CSCI3100 Software Engineering

Tutorial 2

17:30pm, 22nd Feb, 2021 (Mon.)/

17:30pm, 24th Feb, 2021 (Wed.)

Please read the questions before the tutorial.

Answer the following problems based on lecture Topics 4 notes.

1. Finite State Machine (2020 / HW2 / Q3)

As we learned in the course, an FSM can be used to represent and control an execution flow. Therefore, it is perfect to model the decision-making for a character in a game, i.e., a game AI. In this problem, we are going to use FSM to design a game AI.

In a classic fighting game, a monster usually patrols around until a game player appears within its sight. When it sees the player, it quickly chases the player to get the player into its attack range. If the player falls into the monster's attack range, the monster starts its attacking until the player escapes from its attack range. Note that the monster will keep chasing and try to attack the player unless the player is out of sight, in which case the monster will resume its patrol state.

Question:

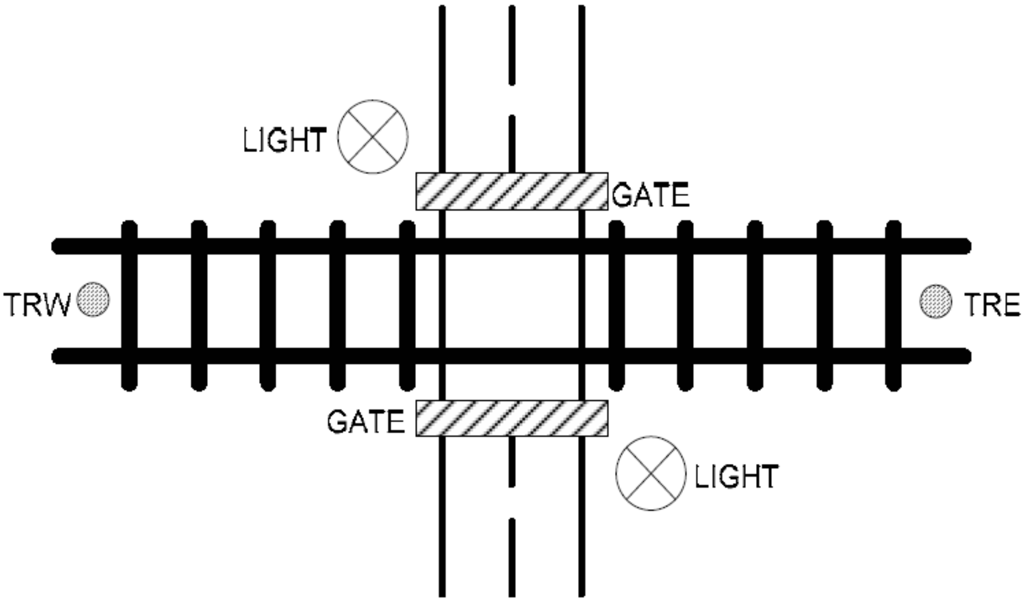
(1) Draw an FSM of the monster in a fighting game according to the description above.

(2) In a real fighting scenario, we need to consider the HP (Health Point) of the monster. When the monster fights with a player in its attack state, its HP value drops. If its HP value drops under the threshold ***t***, the monster will turn to escape and stays in this state until its HP increases to ***t***. Then the monster chooses different actions according to the distance between itself and the player.

After introducing the new escape state, how will you design the FSM?

(3) In (2), we introduce only one new state. By comparing two FSMs and specifications in (1) and (2), please discuss the advantages and disadvantages of using FSM to define the actions of game AI.

2. Finite State Machine (2018 / Final / Part 2: Q1)



There are two types of Finite State Machine. A Moore-type FSM is one whose output values are determined only by its current state. A Mealy-type FSM, on the other hand, is one whose output values are determined both by its current state and by the values of its inputs.

Consider the street/train track intersection shown above, where trains may be traveling eastbound or westbound. A Moore-type finite state machine is needed to control the crossing gate (GATE) as well as the warning lights (LIGHT) as output values, based on input values from two train sensors, TRE and TRW. The sensor values will be asserted (i.e., set to 1) if and only if a train is immediately passing over them. The gates will be closed if and only if GATE is asserted (set to 1), and the warning lights will be on if and only if LIGHT is asserted (set to 1).

Safety considerations dictate that your finite state machine should behave as follows: When no trains are detected, the gates should remain open with the warning lights off. Whenever a train is detected on either sensor, the gates should close and the warning lights should be ***flashing*** (i.e., set to 0 and 1 alternatively) until the opposite sensor is first asserted (set to 1) and then de-asserted (set to 0). Only then can the gates open and the warning lights turn off. For simplicity, you may assume that the two sensors will never be asserted (set to 1) simultaneously, and therefore your design may handle the input combination TRE*TRW arbitrarily.

