QUINTET.NET: AN INTERACTIVE PERFORMANCE ENVIRONMENT FOR THE INTERNET

Georg Hajdu

Hamburg Academy of Music and Theater

For me, the highlight was Georg Hajdu's "Quintet. net: A Quintet on the Internet" because it is an immediately pragmatic system and overcomes in a musical way some of the limitations of Internet-based performance systems.

Ian Whalley
Computer Music Journal 28.1 Summer 2004

1. Intro

Quintet.net is an interactive network performance environment invented and developed by composer and computer musician Georg Hajdu. It enables performers at up to five locations to play music over the Internet under the control of a "conductor." The environment, which was programmed with the graphical programming language Max/MSP consists of four components:

- Server,
- Client,
- Conductor,
- Listener as well as the
- Viewer add-on.

These components exchange data using Matt Wright's *OpenSoundControl (OSC)* and *otudp* objects, which were the first objects to implement the User Datagram Protocol in the Max programming environment. Quintet.net was conceived in 1999, soon after CNMAT made these objects publicly available.

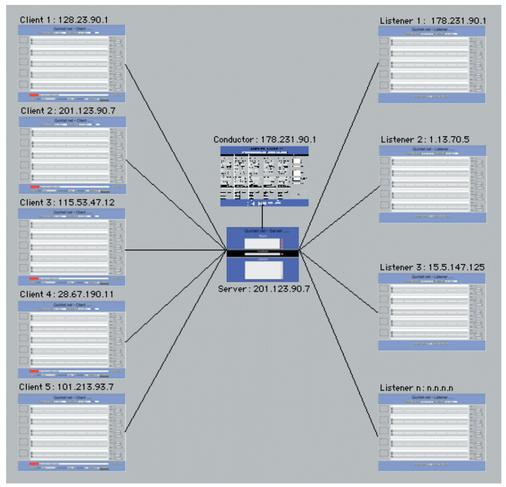
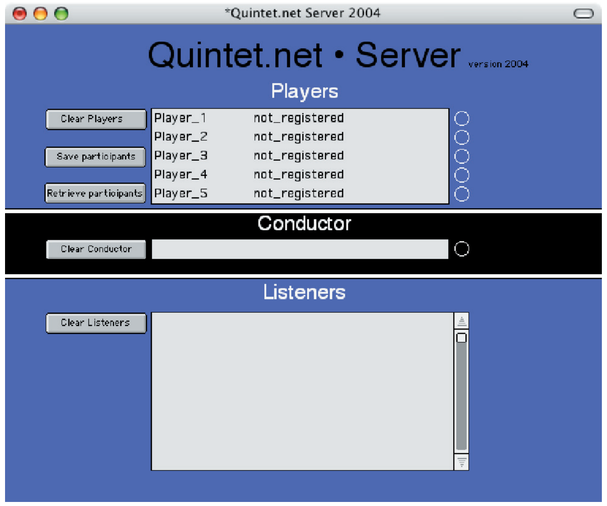
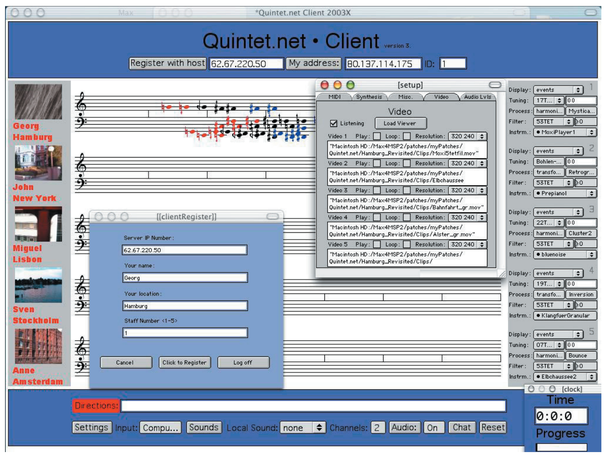


