

Net Centric Computing – Graphics

Sandro Spina
Computer Graphics and Simulation Group

Computer Science Department
University of Malta

Graphics + Web = Web3D2009

- Web3D 2009 Symposium, 14th International Conference on 3D Web Technology.
- Sponsored by ACM SIGGRAPH and in cooperation with EuroGraphics
- www.web3d2009.org
- “14th in the series, the Web3D 2009 International Symposium will address this wide range of topics covering 3D hypermedia on the web. The annual Web3D Symposium is a major event, which unites researchers, developers, experimenters, and content creators in a dynamic learning environment. Attendees share and explore methods of using, enhancing, and creating new 3D web and multimedia technologies, such as X3D, VRML, COLLADA, Croquet, MPEG4, Java3D, and Canvas3D. The symposium will also address new trends such as interactive 3D graphics applications on mobile devices. “

Net Centric Computing

- Three main components
 - Traditional computing
 - A network
 - Computer Graphics + Visualisation
- You have done work on traditional computing (at the programming level)
- You have also discussed low-level networking in the initial part of this course.

Graphical Component

- In the next 4 hrs we shall introduce – at the application level – a number of CG concepts together with a number of tools.
- Concepts
 - SceneGraphs, Rendering, etc
- Tools
 - Scripting Languages
 - XML formats for 2D and 3D
 - Graphical APIs (Java3D, JavaFX, Croquet)

Graphical Applications

- Traditionally desktop based using graphics APIs (DirectX/OpenGL)...
- Which include networking code as part of the application...
- But each application is made to live on it's own ... there is no integration if not planned beforehand at the design stage ...
- It would be nice to have something similar to web pages where you know that once you've written your site/program it can be viewed by anyone over the internet.

Graphical Applications on the Web

- One can categorise software which somehow employs CG by the technologies they use:
 - HTML/Javascript (eg. ikariam.org is a widely popular MMOG)

- It would be nice to have something similar to web pages where you know that once you've written your site it can be viewed by anyone and it can somehow connect to other web sites.

Ikariam (Online Multiplayer)

The screenshot displays the Ikariam online multiplayer game interface. The browser address bar shows the URL: <http://s13.ikariam.org/index.php?view=city&id=64827>. The game interface includes a top navigation bar with links for Help, Ikariam PLUS (0), Highscore, Options, Board, Logout, v.0.3.0, and the date/time 18.02.2009 14:37:34. Below this, a status bar shows various resources and actions: 15 ships (15), 6 actions, 200 (391) people, 546,076 gold, 22,410 wood, 0 food, 3,282 stone, 29,512 iron, and 6,388 wheat. A dropdown menu shows the selected town: [80:85] Mladris. To the right, there are buttons for Show World, Show Island, and Show Town. Below these are four character portraits representing different game aspects: Towns, Military, Research, and Diplomacy. The main view is an isometric 3D rendering of a city with various buildings, a river, and a harbor. On the left, an 'Info' panel displays details for the town Mladris: Name: Mladris, Town size: 14, Player: Sandro, Ally: -. Below this is an 'inspect military' section with an image of troops and a button for 'Troops in town'. At the bottom, there is a copyright notice: © 2009 by Gameforge. All rights reserved., and links for Rules, T&Cs, and Imprint.

