

There is now widespread ownership of basic data communications hardware, such as low-cost modems (and in particular we are about to witness the advent of cheap, mass-produced personal computers with modems and modular jacks built in, such as the Amiga and maybe the Phillips). There is also a growing base of home computers of reasonable cpu power, memory size and disk capacity (such as the Macintosh, the IBM PC-AT, and newer machines yet to hit the market -- Amiga, Mindset, Phillips). The coupling of these two facts suggests a number of interesting opportunities for multi-player computer games conducted over telephone lines.

We are particularly interested in the potential for using these machines' considerable processing capability to fashion a loosely coupled system. The idea here is that we do not wish to maintain continuous communications between one game player's local computer and another, or between several players' machines and some centralized host, as has often been the case traditionally. In such a situation the player's machine is usually acting as nothing more than a glorified (or even unglorified) terminal. This is disadvantageous for several reasons. First, most of the processing power of the player's machine is simply wasted. Second, communications costs are still comparatively high. Maintaining a connection for several hours at a stretch can run up a pretty big phone bill, which in turn can discourage potential customers. Third, a centralized host, if one is used, must be capable of supporting a great number of simultaneous users and have large numbers of I/O ports with which to talk to them. This contributes to the host's operating costs and ultimately, in the case of a commercial system, to the connect-time charge that the user gets socked with.

Instead of maintaining continuous contact between two machines, where the local machine simply passes the user's input on to the remote machine, we can use the local machine's own capabilities directly. To do this, we arrange to have all of the interactive virtuality of our game provided by the local computer. It displays data quickly and graphically. It accepts the user's input and figures out what it means. As it does this it stores up commands and queries for the remote system. According to some scheme (like a schedule, or perhaps just as needed), it calls up the remote system by itself and exchanges a brief flurry of highly compressed, abstract packets of data.

Consider a game like "Empire". In many ways each player is independent. If each started the game on his own machine, but all players shared some common, mutually agreed upon random number seed to create the world, then each could play along with only occasional contact between machines to synchronize their world states, conduct combat, transmit diplomatic communications, etc.

With the above thoughts in mind, there are two obvious paths to follow. Either or both of these may be worth pursuing:

Path 1 -- The Lucasfilm Games Alliance

A master computer (the non-evil variety) would maintain a huge (1000+ players) database for games. This central database would represent a large simulated universe in which all the players would be participants. The game universe would be updated by players whenever they logged in. The central computer would do the database manipulation, and simply accept data from and transmit data to the players' local computers, without letting them actually run programs on it (i.e., the host would not be a general-purpose time-sharing system).

While many possible settings, themes and game concepts may spring readily to mind for such a system, one such particularly appeals to me. The flavor of the game is rather science-fictiony (?) and the idea is somewhat grandiose.

The scheme is to have a large (i.e., apparently unbounded) universe with worlds, star systems, alien races, different types of technology, and well-defined physical laws. Players would buy a program in a local store to run on their home computer. Each copy of this program would have its own serial number to provide each player with a unique identity from the moment he or she first boots it up. All the home machines running the program would communicate with the single large centralized host (like a VAX) over the phone or Telenet or whatever's cheapest and most convenient. When the player starts the game, he finds himself in much the same situation as one does in "Empire": in an expanse of unowned wilderness, in a relatively primitive state of development and with some standard amount of starting resources. He would then expand from there, developing his industrial base, moving over the world and ultimately out into space, encountering other players and generally having a grand old time.

The difference between this game and "Empire" is that it would support a much larger number of players (thousands) in a much larger and more diverse and open-ended universe. The universe would have its own abstract existence independent of the players. The system administrators (i.e., *us*) could liven the game up ala D&D by placing alien artifacts, non-player characters and races, inhabited but unclaimed worlds, and so forth in the universe for the players to discover and interact with. In addition, since we would be devoting the player's whole home machine to information display and control functions, the game would be more interactively fluid and more visually and aurally stimulating than "Empire".