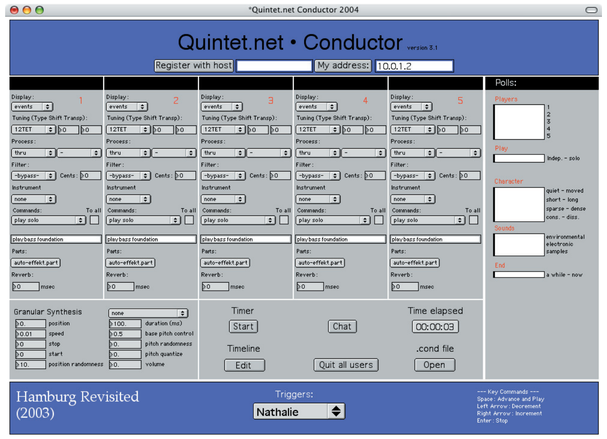
Figure 1: Quintet.net consists of 4 program components connected in a local network or over the Internet.

2. Interaction under the control of a conductor

The players interact over the Internet by sending control streams to the Server either using a pitch-tracker, MIDI or simply the computer keyboard. On the Server, the streams get copied, processed, and sent back to the Clients as well as to the Listeners. In addition, a conductor can log onto the server and control the musical outcome by changing settings remotely and sending streams of parameter values as well as short commands to the players.







Figures 2-4: Screen shots of the Server, Client and Conductor components

3. Audio, music notation and live video processing

Quintet.net uses a sampler or MIDI for instrumental playback. It also features granular synthesis as well as vst~ plug-ins for sound processing and playback, and has additional video and graphical properties, which permit better interaction and control on a symbolical level: The performers along with the audience see the music which the participants produce on screen in "space" notation on five grand staves. In addition video clips and/or live video can be displayed by the Viewer add-on and mixed with real-time music notation for an enhanced viewing experience. The Conductor can also read musical scores and send parts to the performers, which will be displayed along with the notes produced by the musicians.

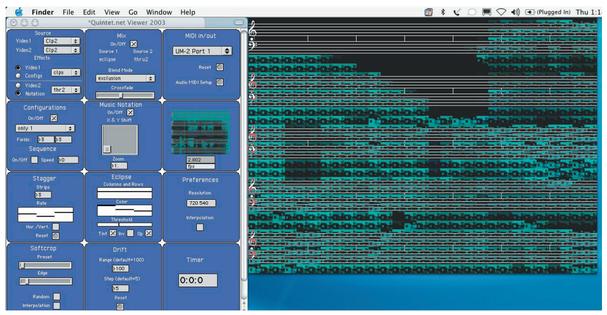


Figure 5: Screenshot of the Viewer add-on which can be used with the Client and/or the Listener.

Figure 6: Guitarist Erhard Hirt during a performance of *Five* by John Cage

4. Composing for Quintet.net

The music performed with Quintet.net is a combination of composed and improvised elements. The lack of real synchronicity due to the usual delays on the Net, necessitates the adaptation of a genuine "Internet" performance style for which John Cage's number pieces could be considered a model: These pieces require certain notes or phrases to be played within "time brackets."

