COMPUTER SYSTEMS RESEARCH
Running Version of Your Program 2007-2008

1. Your name: _____Brett Jones_____, Period: __7__
2. Date of this version of your program: ___10/30/07___
3. Project title: __First-Person PacMan__
4. Describe specifically what files are needed and the command(s) necessary to run your program.

    ```
    How to run your demo:
    Open the Game.java file in jGRASP and use jGRASP's run utility. Java 1.5 and
    Java3D 1.5.1 are required for the program to run.
    ```

5. Your program is running, now what? List test input(s) for the user to interact with your program. Specifically what should the user expect to happen?

    ```
    Input(s):
    The program's main menu accepts mouse input when the buttons are clicked.  Once
    a game is running, the program currently accepts keyboard input from the Escape
    key and the left and right arrow keys.

    Program's expected response(s):
    In the main menu, the mouse clicks should activate the buttons.  Currently, the
    only buttons that actually do anything are the New Game and Quit buttons.  All
    buttons, however, should show the click animation (that looks like the button
    is actually being pressed) when the user clicks them.  The Quit button exits
    the program, and the New Game button puts the program into fullscreen exclusive
    mode (FSEM), where it currently will display a white screen with a gray box
    where the menu was (this is the default; the code doesn't actually display
    anything on the screen).
    Once in the game, the keyboard keys Escape, left arrow, and right arrow hide
    the game and return to the main menu, turn left, and turn right, respectively.
    The latter two are at this time unnoticeable.
    ```

6. What about user input errors? Are there incorrect user input(s) that your program handles?
    The program does not handle user input errors per se, but it does handle errors with non-existent map files or map files that don't fit the specifications. The program requires a map with a numerical filename (for the level it's to be played on) with a .txt extension. If this criteria isn't met, the createScene method that the map input code is in will throw an error back to the initialize method (both of the World class), and the initialize method has a try-catch block that will report that the map file does not exist and terminate the program.

7. What is the programming doing, demonstrating, or analyzing? What is the user looking for in order to understand what you've been studying and developing with this project?
    The program is intended to demonstrate that a ray tracing algorithm can render a game, simplistic as it may be, in real time. The user should see no tearing, flickering, or lag/stuttering (signs that the computer isn't generating frames of the game fast enough), and should see a relatively realistic view.

8. How has your program evolved so far?
    So far, the program has evolved from nothing (it was started from scratch), to a program that simply displayed a menu of non-functional buttons, to a program that can enter FSEM and leave on command.

9. By the end of this school year, what do you hope to have as a final version of your program in relation to this current version? What will you demonstrate during your final presentation?

For the end of the year, the program should be a 3D first-person recreation of the classic PacMan arcade game. I intend on using ray tracing, so a discussion of the algorithm will be included in the presentation, as well as a video of a sample run of the game. The program will also include a basic AI to control the ghosts in the game, and will run in FSEM.