Marist College

The Memory Game

Corley Herman

CMPT 404

Pablo Rivas

12/15/2016

Back in Game Development II, I worked with a partner on creating a memory card game. This gave me the idea for my final project. Rather than program the entire game of memory, including the player input, I just had the artificial intelligence match every possible pair and focused more on the memory code. In the long run, the project is to create an artificial intelligence that can pick two cards and make matches while dealing with a non-perfect memory.

Last semester, I created a memory game that set up the player against the computer, where the computer would remember each card flipped. The problem with this code comes in with the idea of how one's memory works and the computer's way of picking cards. During my experiments, I found the artificial intelligence processes didn't match that of an actual person very well. A person typically remembers cards better the more recently ago that they were flipped. The code provided in this example has it as a certain percent chance of storing the flipped card permanently in its memory. In other words, the memory was based off of a one time calculation, which is now how people remember things.

The approach taken in the final project focuses mostly on card values, rather than position. The game board is auto generated to create X matches, then randomly distribute them in random positions of a two-dimensional array. The array itself doesn't require a size, meaning that it needs a minimum of one row and one column for each row. This is because the array is used purely for visual representation due to the data being created and provided by the code, not the user. The artificial intelligence is passed an array of facedown card positions that relate to where they are in the game board. Each time a card is revealed, it is associated with a position. Now, as mentioned before, people slowly forget things over time. In order to match this idea, each time a card is revealed, the memory of other known cards degrade as well. To do this, each card contains an amount representing the degradation of itself in the AI's memories. After reaching a certain threshold, of which depends on the difficulty setting, the AI will forget the value of the card. With the basics down, the AI must follow a certain strategy. As it is not playing against anyone, its goal is to find all the matches in the least amount of turns possible. To do this, the AI checks if it knows any matches. If it does, it will a card that is in the first match it can remember; otherwise, it just picks at random from the cards it doesn't already know. It does the first pick this way because the it wants to eliminate any known matches before it forgets about it, but still prioritizes learning unknown cards if no matches are remembered. The second pick is determined based on the result of the first pick. If the value of the first card matches the value of a card it remembers seeing, it will complete the match; however, if it doesn't recall a match, it will pick a random, unknown card. The logic behind this pick is identical to the first pick; completing a match first, while still desiring to learn new values. While this, so far, has covered what the current project code provides, there are a few additional approaches that I wanted to bring up that I didn't have time to put into the code. The first of these include the separation of a cards value and position. What is meant here is that while you might forget which value is associated to which card, you can still remember which values have been revealed and the positions of cards previously flipped. The way this would go down is that the value, position, and connection between the two would each have a different degradation representation. This would allow for a better flow of memory, providing better grounds for learning and reasoning. In one instance, for example, the artificial intelligence would be able to recognize when an unknown card matches a known value, leading it to pick a known card for its second flip, in both hope for a match or the prevention of revealing an additional match that would potentially allow the opponent to gain an additional point. Another example would be, during the late game, when the potential to reveal a match for a known value for your opponent is higher, the AI would prioritize a card position it knows it was flipped, but no longer remembers the value associated with it. The other topic of these post-additions I wish to cover is the ability to play with more than just pairs. I have it set up so that it is possible to have multiple card positions tied to the same value, but I did not have time to implement to code to register matches more than a pair. The idea behind this is simple enough, mainly relying on the idea of informing the AI how many cards are required in a match.

In order to get data on how an actual person plays memory, I spent the course of a week finding people to play me in memory. While, in the end, I only managed to find a total of five people to play against me, I did manage to get around twenty games in. Each game was a set of nine pairs, coming to a total of eighteen cards. When I compiled the data gathered by the previous experiment and confirmed that the AI's memory was coded to work in a similar way, I amped up the game board from holding a couple matches to holding hundreds of matches, checking how many turns it took to find them all depending on the AI's memory level.

During the first set of experiments , I found that remembering the exact position of every value was not as easy as initially expected. For instance, one game ended with my opponent getting a total of zero matches, missing obvious connections. In the end, however, it seemed that, on average, it was common to forget which of the flipped cards had which values and which of the cards in an area was the card that you needed. The most prominent factors in the degradation of each players' memories included the time since the card was last flipped, flipped cards in the same region, the similarities of different numbers, and faith in one's own memory. For the first factor, the time since the card was flipped, each turn you don't find a match, the less you can focus on a specific card. Typically, the each new pair of cards flipped are more prevalent than the last. However, in a count to this, as more matches are made, it because easier to remember which cards are which due to the decrease of positions to remember. In the next factor, flipped cards in a region, time since flipped plays a role as well. It appears that, when both players flip cards in the same region, it is simpler to remember. I believe this is because there are no major gaps – that every thing is filled in. However, when you start jumping between regions, I found that the longer it took to get back to a region when a pair was found, the more likely the players will have either switched the values of two cards that were flipped in that region, or remember the general area the card was in. The third factor mentioned takes a play at pattern matching. The players see a card with the value of 6 and a card with the value of 8. Both 6 and 8 are similar to each other, so switching the values of the two cards can happen. This is more likely the closer the two values are to each other, which overlaps with the previous factor; when thinking about flipped cards a general area, it might be easy enough to distinguish that the 6 was on one side,by itself and the 8 on the other, but when they are in the same region, the clarity of the idea becomes more ambiguous. On the topic of clarity brings us to the final factor that takes a role in remembering – trust. Each person can trust their memory to a certain extent; the question is how much. This role is the most difficult to calculate without being the person themselves, since it requires knowledge of how well the person can memorize things. Trust is a concept built over time that leads to you either following your gut or second guessing yourself. The results from the second set of experiments isn't as in-depth. A summary of the results leads me to believe that the best result you can get is to set the memory tolerance to the number of matches that needed to be made. After that point, raising the tolerance did not change the number of turns before completion. As you drop from that point, the turn to make all the matches grows. The basic explanation behind this is that the amount of cards picked before a card value is forgotten grows shorter along with the memory tolerance.

This project has been a wild ride of decisions and changes. As the semester passed and the idea for this project grew, the end product changed from an AI for some kind of card game, to an AI for memory that can make more strategic decision, to what it is now; an AI that plays a game of memory like an actual human being. The only downside being that the final transformation of the idea came a little too late in the semester. Still, much was learnt from this project as the inner workings of how the computer would function, leaving a wider understanding of what is necessary when it comes to building something that has a form of artificial intelligence. Still, the data was gathered and parsed into a breakdown of the requirements needed to work and the key factors present when it comes to retaining the information provided.