**Comprehensive Meeting Topics**

The basic discussion topics for each meeting are recorded here. In some cases, details are provided below the table. Not all meetings had notes or minutes, especially for days where most of the team members were programming or doing design activities as a group.

|  |  |  |
| --- | --- | --- |
| **Date and Location of Meeting** | **Agenda** | **Plans** |
| April 1 (4:00 - 4:20) ***In-class*** | Trade Contact Information and Backgrounds | Meet on April 3rd for Planning & Design discussion |
| April 3 (3:30 - 4:45) ***Engineering* *Building*** | Begin Design Phase by assigning tasks to each member of the team. | Use Cases  Risk document  Communication model  Technology choices  Architecture overview  Process model  Specifications (requirements) doc  Mock-ups  Schedule |
| April 8 (3:30-4:45) ***Engineering* *Building*** | Design choices discussed  Iterations and schedule established  Discussed presentation  Review Use Cases, Architecture, and all docs | Slides assigned for Presentation  Technology research |
| April 10 (8:30-9:30) ***Google* *Chat*** | Review Presentation  Discuss Deliverable Package | Meet at Rockwell Collins on April 15th |
| April 15 (3:30-6:00)  ***Rockwell* *Collins*** | Architecture Design  Transfer of Ownership  Conflict Resolution  Network State  keepAlive  Canvas State and Canvas Events  Sockets | Outlined details for Iterations 2-4 |
| April 22 (3:30-6:00)  ***Cedar* *Rapids* *Restaurant*** | Network  Chat  GUI  Save/Load Session  Tool Loading | Continue working on iterations as planned in the schedule |
| April 26 (9:00-2:00)  ***Engineering Building*** | Networking  Loading  GUI  Code and documentation work during meeting | Continue working on iterations as planned in the schedule |
| April 29 (3:30-4:20)  ***Engineering Building*** | Network Drawing Canvas  GUI  Save/Load  Peer Discovery  Presentations | Finish final Documentation |
| May 1 (3:30-4:20)  ***Engineering Building*** | Final documentation discussion  Final code discussion | Wrap up documentation and code  Begin on Presentation |
| May 3 (9:00-1:00)  ***Engineering Building*** | Code and documentation work during meeting | Wrap up documentation and code  Begin on Presentation |

Detailed Minutes or Notes for certain sessions:

**Meeting 04/03/2009**

3:30-4:45 approximately.

**Attendees**: All

**Outcome:**

Following items assigned to be worked on.

Use cases: (after spec)

-          Joe  Transfer control

-          Joe  Join session (find, join, get state)

-          Joe  Check user alive

-          Mark Object event (add line, change color)

-          Mark Save session

-          Amanda Load session

-          Amanda Add tool/ passing tool

-          Amanda Leaving session

Things of interest

Architecture overview (spec document)  Ben after Spec/Use cases

Tentative technology choices (deferred)

Communication model (esp. mastership)  Ben

Development Plan/schedule  Mark

Risks  Amanda

Process   (iterative, some details, svn)  Amanda

Mock ups      (after Use case)     Satpreet

Spec document (requirements )  Joe

**Meeting 04/08/2009**

**Attendees**: All

Time: 3:30-4:45 approximately.

Review of work done since last meeting

**Todo**:

Slides as assigned

Architecture overview (spec document)  Ben after Spec/Use cases

Tenative technology choices (deferred)  Suggest RMI/SOAP -> Joe

Schedule -> Mark, some outline below

Issue resolution:

o Not doing passwords

Discussion on Apr10 Presentation

o Slides assigned to everyone

Discussion on Apr10 deliverables

General:

o Possible individual responsibilities:

Standardizing documentation

Iterations Plans:

Pass an object over the network / connect to other peer  (1/2 week)

Establish comm. Hierarchy objects  ( ½ week)

Div (1 week yikes!)

o   1.  Setup networking model

o   3.  Setup interface w/ object draw   revised GUI

o   2.  Layered event feedback (text output,  show events, (chat optional) etc..)

Div  (1 week)

o   Transferring control

o   Saving/loading session

o   Add tool / Dynamic tool loading

o   Network robustness

**Iterations:**

o Iteration 1:

Understand underlying technology

Connect to another peer;  (use old GUI)

Pass an object over network;  Acknowledge reception of object in some way;

o Iteration 2:

Establish a hierarchy of objects sent over network

Acknowledge reception and recognize object received

o Iteration 3:

Setup networking model (how we send/receive events, canvas, objects, etc.)

Make new GUI supporting collaboration

Make canvas receive a session “owner’s” canvas in real-time

Display network object feedback on chat window (layered event feedback)

(Optionally) Make P2P chat work

o Iteration 4:

Implement transfer of “ownership”

Loading and saving session (canvas + objects)

Dynamic tool loading

Tool sending and receiving over network

Network robustness/optimization

Scheduling

o We have 4 weeks + weekends left

o Iterations 1 & 2 – half week each

o Iterations 3 & 4 – 1 week each

o 1 week buffer

Decide on a code-set to use by Sunday evening

o Send access info. of your code-set to other members

*Discussion before presentation @ 9pm on Thursday on GTalk*

o Ben out to Minn.

