Programming Assignment 6: Lobby

Due Date

- Assignment due on Finals Week (Regular scheduled Final's Class)
  - Grading Sunrise Next Day
  - ----- ABSOLUTELY NO EXTENSIONS ----------
- Submit all files and directories to Perforce
  - Create a directory called: Omega in your student directory
  - PA5-PA8 will be in the same directory
    - /student/<yourname>/<Game>/…
    - You will identify the discrete submissions in your readme file
      - Please remember to add descriptive check-in comments
  - Fill out the *PA6 Lobby - Submission Readme.pdf*
    - Describes the summary of work for this PA6
    - ***Changelist numbers and dates*** associated to the assignment
      - ***This needs to be there or NO CREDIT***

Goals

- Add a Lobby to the Omega Race game

Assignments

1. Add a lobby to the Omega Race game.
   a. Use InvitesSample, ClientServer, PeerToPeer and *NGSM* (my favorite) for references
   b. They are on perforce/references/sample or on App Hub website
2. Omega Race X Game Example:
   a. Linked to another Omega Race using PeerToPeer or Client Server networking topology.
   b. Add a lobby to Omega Race on the interface screen. (Has to be this screen or enhance this screen, no XNA sample screens)

c. Dialog button can be added anywhere on the screen.
    i. If you are crazy ambitious, you can create your own screen, (not required)

3. Screen modifications
    a. You will need to investigate and create new text, fonts, symbols to get this screen to behave like a REAL lobby. Start reverse engineering the screen.
4. Background
    a. Everyone is required to use ***System Link***
        i. System Link is 100% easier if you have access to 2 machines on the same LAN.
    b. XBox Live - can be *optionally added* (but must also support System Link)
        i. You will need two machines (PC) with two Live (DreamSpark) subscriptions.
            1. This can work on one machine, but is difficult due to the two instances and multiple logins.
        ii. You need two logins because invites work only with LIVE PlayerMatch sessions, unlike other networking features that can be tested locally using system link. If you don't have two subscriptions, the easiest way to try this out is to get a friend to help, or ask for assistance. Hopefully this phase is working for you, ask our forum for help.
    c. System Link
        i. You just need two machines on the same Local Area Network (LAN). Make sure you create local profiles.
        ii. Follow, live, breath the NGSM Sample demo. It's all there

5. To Join a Session
    a. To join a session through your lobby using XBox Live (optional)
        i. Run your custom Omega Race on two machines
            1. Find your friend - using LIVE profiles that are friends.
        ii. Press **A** to create a session on the first machine. There are three ways the second machine can join this session:
            1. Using regular matchmaking, the second player can press **B** to search and join the session that was created by the first.
                a. Easiest Way – my suggestion
            2. The first profile (which is hosting the session) can go to their friends list, select their friend on the second machine, and choose the **Invite to Game** option. An invite notification will appear on the second machine. If the second profile presses their Guide button and accepts this invite message, they will automatically be joined into the session. This can be thought of as a "pull mode invite," because the first profile sends a message to pull the second into their session.

3. The player on the second machine can press the Guide button, go to their friends list, select the first profile (who is hosting the session), and choose the **Join Session In Progress** option. This is a "push mode invite," because the second player pushes themselves into the session without any direct involvement from the first.

b. To join a session through your lobby using System Link (required)
   i. Make sure each machine has a local user
      1. You can create a local user through the menu system
         a. Start NGSM sample and add user.
      2. Make sure each local user is enable and logged on
   ii. Connecting
      1. Have one machine create a session
      2. Have the other Find a session
         a. Select and join

6. Once both Systems are joined in a session.
   a. Print to the screen:
      i. CONNECTED
7. Now you are 1/2 of the way there
   a. You need to cycle through a game
      i. Create Session --> Join session --> Start Game --> play game -> Return to Lobby
   b. Make sure you can cycle
      i. NGSM Sample has a very good framework to understand and reverse
8. Fill out the *PA6 Lobby - Submission Readme.pdf*
   a. Describes the summary of work for this PA6
      i. Quick step by step
      ii. How you accomplished this task (engineering perspective)
   b. Describe any issues you had in completing this task.
      i. What was your design/debugging process.
      ii. Describes the summary of work for this PA6
   c. Changelist numbers and dates associated to the assignment
9. Obviously the next step is to start linking and staging the game, but for this assignment. It's just establishing the connection successfully.
   a. The next assignment is to connect the game and move the player B remotely on the host game.

Validation

Simple check list to make sure that everything is checked in correctly

- Program compiles and runs without crashing?
- Program warning free?
- Can you successfully connect to another machine?
- Did you write your pdf file?

Hints

Most assignments will have hints in a section like this.

- Baby steps, use an very incremental process
    - Big steps will prevent you from finishing task
- Study the InvitesSample OR *NGSM sample* (trust me)
- Look at the documentation
    - Start menu -> XNA Game Studio -> XNA Game Studio Documentation
    - Programming guide is very useful.
    - Look around
- Since you are going to modify software on 2 PCs frequently
    - Create a share directory between them.
        - Have the host PC A hold the lead copy of the program.
        - Do not edit on PC B
            - You modify there, and push it to PC B.
            - That PC is only used for debugging and button presses.
    - I would also add version number on the screen window
        - Easy visual to insure same code drop
    - Use perforce for your syncing between machines
        - Open a client on each machine
        - Submit on one machine, get on the other.

Troubleshooting

- Baby steps
    - You'll be in trouble if you don't
- This is so slow and painful, takes forever to get working.
    - You cannot escape the agony of this part
    - Just do it.