The difference between this game and similar services offered by CompuServe and its ilk is that we would be dedicating our system to the game. We would not be offering a general purpose, multiple-service system. We would be communicating with computers running software of our own devising rather than with terminals. The level of interaction that the host would have to support would be much more distributed. As described above, players' computers would correspond with the host in relatively infrequent (i.e., a few times a day) high information density burst exchanges. Since the average connect-time per player would be quite low, the number of players logged in at any given time would only be a small number in comparison to the total number in the game. This means that the number of incoming communications lines required and host I/O and CPU bandwidth required would be substantially less than that needed to support a conventional terminal-interactive game of similar scope. This in turn means that the cost to establish and support the host system could be substantially less than the cost of a conventional system. The level of support for dial-ups (i.e., the number of ports) would be correspondingly less and the host system could be optimized for handling the game. This all means that the level of hardware resources and consequent support overhead could be less, user connect time would be much less (per user), and we would therefore (I hope) be able to charge the players a LOT less than CompuServe or its brethren (I think we'd produce a more interesting game than they would too!).

In summary, for the particular game I have in mind:

- The universe should be open-ended. It should not have any apparent physical boundaries, inherent resource limitations, or time limits. The overall game would not have any defined winners or losers or explicit goals for the players. Rather, as in Empire, the goals would be provided by the players themselves. Perhaps subsets of the game could be established that had victory conditions for those involved but which still took place in the same universe and required interaction with possibly non-participating players (rather like a scavenger hunt or a road rally in our mundane world).

- The scope of the game should be large. A large number of players helps

make the universe more interesting. A large universe provides room for a variety of experiences. The open-ended time scale enables players to develop their characters/worlds/personae as much as they would like. One consequence of the open-ended time scale is that players would be dropping out of the game and new players joining all the time. A large physical universe allows new players to be placed in remote locations so that they have time to develop and encounter similar neophytes before bumping into the experienced players who have developed very powerful intergalactic empires.

-- The cost to play should be low. I'm sick of on-line services that cost several dollars an hour. I certainly couldn't afford to get casually involved in something requiring that degree of financial commitment, and I'm sure I'm not unique in this regard. If the game is something that will appeal to only the most fanatical and devoted on-line system junkies, then the market is probably too small to support it. Keeping the cost down is one way to encourage wider participation.

-- It should permit people with varying time commitments to play and enjoy it. This is another aspect of keeping the audience as broad as possible. Obviously someone who is interesting in killing a few minutes every so often is looking for something different than the person who is willing to commit several hours a week. The game could be set up to let someone who wanted to spend 20 minutes to, say, fly a reconnaissance mission or defend a planet from an invasion fleet, but let someone who is willing to spend 10 hours start building an interstellar trading empire.

-- It should have the SF/interstellar theme discussed above. There are two reasons for this. The first is that the interstellar theme is particularly suited to the open-ended structure demanded. The second is that I'm currently the person spec'ing this game, and it's what I want to do.

-- It should take advantage of distributed processing on home machines. This has been discussed extensively above.

-- It should permit different levels of interest and ability. Ideally such a game should be playable at a variety of levels depending upon player inclination and sophistication: as a zoom-thru-space shoot'em-up, as a game of exploration, as an economic simulation, as a diplomacy game; as a serious effort or as a light-hearted romp; as an occasional weekend diversion or as a full-time obsession.

Path 2 -- Lucasnet Games

Lucasnet Games would be a series of games, strongly related in internal structure, loosely related in external (visible) structure, unrelated and unique in theme. These would be multiplayer games set up for people to run on their own (multiple) home computers.

Unlike the centralized, large scale proposal above, these games would be smaller in scope, involving 2 to 20 players and coordinated and administrated by the players themselves. We would write the software to make it all happen but the players would be responsible for finding each other and establishing communications and play.

The games would be played in phases:

Phase one: Order entry phase. Player enters new orders for his units. Units are ordered to move, fight, explore, build, destroy, change. Extensive user-friendly graphics and forecasts of the effects of the moves are provided on the screen.

Phase two: Order transmission. All players exchange orders, either by calling one player, or circulating them in a chain until everyone has everyone else's orders.

Phase three: Update. Each computer updates the state of the game universe, based deterministically on orders of units and a shared random seed. Update is depicted graphically, with the same displays as in order entry but with final results affecting the database. Public key encryption and various copy protection tricks are used to avoid cheating. Winner is announced if one or more players have achieved victory conditions. If not, play proceeds back to phase one.