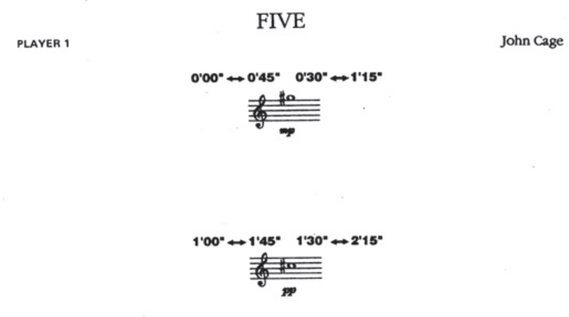
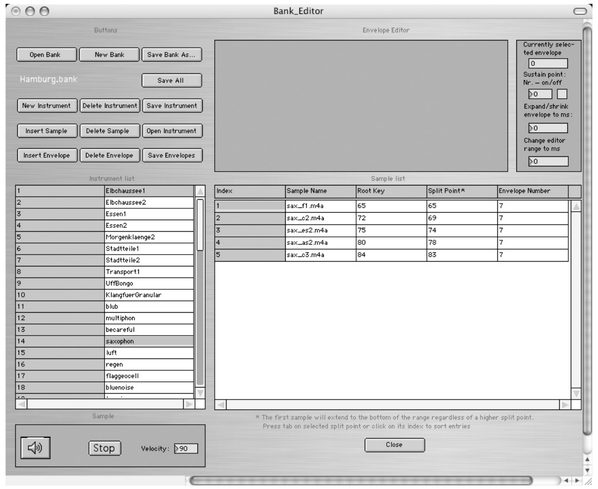
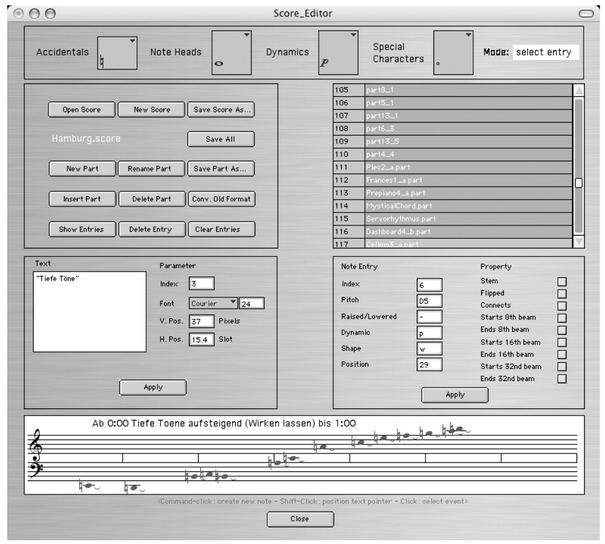


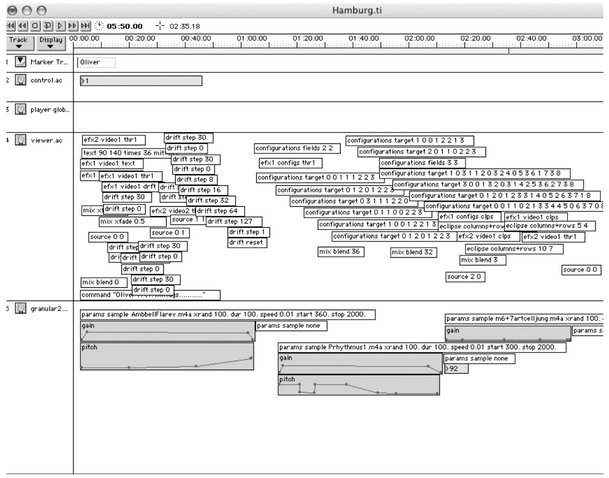
Figure 7: Excerpt from the score (part 1) of *Five* by John Cage. The musical events are allowed to occur freely between time brackets

5. Composition development kit

Recently, a composition development kit, including several visual editors, was added in order to facilitate the creation of pieces for the environment.







Figures 8-10: Screen shots of the Bank editor, Score Editor, and a timeline which the conductor loads to control the performance

6. Pieces realized with Quintet.net

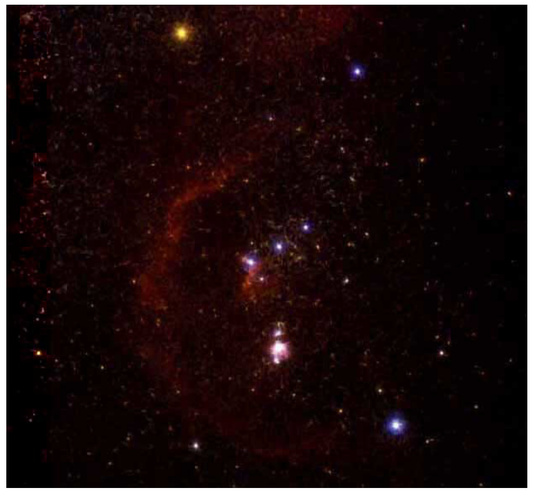


Figure 11: Georg Hajdu's MindTrip is a piece about the possibility of communicating with an extra-terrestrial intelligence.

