NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER EXAMINATION FOR
Semester 1 AY2007/2008

CS4344 Networked and Mobile Gaming

November 2007 Time Allowed 2 hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains SIX (6) questions and comprises FIVE (5) printed pages, including this page.
2. Answer **ALL** questions. State your assumptions, if any, clearly.
3. The total marks for this paper is SIXTY (60).
4. Answer all questions in the answer book provided.
5. This is an **OPEN BOOK** examination.

1. (5 points) In a game running on centralized client-server architecture, unfairness arises due to variable network latency between the clients and the server. To compensate for such unfairness, it is proposed that players experiencing higher average response time be given some advantages through changing the rules of the game (for instance, by making their character stronger or faster).

   Do you think this is a good idea? Why or why not?

2. (5 points) In Voronoi Overlay Network, give an example (draw a diagram) to show that it is possible for a node A to be a boundary neighbor of a node B, but B is not a boundary neighbor of A (i.e., the boundary neighbor relationship is not mutual).

3. (5 points) Is redundant data bundling for thin streams necessary for movement updates in a first person shooting game? Justify your answer.

4. (15 points) An issue that needs to be addressed when using dead reckoning is the value of the distance error threshold τ, i.e., the threshold of the distance between actual position and predicted position above which an update needs to be sent. When two players are close to each other in the game world, accuracy is important and therefore the threshold should be small. For two players that are further apart, the threshold can be larger.

   Now consider how such adaptive dead reckoning scheme can be used in the following scenario. We have a centralized client/server architecture, with a server s, and clients (i.e., players) p_0, p_1, ..., p_n. We may assume that the latency between the clients and the server is negligible. Let $d(i, j)$ be the distance between player p_i and p_j in the game world, and $\tau(i, j)$ be the error threshold between p_i and p_j. The game designers have determined that the following relationship between error threshold and players distance is suitable for their game.

$$\tau(i, j) = 0.2 * d(i, j) \tag{1}$$

   A player always updates the server s whenever its velocity changes. When s receives an update from a player p_i, s uses adaptive dead reckoning to decide whether to forward this update to another player p_j ($i \neq j$) based on $\tau(p_i, p_j)$ and the current prediction error of p_i's position at p_j.

   (a) (5 points) Explain how the adaptive dead reckoning approach above can lead to huge computational overhead at s.

   (b) (5 points) Suggest how the computation of error threshold can be modified to reduce the computational overhead at s, while still being adaptive to distance between players.

   (c) (5 points) In the scenario above, should a player use dead reckoning between itself and the server? Justify your answer.

5. (15 points) Consider a first person shooting game, taking place inside a game world consisting with rooms and corridors (represented as *cells*). We would like to run this game using a fully decentralized, peer-to-peer architecture. Instead of exchanging updates between every pair of players, we would like to restrict exchange of updates to exactly those players whose cells are visible to each other. In other words, if two cells c_1 and c_2 are visible to each other, then a player P_1 in c_1 and a player P_2 in c_2 will exchange updates with each other. Conversely, if two cells c_1 and c_2 are not visible to each other, then P_1 and P_2 should never exchange updates.

   (a) (4 points) Explain why Frontier Sets cannot be used here, by giving an example where two players might still exchange updates even though the cells they belong to are not visible to each other.

   (b) (7 points) One way to implement the above scheme is for a player to actively query for the other players it should exchange updates with. Explain how you can use distributed hash table (DHT) to support this operation in an efficient and scalable manner. Your design should try to reduce the number of DHT updates when players move across cells, as well as the number of DHT lookup operations for each query. Give the relevant details of your design, including what is hashed, what is stored in the DHT, when the DHT is updated, and how your DHT can be used to find all players in the visible cells. Explain why you pick your particular design.

   (c) (4 points) Suppose a player can intercept packets sent to him/her and examine its content. Explain how the player can cheat using this scheme.

6. (15 points) To support seamless game world in an MMORPG game, the following approach, called replicated zoned servers, is proposed. The game world is organized into regions and the regions are mapped onto servers. Each server is responsible for maintaining the primary copy of states within a region, as well as replica of states in neighboring regions (two regions are neighbors if they share a border). Note that regions are not necessary contiguous.
   (a) (5 points) What are the pros and cons of this approach, compared to mirrored servers architecture and zoned servers architecture?
   (b) (5 points) Describe how states should be synchronized among the clients and servers using the replicated zoned servers architecture.
   (c) (5 points) How should load balancing mechanism for replicated zoned servers be different from the one you learnt in class for normal zoned servers?

END OF PAPER