1 Getting Started

1.1 Platform and Tools used
This program is developed in Windows XP Professional with SP3, using Eclipse Helios (version 3.6) and JDK 6 (Update 21). We test our intelligent agent with Firefox (version 3.6.8) in Windows XP (Color quality: 32-bit).

1.2 Start the Game Emulator Using Web Browser
Launch the web browser and go to http://www.webpacman.com/ to start the game.

1.3 Start the Intelligent Controller
To start the intelligent controller, double click the runnable JAR file (bruce-2010.jar). Of course, run the following command will produce the same result also: java pacman.MsPacInterface

The intelligent controller will find the location of the game screen automatically by searching the first row of the game map. In order to know if the coordinate of the game map has been detected correctly, check the controller output window. If it looks like figure 3, that means the game screen has been located successfully.

Please be noted that the game screen’s search process will perform once only at the start of our program. Further change in the location of the game screen is unnoticeable by our controller. So if the web browser has been resized, or moved to a new place, please restart the controller again to search for the updated location. Also, the preprocessed map information (bitmap1.txt, bitmap3.txt, bitmap6.txt) should be placed in the same folder otherwise the program cannot be started properly.

In case the location of the controller window or color settings is incorrect, please modify this file: src/pacman/Settings.java
Figure 1. The controller will search the first row to detect the location of the game screen.

Figure 2. Searching the game screen.
2 Methodology of the Intelligent Controller

Our intelligent controller has pre-processed the map and stored the path, walls and initial pills location as external text files. The controller reads the files after the game image has been found and represents the game map using Graph structure. The shortest distance and path between coordinates are calculated in real-time. Our intelligent controller uses breadth-first-search to look for the nearest pill but least close to the ghosts and will chase and capture the ghosts if they are edible and close enough. If the ghosts are close to the Ms. Pac-Man, she will try to escape from them by selecting a suitable path. The path is determined by performing thousands of Monte-Carlo simulations in real-time. The simulation will return the survival rate and rewards (e.g. the number of pills and ghosts captured). In order to acquire accurate simulations, we have modeled the behavior of the ghosts. For instance, Blinky often chase towards Ms. Pac-Man, Pinky tends to go ahead towards Ms. Pac-Man’s next position, Inky likes to follow Ms. Pac-Man’s previous moves while Sue usually select a path randomly. We have spent most time in developing the Monte-Carlo module. The controller can often select a suitable path to prevent it being trapped by the ghosts and at the same time capture more pills and edible ghosts.

The recent score in our final testing on Intel Core i7 860 2.8GHz, 4GB Memory, Windows XP Professional SP3 is ranged from 5,000 to 14,000 (usually game over in stage 2 or 3).