Possible themes:

Historical battles (actual and hypothetical)
Exploration (of America, of the Solar System, of our Galactic Arm)
Politics (elections, running countries, revolutions)
Economic (running countries, running businesses, world trade, lemonade stands, interstellar trade)

All of the above (PSL's Empire is a good example)

Program Structure:

The programs would have three parts to them:

1. A standard 68000 "game" with data base and programs to manipulate it, with "black box" interfaces to the other parts of the program,
2. An I/O-graphics package, computer specific, that displays the game information colorfully (or, in Mac, shadefully) and clearly,
3. A communications interface that standardizes the inter-machine communications (so any of the target machines can play with any other, and part 1 can be machine independent).

Game Structure:

Games would be played in the aforementioned phases, with three very general types of play (regardless of theme):

1. "Automatic" games where the player sets up some high level goals or "doctrines", e.g., a military doctrine of shoot first, ask questions later, an exploration doctrine of send in a few military, and if unopposed send in civilians, otherwise more military. Updates will take care of actual movement and interactions. ROBOTWAR is an example of this kind of game, although with only one round. Political multiplayer team games with each player being a cabinet minister (or co-dictator) and setting up a doctrine independently, then submitting them together and playing several teams against each other are another possibility. You could win as a team, or as an individual (based on what the computer-generated populace thought of your policies).
2. "Semi-Automatic", which can range from automatic games with a few special orders that can effect the database without an update, through games with internal economics and politics under your control, but with interactions between players happening only at updates and according to preplanned doctrines, to games like

Flying Buffalo's STARWEB, with almost everything being ordered explicitly, and only a few special situations (ambush, retreats, conditional attacks) being mediated by the computer during update. An example of this sort of game would be one of interstellar exploration. Each player would have real-time control over their home planet, building and exploring their home system, sending out colony ships and explorers and warships on missions. If the stars in this world are a minimum of four light years apart, the updates would take place every four years. At that time any new missions would be transferred between players' computers, with the computers keeping the information secret until the time that ships arrive at the opponent's home planets, probably during the off-line phase. These missions would be controlled by the computer, according to whatever doctrine the player sending the ship out had determined at time of launch. Speed of light would limit communications as well as transport-- if an explorer went to a system 6 lightyears away, it would take 12 years (game years of course) to find out if there were any planets there. Once a colony has been established and reaches a certain technology level, they can build an ansible, an instantaneous communications device that works only between planets. Then the colony would be controlled by the player when off-line, and new doctrines could be established instantly. Changing technology would make a great difference in this game, and situations like Haldeman's "The Forever War" would easily happen.

3. "Manual" games, where the game mechanics are simple and players can proceed with full control over what happens, exchanging information at updates but resolving attacks, building, etc. independently. An example of this sort of game is SLAG, by Stewart Eastman for the TRS-80. This game started 2-12 players with identical forces consisting of industry, bombers, ICBM's, ABM's, Subs, Sub missiles, Anti-sub planes, and a spy network. Each turn the players decide what to build with their industrial output (any of the forces mentioned except spys), and whether to move their spy network to a different country. Then they decide whether to launch an attack, and what forces to use. Sub missile attacks are not identified by country. Bomber attacks take two turns to complete, and can be called off on the second turn (keeping the bomber circling over enemy countries). ICBM attacks come through in one turn, so bomber attacks must be launched the turn before to co-ordinate. After the build/attack phase, an update would occur. Before seeing industrial output for that turn, the players would find out if they were under attack that turn, and attempt to fight it off if they were. This is done in an arcade style game with missiles and bombers passing overhead and ABM's being used to shoot them down. Bombers hit only once will drop their bombs and crash. Hit twice, they are destroyed before attacking. Missiles must only be hit once. ICBM and Bomber attacks identify the attacking country, sub missile attacks do not. If the player survives the attack phase with any industry the player's ASW planes can try to spot subs, and go into an arcade game for that phase, sub against plane. Then another build phase, started off with a spy report from the country your spy network is in. This game was lots of fun as a single computer game with many players taking turns, and would go faster with multiple computers. Similar game play with an original game is of course possible.