(The city names indicate the location of the conductor)

Georg Hajdu: *MindTrip* (2000)
Festival Mystik und Maschine, Münster, October 2000

John Cage: *Five* (1986, arr. 2001)
Anne around, Groningen, November 2001

Manfred Stahnke: *Orpheus Kristall* (2002)
Munich Biennale for Contemporary Opera, Munich, May 2002

Anne La Berge: *Vamp.net* (2002)
Eindhoven, October 2002

Hamburg Network Composers' Collective: *Hamburg Revisited* (2003)
Campus Innovation, Hamburg, September 2003

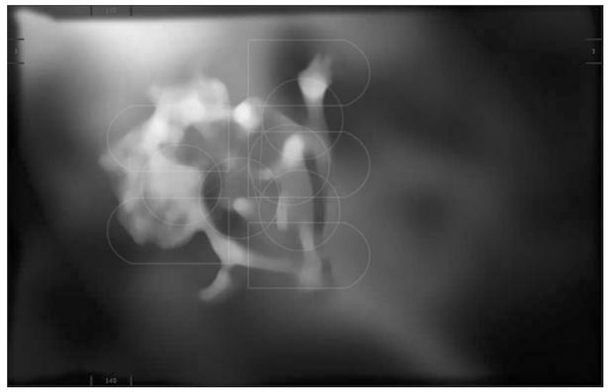


Figure 12: Screenshot of Bettina Westerheide's web project, which retold the Orpheus myth and its adaptation in Orpheus Kristall by means of a Flash-animated website (www.orpheuskristall.net).

7. References

on the Web:
http://www.quintet.net
http://www.mohr-i.nl/content.phtml?328
http://www.tammen.org/ens_hajdu.html
http://www.harvestworks.org/maxreel/

in print:
Hajdu, Georg (2003). Quintet.net – A Quintet on the Internet. Proceedings of the International Computer Music Conference, University of Singapore, Singapore. October 2003. 315-318.

Georg Hajdu, Quintet.net - Präliminarien zu einer vernetzten, interaktiven Echtzeitkompositionsumgebung, in: Global Village – Global Brain – Global Music, hrsg. v. Bernd Enders und Joachim Stange-Elbe, Osnabrück 2003, S. 298-303

Time (min.)	Bart	Carl	Georg	Marko	Robert
00.00	text samples improv	play a few notes and stop	play a few notes and stop	duo with Robert	duo with Marko
01.00	text improv until 2'	play a few high notes and stop	play a few high notes and stop	play low notes	play low notes
02.00	stop playing	play one long note once	play one long note once	sparser duo with Robert	sparser duo with Marko
03.00	don't play -Georg solo until 4'	don't play -Georg solo until 4'	Georg - your solo	don't play -Georg solo until 4'	don't play -Georg solo until 4'
04.00	duo with Georg	text samples improv	duo with Bart	improv off line	play 7 chords and stop
05.00					
06.00	more frantic/soft sounds	sparser text improv	play denser and softly	play two sounds and stop	play one long sound once
07.00					
08.00	stop playing	play a duo with Robert	text samples improv	stop playing	play a duo with Carl
09.00					
10.00	play 1 text sample 20 times	play only chords	play occasional sound & text	play 1 text sample 15 times	play high sounds very sparsely
11.00					
12.00	play a duo with Carl	play a duo with Bart	stop playing	text samples improv	stop playing
13.00					
14.00					
15.00	don't play-Robert solo until 16	don't play-Robert solo until 16	don't play-Robert solo until 16	don't play-Robert solo until 16	Robert solo until 16'
16.00	play 5 notes slowly and stop	play 3 notes slowly and stop	play a duo with Marko	play a duo with Georg	text samples improv
17.00					
18.00			play 5 text samples and stop	play 1 text sample once	stop playing
18.20	play 3 text samples and stop	play 2 text samples and stop			
19.00	play 3 quick fragments	play 7 slow fragments	play 5 melodic fragments	play 1 text sample 4 times	play 2 slow melodies
near end	play one beautiful note	play one beautiful note	play one beautiful note	play one beautiful note	play one beautiful note
20.00	end of piece	end of piece	end of piece	end of piece	end of piece

Figure 13 "Score" to Anne La Berge's piece Vamp.net

Golo Föllmer's dissertation: Musikmachen im Netz. Elektronische, ästhetische und soziale Strukturen einer partizipativen Musik, Dissertation (mit CD-ROM), Martin-Luther-Universität Halle-Wittenberg 2002.