Ikariam (Source ...)

```

view-source:http://s13.ikariam.org/index.php?view=city&id=64827

////////////////////////////////// dynamic side-boxes. ////////////////////////////////////
////////////////////////////////// the main view. take care that it stretches. //////////////////////////////////// -->
<div id="information" class="dynamic">
  <h3 class="header">Info</h3>
  <div class="content">
    <ul class="cityinfo">
      <li class="name"><span class="textLabel">Name: </span>Mladris</li>
      <li class="citylevel"><span class="textLabel">Town size: </span>14</li>
      <li class="owner">
        <span class="textLabel">Player: </span>Sandro          </li>
      <li class="ally"><span class="textLabel">Ally: </span>-</li>
    </ul>
  </div><!-- end content -->
  <div class="footer"></div>
</div>
<div class="dynamic" id="reportInboxLeft">
  <h3 class="header">Inspect military</h3>
  <div class="content">
    
    <p>Inspect troops, that are stationed in the town</p>
    <div class="centerButton">
      <a href="?view=cityMilitary-army&id=64827" class="button">Troops in town</a>
    </div>
  </div>
  <div class="footer"></div>
</div>

<!--
////////////////////////////////// the main view. take care that it stretches. //////////////////////////////////// -->
<div id="mainview" class="phase14">
  <ul id="locations">

    <li id="position0" class="townHall">
      <div class="buildingimg"></div>
      <a href="?view=townHall&id=64827&position=0" title="Town hall Level 14"><span class="textLabel">Town hall Level 14</span></a>
    </li>
    <li id="position1" class="port">
      <div class="buildingimg"></div>
      <a href="?view=port&id=64827&position=1" title="Trading port Level 5"><span class="textLabel">Trading port Level 5</span></a>
    </li>
  </ul>

```


Alternatively ...

- One can split net centric application which are either
 - Deployed and rendered on a browser
 - Standalone executables which make use of custom networking (eg using RakNet)
- Clearly one can come up with other clustering criteria ...

Net Centric + Graphics

- As we've seen there are many forms by which graphical applications can exploit a network connection.
- And there are many technologies ...
- + custom built application such as distributed ray tracing or flight simulators.