Put your schedule/free-hours on the GDoc

**Actions items:**

Joe – Technology choices (pros and cons, SOAP)

Ben – Architecture

Satpreet – Technology choices (pros and cons, Java RMI)

Mark  - Schedule -> PPT, Technology choices (pros and cons, Java RMI)

Amanda – Documentation

ALL: Slides as assigned

Next Meeting: Monday @ 330 @ SC

Attendees: All-

Presentation Analysis / Feedback:

-Slight plan change, tool transfer must be automatic.

- Java sockets , RMI, or SOAP?   Architecture choice

                Sockets seem preferred.  Requested Mark to make example.

Going forward:

                Iteration one:  only first bullet is required.

Primary Mark to make example to satisfy iteration.

Ben to work on encapsulating functionality in session to work w/ network going forward

Joe to refactor/merge code to make shapes work nicer.

Satpreet – Add GUI for IP entry.

**Meeting 04/15/2009**

Attendees:All

Meeting Items:

First Iteration : Done

-          Automatically picking a port.  Done on side project.

Second Iteration: Done

-          Basic model laid out.

Third Iteration:

-          Making the second one more fleshed out in order to actually work.

-          Add button/textbox for join/create session Amanda

-          Add chatbox for output

Ongoing/side projects

                Fixes/merging code base

                Integrating sockets into object draw, required by the third iteration

**Architecture design discussion points:**

1.      Server is owner, ownership can rotate.

2.      Each peer can also serve.

3.      Each server has the free known socket known to the clients where the clients first make contact.

4.      The server sends back a connection message with the port number of another one where it could bind and make a socket.

5.      All peers in multi-draw will have to have a listening port where it’s an active owner.

6.      Current owner, well known port.

7.      These are other ports for which it’s maintaining links.

 -          There’s another collaborator that gets in touch with another guy who sends back an acceptance message that says you can connect to the other port on the list.

**Ports:**

Process all the events via one port?

Internal multiplexing

Not 5 pipes, but one that is multiplexing.

Each has 2 ports, one is sending and one is receiving port.

-          Each user has access to one port to process data for the I/o

-          You can send and receive on the same port.

**Model A**

One collaborator and a port.

Once the owner knows the ports it can broadcast to the clients.

Bidirectional links.

Second iteration will add robustness

**Object hierarchy is next**

ObjectDraw is multithreaded

Q. Are the listeners being processed in a separate thread or is that internal and doesn’t concern us?

- Right now it’s sngle threaded. Listening to all events but only one can occur at a time.

- Internal to the application a single thread is running at one time.

Need 2 threads:

-          One for objectdraw

-          One for network events - keepAlive and managing sockets

KeepAlive is asynchronous. Doesn’t need to be time shifted. If peer one connects at 1:00 and peer 2 connects at time 1:05. Should keepAlive be sent at 1+delta?

-          Send to everyone at a regular interval.

One is acting as a server and one is a client, how does the client become a server without a disconnect?

-          Create an event to pass control.

-          Once his session event says he’s the owner he can then access the canvas as an owner.

-          If it says you can draw the canvas then you can make events happen on that canvas as though you’re an owner.

**Conflict Resolution:**

All frozen state during a resolution phase?

**Things to consider with resolution:**

-          Originator

-          Receiver

-          Data itself

-          Reason the event was created

-          Types of packets

-          how we organize the data (reflection?)

**Object called “data” right  now that needs to be fleshed out**.

1. **Chat String**  Iteration 2

2. **Transfer of ownership**  Event               Iteration 3?

**Join Request:**

                        Is there a leave request ?

                        Notifications of leaving the session. Session end if it’s an owner.

                        Join request session.

            Right now: use a message number, source destination, message type and data.

            Serialized version: [msg no][src][dst][msg type][data….]

3. **KeepAlive:** list of peers + IP

            Tool, transfer?                          4th iteration?

            Request for new tool?

            Get/Send every 1/10 sec

            Only update the list if it changes

4. **Network state**

            Who all is on the network

In case network failure or other interesting events we can find out who else is available

            Can keepAlive piggyback this info?

            Start session messages?

Time allowed, message number as part of the object. Say you get message 10, where is 1-9? Say message 9 was “create line” and message 10 was “resize line  then it should be processed in order.

5. **Canvas state:** Full object list (end of iteration 3)

***Discussion:***

Start off, first thing you receive is entire object list.

Then you get whatever is changed.

What does Java guarantee with the socket?

Exception: socket lost, session lost, etc?

Examine this in a later iteration? Let’s keep this in mind and look at it for debugging if nothing else.

From the start or the life of the owner, do we maintain the state/message number? keep it in mind for debug too.

Do we track the events between the start and end result?

If a guy is drawing an object do we send the intermediate positions before the end points are selected? **No**

**Iteration 2 Details:**

- Want to implement the basic idea, make the classes so everyone inherits them in the next iteration.

            - Fixing up the codebase.

