NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER EXAMINATION FOR
Semester 1 AY2006/2007

CS4344 Networked and Mobile Gaming

December 2006 Time Allowed 2 hours

INSTRUCTIONS TO CANDIDATES

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

1. (15 points) **Mobile Gaming**

   (a) (5 points) List any FIVE challenges in developing mobile networked games.

   (b) (4 points) Describe any two benefits of using key polling instead of key interrupt to track user interaction in a MIDP game.

   (c) (6 points) Why Bluetooth clients communicate with the Bluetooth server before establishing the connection? Describe the processes required in establishing a Bluetooth connection in both server side and client side.

2. (30 points) **Network Architecture and Interest Management**

Consider the following network architecture for an MMORPG. The system consists of many servers, organized into a k-ary tree of height n. A player/client can connect to one or more servers in the tree (See Figure 1). A host can only communicate directly with other hosts it is connected to.

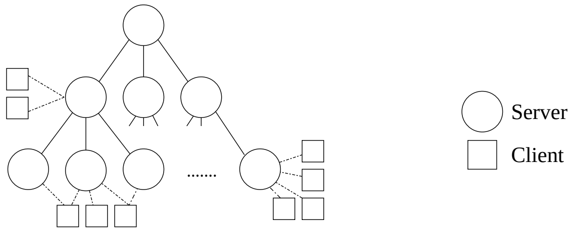

Figure 1: An example depicting a system of servers organized as a tree with $k = 3$ and $n = 3$.

The game world is partitioned into uniformly size rectangular cells. Each server is responsible for maintaining the states of one or more cells. The states of a cell is replicated among n servers, forming a path from the root to a leaf server in the tree. The root server maintains the states of the whole game world (i.e., all cells). The set of cells maintained by a server is always a subset of the cells maintained by its parent.

Answer the following questions. Note that the questions are inter-related, so you should read *all* the questions and think carefully about the answers before you answer anyone of them.

(a) (8 points) After partitioning the world into cells, the system needs to decide which cells should be maintained by which server. **Give two desirable properties of the set of cells assigned to a server. Why are the property desirable?**

(b) (8 points) When a player joins the game, his/her game client needs to connect to one or more servers. **Give two factors that should be considered in deciding which server or servers a client should connect to? Justify your answer.**

(c) (4 points) **Give a game scenario (e.g. capability or action of an avatar) where it is advantageous for a client to connect to a non-leaf server.**

(d) (4 points) Suppose a player updates its own state, causing the game client to send an update message. **Which other servers/clients should eventually receive this update message?**

(e) (3 points) **Is it better to organize the servers into a short and fat tree (large k, small n) or a tall and thin tree (small k, large n)? Justify your answer.**

(f) (3 points) Someone suggested that we should add connections among servers at the same level in the tree. **Will this improve a player's gaming experience? Why? or Why not?**

3. (15 points) **Dead Reckoning**

Dead reckoning is a technique used when a player A wants to predict the location of an object controlled by another player B. The method consists of two parts: *prediction*, which computes the location of the object when a position update from B is not available, and *convergence*, which smoothly moves the object to its proper location when new position (and velocity) is received.

(a) (9 points) Consider the following four variations of dead reckoning.

- V_P: perform prediction only but no convergence (object is teleported to the new location immediately);
- V_C: perform convergence only but no prediction (object remains static until update is received, after which the object moves smoothly towards accurate position);
- V_{PC}: perform both prediction and convergence.
- V_ϕ: perform no prediction and no convergence, i.e., not using dead reckoning (object remains static until update is received, after which the object teleports to new location immediately);

We then evaluate the accuracy of the methods by summing up the difference between the object position as perceived by A and the actual position at B in a *first person shooting* game where the object moves and turns quickly and frequently.

Arrange the four methods above according to the accuracy (most accurate first). Does you answer depend on the network delay between A and B? Justify your answer.

(b) (6 points) Instead of dead reckoning, someone proposed a history-based prediction method. When update from B is not available, player A extrapolates the next position of B, by looking at the two last positions of the object. Based on these two positions and the time it takes for the object to move from one position to the other, the velocity (speed and direction) of the object can be computed, and therefore the next position of the object can be predicted.

What are the advantages and disadvantages of this history-based prediction method?

4. (20 points) **Cheating**

Consider a location-aware, multi-player, mobile game played on a mobile device with GPS. The players are divided into two teams, and are physically distributed in different locations in a city. The city is logically divided into squares. The game does not reveal the square a player is in to any other players (include those on the same team) until the player is eliminated.

The objective of the game is for a team to "bomb" the players of the opposite team. Team takes turn to bomb each other. Bombing is done by a designated player from each team, through SMS-ing the target square ID to all other players. Any player whose position is inside the bombed square when the "SMS bomb" is received will be eliminated from the game. Note that "friendly fire" is possible, i.e., a player may eliminate players from the same team accidentally.

A team loses when all its players are eliminated.

(a) (5 points) **Describe a scheme where players of the above game can cheat *by collusion.***

(b) (5 points) **Describe a scheme where players of the above game can cheat *by exploiting misplaced trust.***

(c) (10 points) The game above is played without a game server. **Would introducing a centralized, authoritative game server into the game mitigate one of the above cheats? If your answer is yes, describe how cheating can be mitigated. If your answer is no, describe how players can still cheat with a server.**

5. (20 points) Some first person shooter (FPS) games support two types of players: (i) *shooters*, who can move around in the game world with a weapon and attack other shooters, and (ii) *spectators*, who can assume the viewpoint of any shooter currently in the game. A spectator can only watch the game passively and not allowed to interact with the game world. For this question, we are interested in an FPS game that can support a small number of shooters but potentially huge number (in the order of hundreds or thousands) spectators.

   Recall that FPS games typically employ a client/server architecture with the centralized server as the authority for determining game states. **Describe how the network architecture and synchronization protocols can be augmented to support large number of spectators in a FPS game.**

END OF PAPER