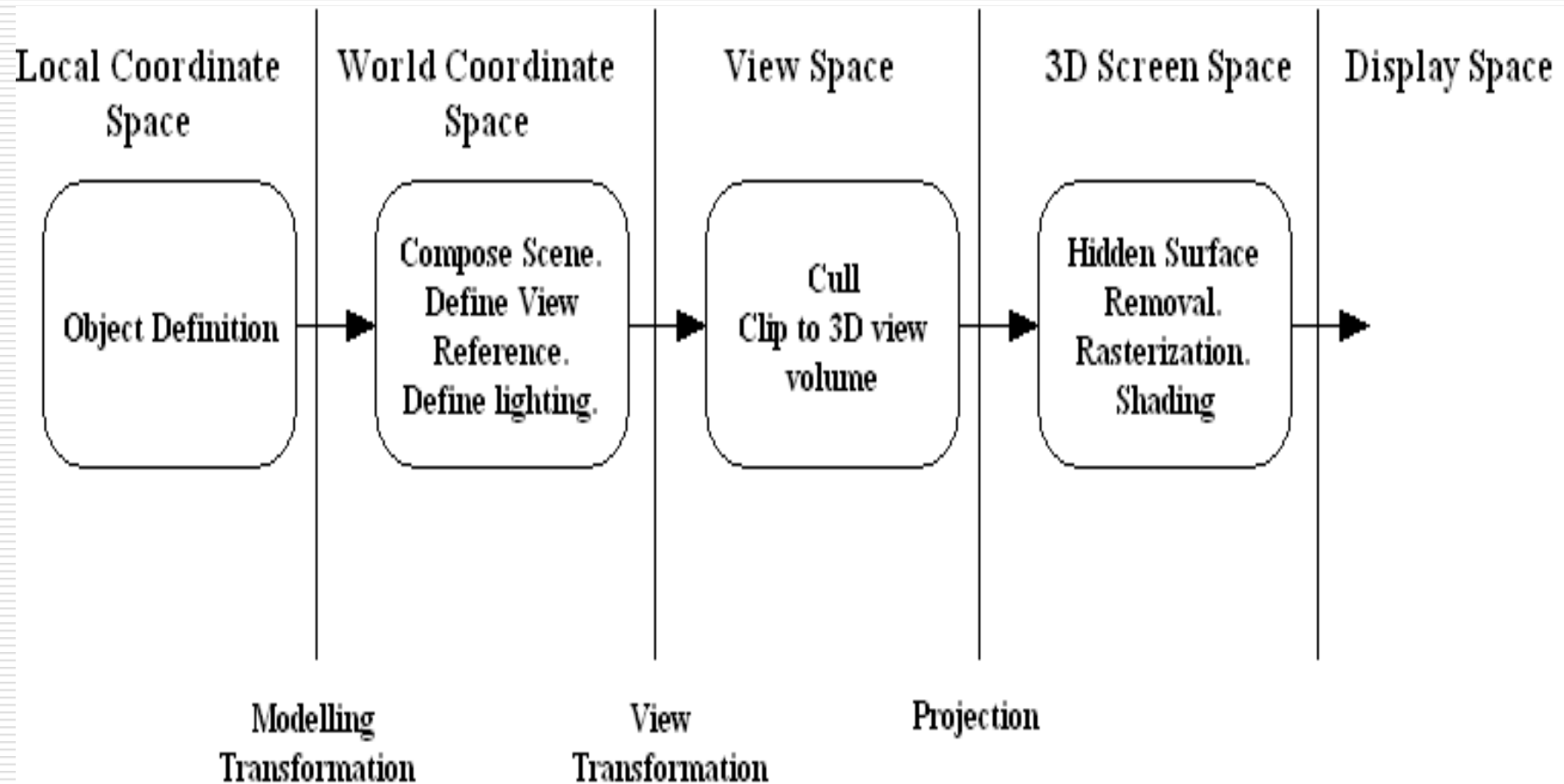
4hr Outline – Discuss Technologies

- Networking Libraries
- Graphics Pipelines
- SceneGraphs (concept)
- Middleware+MMOG (eg. RakNet)
- SVG: 2D graphics for the Web
- JavaFX
- X3D: VRML + XML (X3D Earth)
- Java3D (Applets or WebStart)
- Croquet (3D browsing !!)

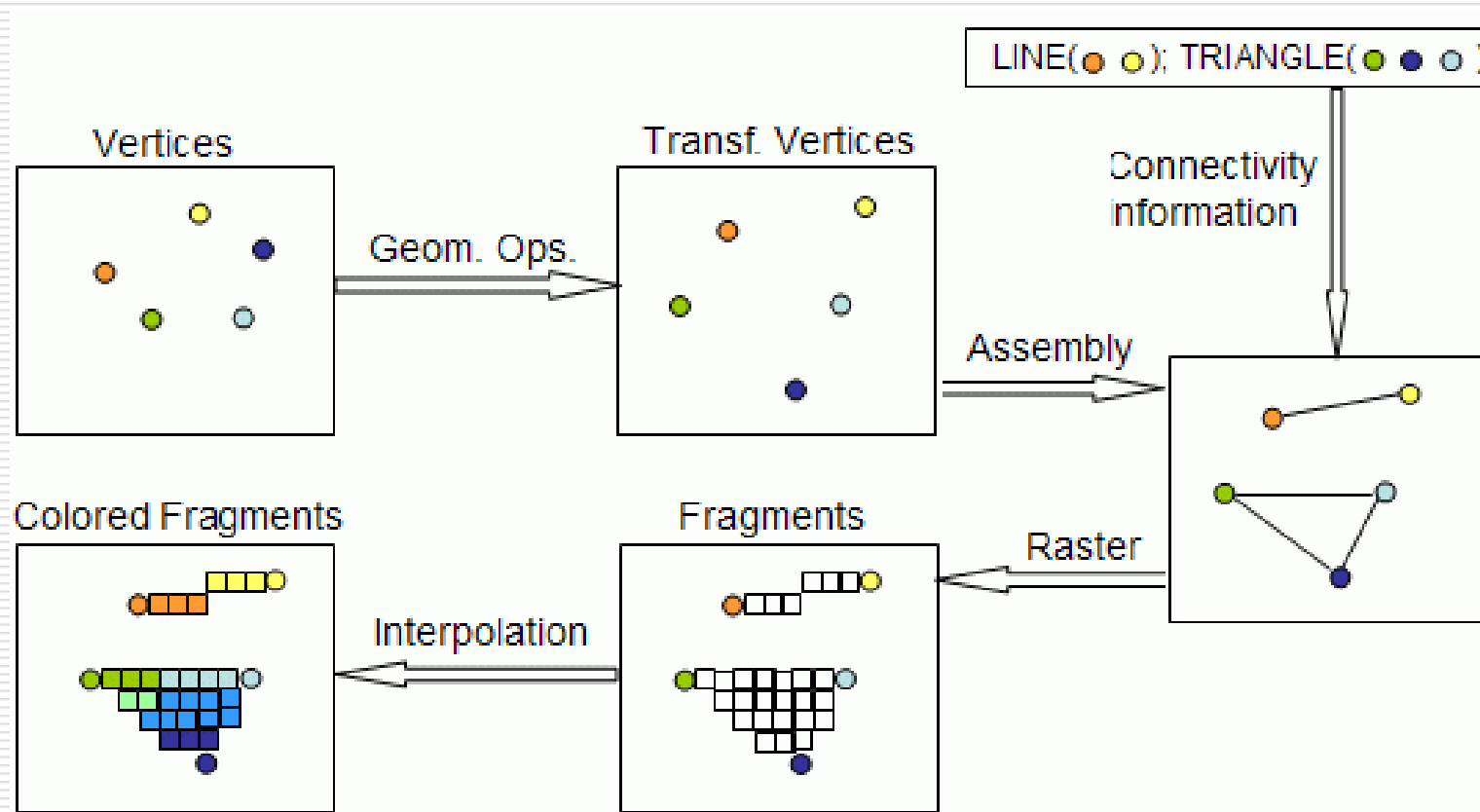
Graphics pipeline ...