            - Simplify it

- Add start button to add IP and port and put it to the bottom and set it up to feed in to start a session and feed to a new computer

            - Get buttons on the screen so they’re there to click on

            - Buttons

            - Text window to print chat

**Iteration 3**

            Need to be able to at least view the owner’s canvas.

            Full iteration with canvas events.

            Making it actually work, fill in the blanks made by the 2nd iteration.

                        Canvas objects.

            Convert the MockUp into Java Code  end of the 3rd iteration

            Even if they are just inactive items

**Events and Messages**

                        Once you join you get the current state and every event is sent

                        Define different kinds of events

                                    Should we use reflection?

Allows you to check what methods and properties of an object.

                                                Still need to do some structural stuff. MousePressed and MouseReleased are events or do we send every event that happens on the canvas to them?

                                                This is a Proxy Controller (MVC):

Everything goes through the session and creates event on the local canvas and goes over network and creates the same response or they don’t send over the network.

**Iteration 4**

            Saving sessions - dependant on the networking issue.

            Session object is already there.

Might not care about who belongs to the session.

Call it “Save Canvas State” so that we can save current objects on the canvas so we can pull them back in.

***Situational example:***

Load a New Canvas state

Peers Join

Receive current canvas state

Owner so the mouse clicks the tool

                        Not a new canvas event, it’s not something the views need to see.

Owner clicks on the canvas

Gets mouse clicked event, determines tools, responds appropriately on the master side, and then the

**Canvas Event Object**

            Current tool

                                    Mouse press

                                    Mouse drag

                                    Mouse release

**ControlBar Event Object**

*Color change as an event?*

                                    Master changes color, does it change on the receiver side?

Not until the shape is finalized that we care what the color is because it draws the outline boundary in grey anyway.

If you change the color of an object then it’s a canvas event. If you just click the color then it’s a ControlBar Event.

*What happens when you select?  What happens when you move?*

                                    Object is grey

                                    Retains its color though

**Ex:** Object is red but it’s drawn as grey to show that it’s selected

Should we combine the stuff from the 2nd color that shows the grey box and the red color when it’s drawn?

**Color**

**Fill**

Is ControlBar meaningless if you do an event from the MenuBar?

**Use Case: Select**

*Click on the rectangle object. What does it do?*

                        Sends  Select Event at this point with MousePressed.

                        Release mouse sends the same thing and it’s done.

                        DragMouse sends the displacement values for the origin point.

            Ex: *If I have nothing selected and I change the color to red and draw a rectangle, it should come out as red on my screen, but what about on their screen?*

It gets attached to the MouseRelease when it finalizes the color & sends it to users

**Sockets**

Code base fixing

Peer to Peer

GUI

Save/Load Canvas State

**Discussion Notes (4-22-09):**

*All attendees present*

Things to do:

-          Ben just work on remaining events.

-          Satpreet – integrate Chat program, p2p model

-          Amanda Gui Work

-          Mark continue on saving/loading.

-          Joe look at tool loading.

**Discussion Notes (4-26-09):**

Coding meeting.

Ben

-          Keyboard tool events don’t transmit,

-          Object equivalence over network.

Amanda

-          Integrate GUI

Joe

                Finish start up

Satpreet

                Networking

Mark

                Finish loading/integrate

**Discussion Notes (4-26-09):**

Group meeting, attendees all:

-          Need to schedule personal presentation.

-          Some GUI refactoring Joe/Amanda

-          Saving / loading -> Joe

-          Network peer discovery -> Ben

-          DrawingCanvas.doLoad -> needs to be on network -> Ben

Documentation -> Satpreet, Mark

FINAL PRESENTATION

10 minutes

Overview of Architecture and design

Schedule initial versus actual

Brief demo.

**Discussion Notes May 1, 2009**

**Group meeting, attendees Ben, Joe, Satpreet & Mark:**

**Meeting items today:**  
- Network robustness and other corner-cases pushed to later  
- We meet all day Sunday to get as much of the job done, start at 10 am @ the Engg. (Elder) lab.   
  
**Actions:**  
- Ben: taking care of Peer-discovery and Chat (waiting for GUI chat enablers to be done by Joe/Amanda)  
- Joe & Sat: figure out reflection  
- Amanda , Mark & Sat: take the lead on documentation. Demand whatever info. you need from the others to get the job done... It would be nice to have a step-by-step walkthrough of the features we will demo on Wed.  
- Mark: check out the projector setting for the classroom in which we are presenting

**On documentation:** One suggestion is to make a list of diagrams that you'd like to put.

When we meet on Sunday**,** we can run through the code so to be able to make the necessary block-diagrams + UML/class-diags + collab-diags etc. and the accompanying documentation.

**Discussion Notes May 3, 2009  
Group meeting, attendees ALL:**

**Meeting items today:**  
- Ben/Joe: worked on transfer-of-ownership coding  
- Amanda/Mark/Sat: Documentation work  
- Joe/Ben/Sat: Tool-loading / reflection  
  
**Actions taken back:**

**Notes:**

Ben & Joe & Satpreet: Working on code, networking and testing final work.

Mark & Amanda: Working on documentation final work