Assignment 3: HearthStats

Write a report (pdf or word) in which you answer the questions below. For each question:

- 1. repeat the question you are answering;
- 2. explain your approach;
- 3. describe your answer(s);
- 4. show (relevant) code snippets;
- 5. show the output of your program.

Once your report is finished, make sure your names and student numbers are on the title page, and send it to your instructor before the deadline.

Note that you are required to program in C# or C++, and must work in pairs of two. Assignments are graded with a G (good), V (sufficient) or O (insufficient).

The following state machine describes the possible states and actions for an NPC in the game HearthStats:



As you can see in the diagram, this NPC robs bank in order to increase his wealth. He continues to rob banks until he either gets rich (and starts having a good time), or spots a cop (and starts fleeing until he feels safe or gets tired).

Question 1)

Add two new state transitions to the diagram above. Name these transitions and draw out the new diagram.

Question 2)

Implement your new state machine using switch statements, a state transition table or the state design pattern. Build it in such a way that the user can change the NPC's state by writing a string (get rich, spot cop, etc.) in the Console.

Question 3)

Add output to your program by showing a line of text that indicates the current state or action. For example, while he is robbing banks he may say:

"I'm robbing banks and getting loads of money! Pew pew!"

For the other states and actions you can use the following lines:

"I'm rich enough to have a good time" (for the 'get rich' action) "I'm having a good time spending my money" (for the HavingGoodTime state) "I see a cop, so I have to start running" "I'm getting very tired, so I better lay low for a while" etc.

Or make up your own lines.

We can make the state machine more interesting by making the state transitions depend on events and values of variables.

Question 4)

Add variables to your NPC for wealth, distanceToCop and strenght. Wealth increases while the NPC is robbing banks, but decreases while he is having a good time or fleeing. DistanceToCop can suddenly change to 0 during robbing a bank or having a good time, which causes the NPC to start fleeing. Strenght decreases during robbery, having a good time and fleeing, but increases during laying low. Continue this way and make sure that each state transition depends on one or more of these variables, and draw out your new diagram.

Question 5)

Implement the variables and conditions state transitions that you chose in your NPC. Add a loop to the program, such state it updates the variables and looks at the current state at each iteration; it then determines whether a state transition should occur. Also implement a one second pause at each iteration to your program. You can now remove the user input function, as the new states is set only by the variable's values which are updated at each iteration. Adjust your program so that this works and show the output of an example run.

This loop could go on indefinitely. We are going to add a Cop NPC to the HearthStats game that can catch the first NPC, such that the program can end.

Question 6)

Design a state diagram for a second NPC (Cop) with three possible states: OffDuty, OnStakeOut and Chasing. Design logical state transitions that depend on the variable 'dutyTime': the value of this variable increases during OnStakeOut and Chasing until it reaches a certain value, at which point the state transitions to OffDuty, when it starts decreasing again until 0. The Cop initially starts in the OnStakeOut state. When the former NPC starts robbing banks, the Cop will at some point start Chasing. The first NPC then starts fleeing, which changes the distanceToCop value. Draw out the full Cop state diagram and design logical transitions between states such that he is able to catch the first NPC and the program terminates.

Question 7)

Implement the Cop NPC in your existing program loop. Make sure the Cop also outputs a line indicating its current state or actions, for example:

"Duty time's over, I'm headed home" "Relaxing on the sofa, watching Crime Scene Investigation" "Hold it right there buddy" etc.

Question 8)

Show the output of an example run that terminates by itself, and showing alternating lines said by both NPC's.