989.