- Rendering: takes some description of a 3D object, process it and then output a visually pleasing image on the screen.
- 3D geometry;
- Transformations (+Projective T);
- Lighting + Shading;
- Rasterisation;

Simplified view of Pipeline (i)



Simplified view of Pipeline (ii)



SceneGraphs

- Many technologies that we'll be looking at employ some form of scene graph representation.
- A scene graph is essentially a hierarchical representation (graph) of graphical objects in a scene.
- Takes care of the logical and spatial relationships between objects in a scene.
- First build the scene graph then render it ... as opposed to immediate mode apps which use opengl or directx directly.

OpenSceneGraph – an example

- Is an open source high performance 3D graphics toolkit, used by application developers in fields such as visual simulation, games, virtual reality, scientific visualisation and modelling.
- In OSG, the scene graph consists of a hierarchy of nodes representing
 - Spatial groups of related geometries
 - Animations,
 - Lighting + material
 - The leaves would represent the physical objects.
- For every frame the scene graph is traversed and objects in the view volume are rendered.
- Networking is achieved by integrating it with networking libraries such as RakNet and TNL ...

openTNL (Torque Network Library)

- “Is a robust, secure, easy-to-use, cross-platform c++ networking API designed for high-performance and games” ... www.opentnl.org
- Eg. Player class defines methods pack (converts useful player states such as velocity, ammo, location, etc ... into a stream of data) and unpack (which given this stream of data recovers velocity, ammo, location, etc...)
- The above is a simple example of how networking is carried out in a gaming scenario.

Network Protocols (UDP & TCP)

- UDP is unreliable while TCP is reliable
- In network programming (especially of real-time gaming) there are three fundamentally important limitations
 - High packet latency
 - Limited bandwidth
 - Packet loss

Coping with bandwidth limitations

- In multiuser simulations conserving bandwidth is extremely important
 - Clients may be connected over low bandwidth transport
 - Servers have to pay for bandwidth usage and cope with high number of clients

- Strategies to conserve bandwidth
 - Send static data once!!
 - Compress data to the minimum possible
 - Update only the parts that change (eg. Player updates do not need to include health level if that has not changed but just the new location and speed)

Coping with packet loss

- ...some packets will inevitably be lost!
- One can use TCP/IP but the overheads are huge for real-time networked simulations (although sometimes it is used)
- TNL implements a new network protocol similar to UDP but with more capabilities.

Latency ...

- Is the time it takes for information to travel on the physical medium from one host to the other
- On the client side one can make use of interpolation of data to smoothly move an object from where the client thinks it is to where the server declares it to be.

Typical Network Topologies (p2p)

- ... particularly adept to gaming (or any visualisation app)
- **Peer-to-Peer:** In a peer-to-peer network application, the client processes involved in the network communicate directly with one another. Though there may be one or more hosts with authoritative control over the network of peers, peer-to-peer applications largely distribute the responsibility for the simulation or application amongst the peers.

