**PacManDBZ World and Mind Description**

In the world I developed I decided to use Points to keep track of all elements on the grid. I found Vectors very useful for storing Points. When creating the walls for the maze I designed a grid of points. I then inserted these points into the Vector. This process can be seen in the genWalls() method. Printing the maze was then easy, I simply looped through the vector and for any Point that existed in the Vector I printed a wall at that location. The Vector which held the walls was also very useful when ensuring a character could not enter a cell which is a wall. I designed a Boolean method valid position which would check whether a provided Point is within the wall Vector. This method was then used in validLeft , validRight,validUp and validDown. These methods would check whether a point a character wants to move to is valid. The move method then uses the previously mentioned methods to ensure they don’t run into a wall. Unfortunately the move method is the downfall of my Ai. If a character tries to move to a walled cell the move method decides to return a valid random action. This can sometimes lead to the character getting stuck going back and forward between the same two points over and over again. This random method does however perform quite well a lot of the time and frees a character.

At the start of the game all of the characters are designated a random position. The initPos() method ensures that none of the characters start at the same position. It also ensures the dragonball is generated at a unoccupied cell.

I designed a freeCells() method which returns a Vector of all the free cells on the map. This could easily have been used to place food on the game screen and have a character collect it but I wanted to do things a little differently.

I decided to have four characters in my game. This led to a lot of confusion when writing the addImage() method. It meant a lot of checks and the code isn’t very pretty to look at but I think it made the game a little bit more entertaining.

The takeAction method was also a little bit complicated as I had to account for the values each character had along with how they would be affected in certain scenarios. I wanted to have a character disappear if my good guy(Goku) managed to collect a dragonball. I found the easiest way to do this was give each bad guy one life and take that life away if Goku had already collected the dragonball. The bad guys position was then set to (0,0) and he was no longer shown on the screen. I also made Goku forget about those bad guys by checking if they had a life remaining. They were moved to position (0,0) a walled position so they would no longer affect gameplay. The checks for when a bad guy came into contact with goku while he hadn’t collected the dragonball is a little simpler. Just take a life away from Goku and re-initialize all the characters positions. If gokus life drops below three the game ends. An Integer was used to check if the dragonball had been collected. Once this integer is set then Gokus image changes and he becomes super saiyan. When in this form Goku can’t be killed but his enemies can easily be.

If I were to make some improvements to my game I would improve the scoring system. At the moment it is a little bit confusing so perhaps instead of having Goku invincible when he collects the dragon ball I could just keep a count of how many dragonballs he manages to collect and return that as the score. He could lose a dragonball if an enemy were to catch him. This implementation wouldn’t be too difficult but I ran out of time.

Another flaw in my game (I believe) is that all player movements are controlled from the mind. I’m not sure this is exactly what you were looking for. A better implementation may have all of the enemies controlled by the world. I could then pass enemy locations to the mind and make Goku avoid them. The mind I implemented is simple enough. All of the enemies are provided with Gokus location. They then try to move closer to him. They will choose horizontally or vertically depending on which way is closer to Gokus position. Goku in this mind ignores his enemies and makes his way to the dragon ball(another mind could easily have him avoid enemies) when the dragonball is collected(The mind knows this from an integer value that is passed to it i.e. int dragonBallCollected) Goku changes from chasing the gragonball to chasing down his enemies. I thought it would be impressive to see the character change from doing one thing to another.

Unfortunately my game runs much better offline then on the w2mind server. On my computer the characters take one turn at a time and it shows the gameplay in a more understandable manner. On the server some images seem to disappear and it looks like the game has errors but this wasn’t the case on my own computer. If you would like to see some of the runs I performed on my own laptop I would be happy to send them to you.