Wumpus world report

DV2557
Fredrik Cronqvist
Martin Säll
2014-10-17

Intro

Wumpus world is a problem that have been addressed to in many years when describing some of the most common problems with AI. The assignment is simple, find and pick up the gold, but there is a problem, a monster is also living in the cave and we only have one shoot to get rid of it. There is also bottomless pits in the cave. The player have some indications to work with, breeze from the pits and some stench from the wumpus in their adjacent squares. With this information the player can safely move around in the cave and search for the gold but sometimes the gold is in a pit or at the wumpus.

Solution

Before trying to move the player we planned to know where to go next. By knowing where to go next it's easier to apply the rest of the rules for pits and the wumpus. To start with we create an overview for the AI of the world and set every square to be unsafe until we have proved anything else. To find out if a square is safe it need to be empty where we stand. When we find a breeze or stench we mark all unvisited side squares around the player with a possible danger depending on what it is. When going to a new square we update all side squares with the new information from where we stand. If a square have more than one breeze or stench connecting to it, we mark it as true danger and will ignore that square until the end when no other safe squares exists.

After most of the rules for taking out the next square to set as goal was done, moving the player was the clear next step. First we added logic to turn and move to a square next to the player and later used this method with our path list that contains a safe path to the chosen goal. If no safe moves exists or a path to a safe square is found, we check if we have the true position of the wumpus, if we do we kill him. If the wumpus is dead or not found and no safe ways was found we set the goal to the closest unvisited none confirmed danger prioritising a pit over the wumpus.

Conclusion

What we have done is far from the best but it works and solves the base scenarios and have around 75 to 85% success rate on the random maps with near to none freezes or crashes. This is mostly because we use a simple "if else" statement solution instead of any more advanced AI solution.

Solution on Github

https://github.com/kraiper/Tillampad-AI.git