Typical Network Topologies (client-server)

- **Client-Server:** In the client-server model, one host on the network, the server, acts as a central communications hub for all the other hosts (the clients). Typically the server is authoritative, and is responsible for routing communication between the several clients.

Typical Network Topologies (Client-Tiered Server Cluster)

- **Client-Tiered Server Cluster:** In network applications where more clients want to subscribe to a service than can be accommodated by one server, the server's role will be handled by a cluster of servers peered together. The servers in the network communicate using the peer-to-peer model, and communicate with the clients using the client-server model. The servers may also communicate with an authoritative "super-server" to handle logins or resolve conflicts.

RakNet – Multiplayer on a Deadline

- Lobby System
 - Support for friends, rooms, quick match, ranking ...

- Object Replication System
 - Automatically create, destroy, serialize and transmit your game objects.

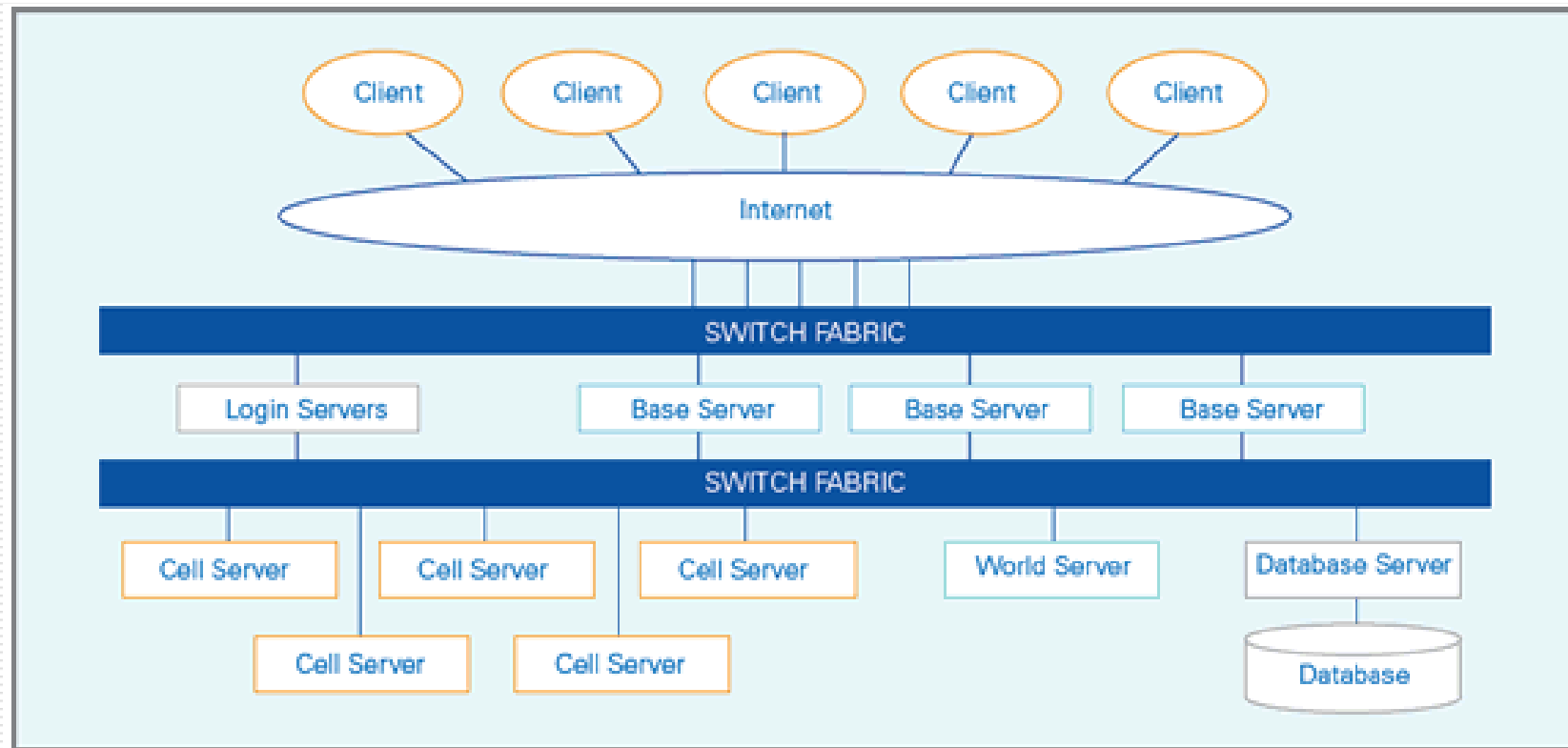
- Secure Connections
 - SHA1, AES128, SYN Cookies, and RSA to prevent and detect network attacks

