Team 018
Bloomfield NM
Daniel Bachert
Craig Stahle
Tyler Burgett
Jacob Justice
Josh Palochek

Executive Summary

The project Team 18 chose this year was an indirect continuation of the project we did last year. But instead of trying to use the massive Quake II Artificial Intelligence module, we decided to take a more classic approach, so we chose Pac-Man. Our goal was to make a game that was similar to tag, where the human and computer players chase each other around in a maze and try to tag the other opponent. With time, the computer player would realize that the human player was more likely to choose to go this way, or that way and would create a more difficult challenge for the human player. To try to accomplish this, after some help from Gina Fisk, we decided to use statistics to teach the computer it would have a better chance if it would choose one path instead of another. We ran into some problems though. The week after we returned from getting the great reviews from the three judges, we found that our school had installed an Internet filter. We weren't able to access any of the pages that had our research materials until after 5:00 p.m. Seeing as how no one else's computers could handle being on long enough to download the source code and games, we had to lobby for the school to give us access. Eventually they did give us access but we lost at least three weeks of valuable time, and it put a huge dent in our coding process. All in all, we learned a lot about artificial intelligence, and we had fun doing it, if only there were enough hours in the day to write code.

Introduction

Our project was based on several Pacman clones, including Cap-Man, Grandpa Pac-man, a nameless Pac-Man clone, and of coarse the original. For those of you who have not had the joy of playing this arcade classic, it is a game of cat and mouse set in a two-dimensional maze, the object being to rack up points by eating the ghosts and eating all of the power-energy to move to the next level. The object of the game is to eat all of the little pellets, and munch as many ghosts as possible in the process, in advancing to the next level that gets progressively harder. While this is not a bad game, and only one man has beaten it, the artificial intelligence can leave something to be desired. Part of the time it seems that the little computer characters are running around in some circles just crying out to be eaten. We hoped to improve on this computer play.

To help us accomplish this task, the computer players need better algorithms to drive them about the maze. In short, we needed to come up with a way to help the computer remember where it had a better chance of catching Pac-Man, and so that we would not grow tired of chasing random computer ghosts.

Background

So how did this little yellow power pellet-munching monstrosity called Pac-Man come in to existence one might ask? Namco was creating its next large game and it was in need of a human player. The rest of the game had already been coded. Wondering long and hard about what a Pac-Man looked like, the creators had been stumped. One night Namco designer Tohru Iwatani, hungry as usual, went out for pizza to begin the long night ahead, and after reaching in and taking the first piece he was inspired, in fact it came to him like the cure for cancer. Circular pizza missing a slice is just what a Pac-Man should look like. In a day where flashy graphics alone can make a blockbuster game, it is comforting to know that Pac-Man is still the most purchased video game in history. Namco estimates that that the original (Pac-Man) arcade title has been played more than ten billion times in its 20-year history. Namco's total Pac-Man revenues have reached $100 million... one arcade quarter at a time. In 1982, there was a Pacman cartoon, and even a spoofed song by Jerry Buckner and Gary Garcia song called "Pac-Man Fever" (sung to the tune of Cat scratch fever). In July 1999, Florida resident and die-hard Pac-Man fan Billy Mitchell achieved the first perfect score in Pac-Man (3,333,360) after playing for six hours straight. He beat all 256 screens eating every dot, fruit, and ghost (all four ghosts were eaten with each power pellet) - using only one Pac-Man!

Description of project

As stated earlier, we chose Pac-Man for this year's project, in order to make coding easier than last years Quake II project. What we wanted to do was make the game's Artificial Intelligence more advanced than the original. Not that we think the game is to easy, seeing as how only one person has beaten it and that was only two years ago, but the game only gets faster not smarter. Our goal was to give an old classic the benefits of modern AI programming, such as making an Apple II E a supercomputer. When we began to look at this project we noticed that we would need a way to make the computer keep a record of all of the successful attempts of capturing the human player, and create a way to know where the human player is. To take care of the problem of navigation, we decided that an X/Y coordinate plane would suffice for precise character movement. In our original plan we had decided that we could have the computer keep a log of all of the coordinates where the human player had been captured and form a table of the most common player movements, then decide on the best route to take. During project evaluation at San Juan College Gina Fisk gave us a better idea. Having the computer log all of the "capture points" would cause tremendous lag, so Gina suggested that we use statistics to have the computer make more logical decisions. Shortly after returning from the evaluation we were gung-ho about getting the project completed, but there was a slight problem. By law the school is required to have an Internet filter, which was installed just at the time we needed to look at game code. We found all of our chosen research sites blocked. After several weeks of begging the school to let us get into the sources we needed we finally gained access to the school has we found most of our material that we needed to complete our project. We were able to find a cheap Pac-clone that was open source that had quite a bit of useable code. We then came to a standstill about the statistical logging, so we emailed Mike and Gina Fisk. [Info in parentheses is pending, if we get this to work this weekend, it will remain, if not then we will change it] (After finding some possible algorithms that could best accommodate our project we had to go in and place all statistical storing information back into the code.)

Conclusion

Our group has learned many things about not only programming, but teamwork as well. The artificial intelligence program that we designed allowed us to make the computer learn and become smarter. The algorithms that we used defined player and computer movement to at definite points. We built the maze on a X/Y axis that allowed us pinpoint precise locations as opposed to estimated locations. By using percentages and statistical logging we can store data and the super computer can use it to decide the best possible route by using percentages. However, the programming has not been the only factor in making this year's AiS challenge result a success. Teamwork made it possible to complete every part in of our project. Our team has come a long way in development and showing results. We came together and made a program that runs: one of our main goals for this year. Through teamwork, collaboration with mentors, and long hours of hard work, our aspirations for this year's AiS challenge have been met to an extent that was not thought to be possible. We have exceeded our video game dreams.