3. Specification by Data Flow Diagram (2014 / Final / Part 2: Q1)

Alice and Bob, students of CSCI3100, are working on their project. Alice is in charge of the security of the project website. She is concerned with the news that NSA has hacked into a lot of websites and stolen the username password combination of millions of users. She wants to make sure that their site is secure so that even if NSA hacked into their site, they still cannot recover and steal the passwords. So she consulted Bob, who is a security expert. The followings were their conversation.

Alice: Hi Bob, do you know that NSA is spying on a lot of websites? I heard that they recovered a lot of passwords! That's awful. I use the same password for all my websites!

Bob: Yes. Those websites are irresponsible for their users. They store a user's password as md5 hash with no salt. There exists a million ways to recover an unsalted password!

Alice: But why? I was taught that security hash function such as md5 should be very hard to reverse.

Bob: Yes that's true theoretically if you use a completely random long password. But most passwords we use are words that can be found in a dictionary. NSA can pre-compute the md5 of all the words and simple combinations of all the words. Then given an md5, they can look up the table to find the corresponding password. This is called dictionary attack. Even worse, they can pre-compute the md5 hash for all the possible passwords up to certain length, for example 8. Then all possible passwords of length under 8 can be recovered instantly by looking up the pre-computed table!

Alice: Hmm, I see. I want to make sure our website is secure. I heard you mention password salt. What is that? Can you talk more about that?

Bob: Sure. In fact the idea of salted password is quite simple. When a registrant is registering for the website, he/she provides the username and password. Our site should generate a long random string called salt. Then the password and the salt are exclusive-ored (XOR operation) to get a salted password. Apply md5 on this salted password and you get the hashed value. Then register function store the hashed value as well as the username in the login database, store the salt and the username in the salt database.

Alice: But why would this salt help preventing NSA from getting the original password?

Bob: The salt is long and randomly generated. Also it should be different for every user. When NSA wants to brute force attack md5 hash value of the salted password, without the salt, they have to try all possible combinations as long as the salt. Using a salt of 2048 bit for example, would require millions of years of computation, rendering the brute force hack impossible! Even if they hacked into the salt database, they still need to brute force attack all possible passwords and this cannot be pre-computed. This makes a properly salted password very hard to recover.

Alice: That sounds great! But how do you authenticate the user?

Bob: That's easy! When the user is trying to login, he or she should provide the username-password combination. You retrieve the user's salt and hash value from salt database and login database respectively using the username. You XOR the salt with the password to get the salted password. Then apply md5 hash to get the hash value and compare it with the hash value retrieved. Display the results to the screen then you can authenticate the user.

Alice: Thanks for the explanation! I'm going to implement the most secure website in the world!

Based on their conversation, draw the Data Flow Diagram of the registration and authentication process described.

4. Specification by Data Flow Diagram (2020 / HW / Q1)

During the flu season, Jackson's Hospital is always filled with patients. He decides to use MediCare, an online pharmacy system, so that patients who have been confirmed diagnosis of flu by a doctor can receive medicines that have been prepared by the doctor. English description of MediCare is as follows:

> In MediCare, a patient should first enter his or her username and password correctly to log in the system. The verification procedure is conducted via a user account database. If there is no matching information, the log in operation will be rejected. Otherwise, the system will search through diagnosis records for user diagnosis given approved user information, and return a medicine list to an order process function. Inventory details with be returned by a dispensary database with the medicine list. After that, an order is generated from the order process function, with medicine list and inventory situation as its input. Then the patient should pay for the medicine through a payment function by entering the credit card information to the payment function. By taking the order and checking the credit card information through a credit account database, the payment function will display the transaction result. Meanwhile, the dispensary database will be updated given the consumed medicine information.

Question: According to the above description, draw a Data Flow Diagram (DFD) to describe the operation of MediCare system. Note the above English description may contain ambiguity. Your DFD can be made more specific and clearer, as long as it meets the English description.

5. Specification by UML Activity Diagram (2020 / HW / Q2)

In this problem, we are going to use UML to specify the workflow of hiring an overseas applicant in the company BetterHK^{TM}. Please draw a UML activity diagram according to the following description of the hiring process for BetterHK^{TM}.

a. An HR goes through the CV of an applicant;

b. If the applicant is qualified, HR arranges an interview for the applicant. If the applicant is not qualified, the whole procedure ends.
c. There are two types of interviews according to the preference of the applicant, which includes either an onsite interview or a remote interview. If the interview is onsite, HR books a meeting room to conduct interview. If the interview is a remote interview, an online interview system is activated.
d. Then an interview is conducted. During the interview, HR talks with the candidate to know his/her skills.
e. If the applicant successfully passes the interview, the HR prepares official documents for the on-boarding employee. At the same time, the overseas employee should obtain the working visa. The whole recruitment procedure finishes after both documents are obtained.
f. If the applicant fails the interview, the HR simply terminates the process.

Again the above English description may contain ambiguity. Your UML activity diagram can be made more specific and clearer, as long as it meets the English description.