- Robust Communication Layer
- Autopatcher – updates user with db driven binary patches
- RPC
- Voice Communication

Middleware Software – (takes care of networking)

- Recently there has been a move towards the development of middleware software for MMOG
- This middleware would be responsible to handle visualisation, networking, AI and all elements of a game.
- An example of such a development is BigWorld.

Middleware Software – (takes care of networking)



- Image source: bigworld site

Blink3D – create virtual worlds then publish

The logo for Blink 3D, featuring the word "Blink" in a stylized, lowercase font and "3D" in a larger, bold, uppercase font, both with a 3D effect and a color gradient from light to dark.

Quick Start Tutorial

Network Game Development in XNA

- Design targets:
 - Enable networked multiplayer games
 - Make the API easy to use
 - Support both Xbox LIVE and Games for Windows LIVE
 - Allow development with a single Xbox360 and PC
 - Don't require dedicated servers

XNA Network Configurations

- P2P networks: all the participants are clients of each other. I.e. when something changes on one computer, that computer sends a message to all other computers telling them what's happened.
- Client/Server: typically has one server, and the rest of the machines are clients. All communications are run through the server. If something happens on one computer, that computer sends a message to the server which in turn sends messages to the rest of the clients.
- P2P is inadequate with games like World of Warcraft ... imagine how many network connections you would need to keep open !!

Chasing game using XNA networking

- Let's say you want to implement a game where one player chases another player around the screen, with the goal of colliding with the other player ... the player being chased will earn more points the longer he stays away from the chaser. (note the real-time nature of the game - collision detection)
- We can use a P2P network for this game ...
- Note that passing data through a network is a bottleneck in terms of performance ...
- ... hence one has to figure out the least amount of information that is necessary to be sent over the network. In this case we only need the positions.

Game States ... in networking

- A typical game would have a start state where you display instructions for eg and splash screens, an in-game state and an end-game state ...
- In networking there could be more states for example
 - sign-in state (where the player signs into Xbox LIVE for eg.)
 - Find session state (where the player is searching for sessions of his game)
 - Create session state (where the player can create sessions)

Message types ... for our game

- It's always a good idea to create a list of possible message types
 - { StartGame, EndGame, RestartGame, RejoinLobby, UpdatePlayerPosition }
- Why do we pass these message types at the start of each message?

...Answer

- Because normally you wouldn't know whether you're expecting an int, or a string, or a vector, or two strings, etc...
- The message type (header) informs the receiving end what to expect.
- Hence you will be sending data informing the other end whether you want to start a game, end the game, rejoin the lobby, or update the players position.
- Data is sent in packets ... I'm sure you've discussed what a packet is in your previous classes.

