In this programming assignment, you will use the grid-based, texture-mapped environment that you developed in your second assignment to insert an animated character that employs \( A^* \) pathfinding to reach a sequence of goal objects. If you would prefer to use the instructor’s programmed environment, its code, enhanced to include a simplistic, non-pathfinding character, is on the course Web site:


Your assignment is to add the following features to the environment:

- An animated character that follows the underlying grid in the environment to reach specific goal objects that appear on the playing field. Be creative in the design of your character, using texture maps or moving parts (e.g., legs, wheels, arms, wings) or both. The character’s movements, both in changing position and in changing direction, should look smooth and natural. Pay particular attention to coordinating the orientation of the character and its features to the direction in which it is currently moving. Note that the texture-mapped spheroid character in the provided code is much too simplistic (and much too large), but that it is always facing the direction in which it is moving.

- A series of goal objects that appear at randomly selected grid locations (always on actual grid points, never within one of the craters). Each object should arise from below the surface of the playing field, and only one object should appear on screen at any given moment. The objects should be designed to correspond to the identity of the character. For instance, if your character is a mouse, the goal objects could be hunks of cheese; if the character is a tank, the goal objects could be small buildings. Once the character reaches a goal object, a reaction between the character and the object should ensue (e.g., the mouse consumes the cheese, the tank crushes the building). Don’t get too elaborate with this effect, but make the reaction clearly visible and consistent with the identities of the character and its goal objects.

- An \( A^* \) pathfinding algorithm that uses Euclidean distance as its heuristic. The character must follow the grid path dictated by this algorithm to reach the current designated goal object without entering one of the crater obstacles. When displaying the grid, the current \( A^* \) path that the character is following should be highlighted by having the grid points and grid edges of that path colored differently from the rest of the grid.

All of your code should be modular and well-documented, with clear comments indicating your additions or alterations to the original program and reflecting the purpose of every major programming component. Constants should be declared in the provided header file and undeclared constants should never be used.

Zip-compress your entire project file and copy it to your drop-box by 10:30 AM on April 22, 2008. Late assignments are not accepted without verifiable medical documentation. You must write your own code, and no one but the instructor may see your code.