NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER EXAMINATION FOR
Semester 1 AY2008/2009

CS4344 Networked and Mobile Gaming

November 2008 Time Allowed 2 hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains FOUR (4) questions and comprises THREE (3) printed pages, including this page.
2. Answer **ALL** questions. State your assumptions, if any, clearly.
3. The total marks for this paper is EIGHTY (80).
4. Answer all questions in the answer book provided.
5. This is an **OPEN BOOK** examination.

1. (20 points) Maintaining state consistency in real-time, interactive, multi-player networked games is difficult. These games, fortunately, tolerate some amount of inconsistency by nature. As such, game designers often sacrifice consistency of game states in order to improve other design aspects of games.

   For each of the items below, describe a technique that

   (a) (5 points) sacrifices consistency to increase responsiveness

   (b) (5 points) sacrifices consistency to reduce number of messages sent

   (c) (5 points) sacrifices consistency to reduce visual disruption

   (d) (5 points) sacrifices consistency to reduce power consumption

   Include in your description how each technique can lead to inconsistency.

2. (20 points) Consider a client/server implementation of two player Pong game similar to your Assignment 1.

   (a) (10 points) Someone suggested that we apply redundant data bundling (RDB) for TCP to packets exchanged between the clients and the server.

   Is this useful? If yes, describe the type of packets (its content and purposes) for which RDB is useful. Otherwise, argue why RDB is not necessary for your implementation of Pong.

   (b) (10 points) The networking library ENet provides optional reliability – when sending a packet, the application can indicate with a boolean flag whether the packet should be delivered reliably or not.

   Someone suggested that we always turn the reliability flag off (i.e., send unreliably) when sending paddle position updates in Pong. Explain why this can lead to undesirable result and suggest how can you use the reliability flag wisely to avoid it.

   (Note that sending all paddle position updates reliably is not a good solution).

3. (20 points) A game map in a first person shooting game consists of two areas: (i) an outdoor area that is wide open with little occlusion, and (ii), an indoor area consisting of many rooms and corridors. A mobility trace of the players reveal that players tend to move around at high speed indoor. Further, each player tends to visit almost all the rooms indoor within a short time (perhaps to search for enemies or items). On the other hand, the players tend to be less mobile outdoor (perhaps hiding and waiting to ambush other players). The game uses visibility-based interest management.

   (a) (10 points) Consider implementing this game using a peer-to-peer architecture.

   Would such mobility pattern support the use of frontier sets to reduce the number of messages exchanged between players? Justify your answer.

   (b) (10 points) Suppose we want to implement this game using a zoned server architecture. We are interested in dividing the game map into zones and assigning each zone to a server for state management. Players can still move between zones seamlessly.

   i. Based on the mobility pattern observed above, give an argument *against* sub-dividing the outdoor area into zones.
   ii. Based on the mobility pattern observed above, give an argument *against* sub-dividing the indoor area into zones.

4. (20 points) According to the Voronoi Overlay Network (VON) scheme proposed by Hu et al., a peer maintains direct connections with its AoI neighbors (all peers within its AoI). When a peer generates an update, it unicasts the updates to all its AoI neighbors. This method is not scalable when the number of AoI neighbors becomes large. To improve scalability, we want to explore the use of application-level multicast to disseminate updates to AoI neighbors.

   One of the fundamental questions in application-level multicast is how to organize the receivers into a tree. *Without resorting to using DHT, and by using the existing Voronoi structure*, describe how the AoI neighbors of a peer can be organized into a multicast tree. In your description, include what messages are being exchanged, and how you ensure that the resulting structure is a tree (depending on your solution, the latter can be either trivial or tricky).

   Assume that the players stay static (i.e., no one moves, joins, or leaves) when the tree is constructed, and assume that the tree can be reconstructed from scratch every time the Voronoi structure changes.

END OF PAPER