Black Hole

Computer Operating Environments – CE0813A
Elinor Townsend - 0900164

Black Hole is a Windows game written in Visual C++. Threading of the main game loop is done with the Boost threading libraries, using version 1.46.1 of Boost.

The game was created and tested on a Sony Vaio laptop with an AMD Phenom II triple-core processor 1.80GZ and 4GB RAM. It has a 64-bit operating system and runs Windows 7.

Compile and run using Visual Studio 2008.

Controls for game play are Enter to progress through scenes, and the black hole is moved using the arrow keys.

The main game loop methods are in the playscene.cpp file. The functions which are threaded are UpdateLightParticles() and UpdateBlackhole(). The threads are created in the set up method of the cPlayScene class, and a barrier is used to sync threads at their beginning and end inside the Update() method, which is called every loop.

In order to measure the performance timings are taken for the threads to run each loop and stored in a global variable, time_taken in the cBlackhole class (blackhole.cpp). An exit condition of 5000 light particles created (10000 iterations) was used for measuring performance by adding a break point on line 33 of sundestroyed.cpp and checking the value of time_taken at that point.

The number of threads used can be varied by changing the value of the noofthreads integer in blackhole.cpp at line 7. Currently it will work with values between 2 and 12. This can be increased by changing the number of array elements for m_lightparticles[] in playscene.h, line 31.

To test the program with no threading, comment out lines 54-62 in playscene.cpp, where the threads are created in the setup() method. Also comment out lines 95-96 in the update() method, lines 153-155 and 163-164 in the UpdateBlackhole() method and lines 173-175 and 203-204 in the UpdateLightParticles() method to remove the barrier checks. Lines 84-93 in the update() method of playscene.cpp can then be commented in to call the methods normally.