Final Programming Assignment

The Wonderful World of Shakey

Due 03/16/2023 23:59:59 hrs. (Midnight)

Information

We close our journey of C++ programming by writing a program to direct an imaginary robot, named Shakey, through some simple tasks. Shakey lives on a fixed size grid, surrounded on all sides by a high fence. Besides Shakey himself, there are two other types of objects that can occupy his world. The first type of object is a wall. Walls are used to represent obstacles that Shakey must maneuver around, such as hurdles, mountains, enclosed rooms, and mazes. The fence along the perimeter is also composed of walls.

The second type of object in Shakey's world is an item. Shakey is equipped with a mechanical arm that he can use to pick up and put down items. Some of Shakey's tasks involve picking up or putting down items, or finding and transporting items. Whenever we want Shakey to carry out a task in his world, we must supply him with a program, that is, a detailed set of instructions that describe how to perform the task. Shakey can receive, memorize, and execute such a program, assuming that it is written in a language he can understand.

History

Much of this material on Shakey was inspired by *Karel the Robot*. The definitive source for information about Karel is the book *Karel the Robot, a gentle introduction to the art of programming*, by Richard E. Pattis, second edition with revisions by Jim Roberts and Mark Stehlik, published by John Wiley and Sons, 1995.

There is also an object-oriented version called *Karel++*. Karel++ is a C++ like language specifically designed to control robots. Karel and the Karel++ language was the pre-basis research for the Mars and Lunar robots programs.

The Assignment

In this assignment Shakey will not be required to memorize commands. We will enhance Shakey's intelligence in the Advance C++ session. Remember Shakey is not a Gundam, Giant Robot or a Transformer; please do not equip Shakey with missiles, gun cannons, projectile weapons, or lasers. Shakey is a peaceful robot and will not be challenged by the evil dominions of the universe. Shakey does not fly or hover. Shakey is simple robot about the size of R2D2 whom is dependent upon your commands to move about within the enclosed fenced area.

Shakey understands the following six commands:

1. **step:** When Shakey executes a step instruction, he moves forward one square on his grid, continuing to face in the same direction as before. Shakey's program will complain if he is ever directed to step into a wall. He may occupy the same square as an item. He cannot occupy the same square where there is a wall. The walls are impenetrable immovable objects. Shakey cannot break through walls.
2. **left:** Shakey executes a left instruction by pivoting 90° to the left, remaining on the same square.
3. **right:** Shakey executes a right instruction by pivoting 90° to the right, remaining on the same square.
4. **pickup:** Shakey picks up an item from the square he is standing on. If he executes a pickup instruction and the square he is standing on does not contain an item, then his program should issue a warning.
5. **drop:** Shakey places one item on the currently occupied square. If he is not carrying any items, or if the current square already contains an item, then his program issue a warning.
6. **home:** Shakey moves directly to his home position. He does not pass go... he does not collect 200 dollars. (Ordinarily, Shakey's home position is in the upper left hand corner of his grid, facing north. (0,0))

There is no limit to the number of items that Shakey can carry at one time. Unless otherwise specified, Shakey is working as a mailman and the objects he drops and picks up could be letters.

These six commands constitute the core of Shakey's language. Later, you will see how to extend the language by using these six commands as building blocks. Because they form the foundation of the language, they are sometimes called *primitives*. When a Shakey program is ran, it will accept a sequence of Shakey commands; he performs each command, one after the other, in the order given. You do have to actually type each program command yourself.

Each command for Shakey should be a method call. Here is the syntax of each Shakey command:

shakey.*command-name* ();

The first part of the command, shakey., indicates that the command or message is being addressed to Shakey the Robot. We could conceivably have several robots running around, all with different names. So, when we give a particular command, we must refer to the robot we are addressing by name. The dot is used to separate the robot name from the command given to the robot. The trailing punctuation after *command-name* is required, just as the period must appear at the end of an English sentence.

For example, the following sequence of instructions could be part of a Shakey program.

```
shakey.step();
shakey.left();
shakey.drop();
```

Provide a prompt to the user to issue the basic commands; all command should be lower cased. When the user enters a command the corresponding method should be called.