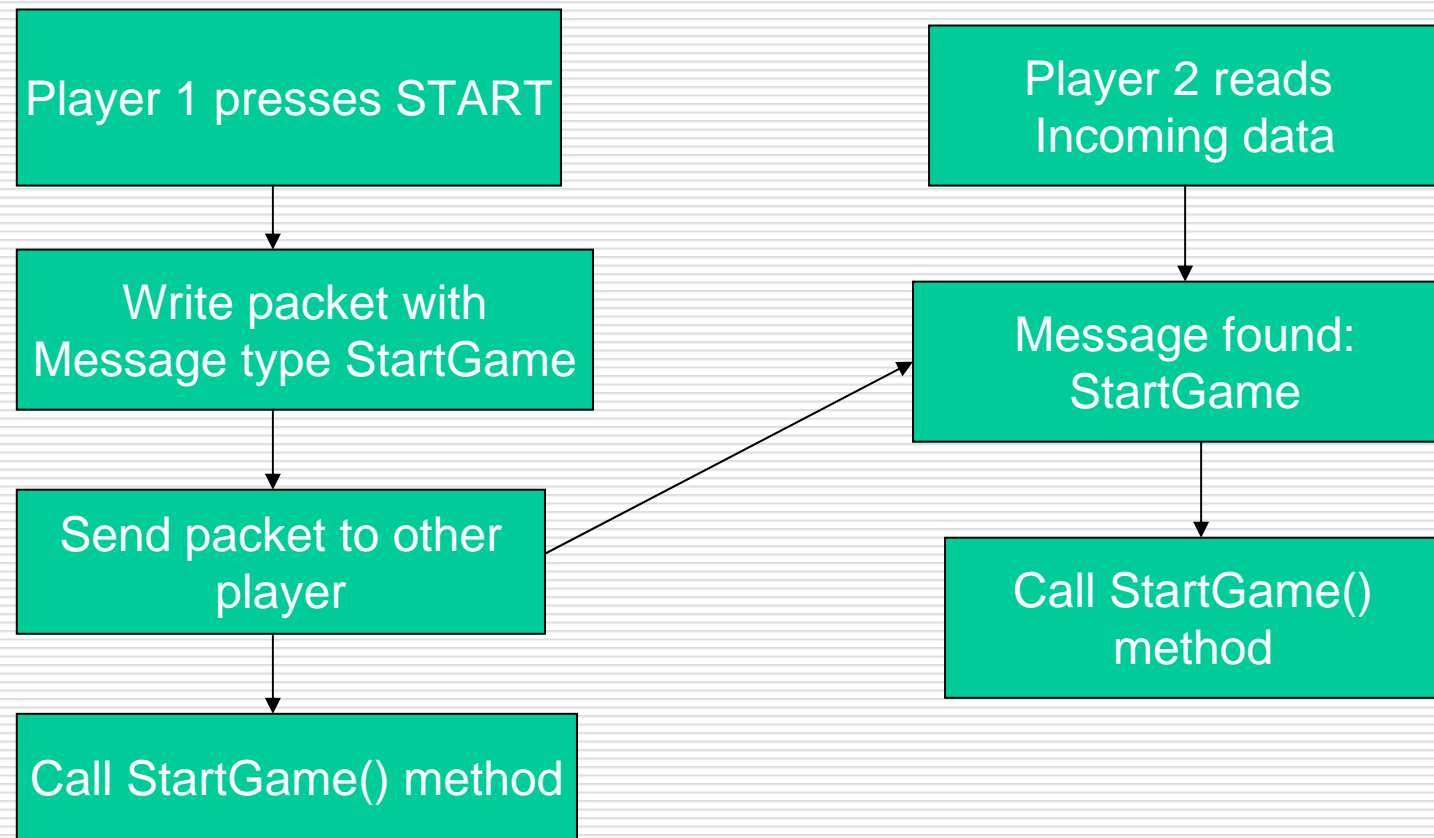
XNA Networking code – some classes

- NetworkSession class – this is the backbone of any networked game in XNA. This class represents a single multiplayer session of your game. From this class you can access all members of the session + other properties pertinent to the multiplayer session.
- PacketWriter class – send data across the network to the other computer
- PacketReader class – reads packets of information from the network

Network updating per state ...

- Depending on which state the game is it will call different methods to handle the different state ...
- This networking code is executed in the update part of the game loop
- It will first check if there is data to be read ... using:
 - `while (LocalNetworkGamer.IsDataAvailable) { }`
- If no data is available then it will skip reading any data from the network ... obviously a game can't block waiting for data.

Start Game flow diagram ...



More game logic ...

- As you can image there will be much more game logic involved the development of the game ...
- ... keeping up the scores
- ... drawing the remote players