Joerg Stelkens' dissertation: Netzwerk-Synthesizer. Konzeptionierung eines internetbasierten Multi-User-Softwareinstrumentes zur kollaborativen Klang- und Musikerzeugung unter Betrachtung historischer und aktueller Ansätze im Bereich der elektronischen und Computer-Musik und unter besonderer Berücksichtigung der Latenzproblematiken digitaler Netzwerke.

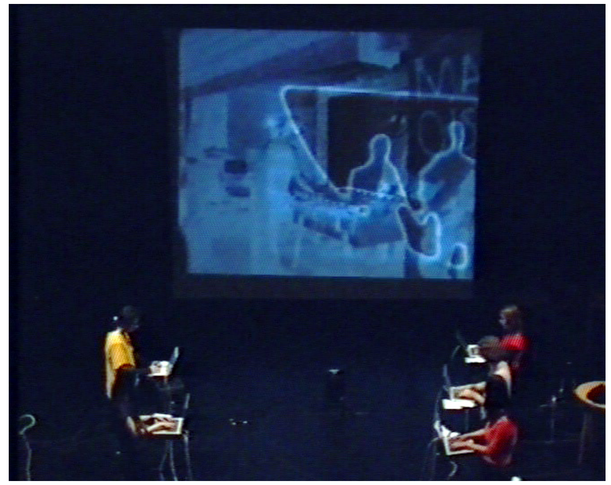


Figure 14: Performance of Hamburg Revisited

8. Appendix: OSC messages

Server participant = client, listener, conductor	**Client**: n = Client ID	**Listener**	**Conductor**
Port 7970	Port 7971	Port 7972	Port 7973
Send: /n/score <sequence> /n/freeze <list> /n/command <list> /n/setting <viewer/gs params> /n/chat <list> /n/player <symbol> /n/location <symbol> /n/id <int> /n/process <list> /n/filter <symbol int> /n/sound <symbol> /n/spectrum <list> /n/event <list> /n/clock <list> /n/video <list> /n/reverb <list> /n/tuning <list>	Send: /client/location <list> /client/name <list> /client/hello <symbol> /client/logoff <symbol> /client/chat <list> /process <n list> /filter <n listt> /sound <n symbol> /reverb <n int> /tuning <n list> /n/spectrum <list> /n/event/ <note/bend list>	Send: /listener/location <list> /listener/name <list> /listener/hello <symbol> /listener/logoff <symbol> /listener/chat <list> /eval <list>	Send: /conductor/location <list> /conductor/name <list> /conductor/hello <symbol> /conductor/logoff<symbol> /conductor/chat <list> /conductor/clock/time /conductor/panel <0 output/process/filter/reverb/¬ video list> /conductor/setting <list> /conductor/score <sequence> /conductor/freeze <list>
Receive: /participant/location <list> /participant/name <list> /participant/hello <list> /participant/logoff <list> /participant/chat <list> /n/spectrum <list> /n/event/ note <list> /n/event/ bend <list> /process <n list> /filter <n listt> /sound <n symbol> /reverb <n int> /tuning <n list> /chat <list> /eval <list> /video <list>	Receive: /n/score <sequence> /n/freeze <list> /n/command <list> /n/setting <viewer/gs params> /n/chat <list> /n/player <symbol> /n/location <symbol> /n/id <int> /n/process <list> /n/filter <list> /n/sound <symbol> /n/spectrum <list> /n/event <list> /n/clock <list> /n/video <list> /n/reverb <list> /n/tuning <list>	Receive: /chat <list> /title (not yet implemented) /n/setting <viewer/gs params> /n/score <sequence> /n/freeze <list> /n/player <symbol> /n/location <symbol> /n/id <int> /n/process <list> /n/filter <list> /n/sound <symbol> /n/spectrum <list> /n/event <list> /n/clock <list> /n/video <list> /n/reverb <list> /n/tuning <list>	Receive: /chat <list> /eval <list> /player <n list> /command <n list> /n/setting <viewer/gs params> /n/freeze <list> /n/player <symbol> /n/process <list> /n/filter <list> /n/sound <symbol> /n/clock <list> /n/reverb <list> /n/tuning <list>