The Music Impro App

Final Report

INF5261 - Development of mobile information systems and services
Spring 2011 - The University of Oslo : Department of Informatics
Our project is a part of the RHYME project

Student group:
Jan Ole Skotterud, Tommy Madsen, Geirr Sethre, Kjartan Vestvik & Joakim Bording
(Master students in Informatics: Design, Use and Interaction)

Content

Introduction

“Without music, life would be a mistake” (Friedrich Nietzsche)

The Rhyme project state one of their goals as “Develop knowledge through design of new concepts of Internet of Things. Things that facilitate play, communication and co-creation, between people with and without severe disabilities, on equal terms, in different communicative situations.” (About RHYME, 2011). The intention of the *Music Impro App* project was to develop a prototype application that may support this goal through the use of common smart phones running traditional software platforms. These mobile devices have a range of different sensor input, like accelerometers, orientation sensors and magnetic fields sensors, that is not yet widely used in application development. By facilitating these types of sensor technology, we aim to turn the users mobile device into a music instrument that will be easy to play with and intuitive to use and inspire co-creation of music through the use of movement and gestures.

The project had to overcome various technical challenges, figuring out how this could be made and identify technical limitations that affected the concept. We also faced a lot of challenges on how to support both musical expression and co-creation by using alternative forms of interaction, that users may not be very familiar with from traditional use of smart phones. On top of that we’ve had theoretical challenges as to who we designed for, and how to explain the ways users reacted on our prototype. We hope this project may clear some ground for others walking down the same road as we did.

Problem Space

Traditional music creation needs practice and skill. To play an instrument like for instance a flute, you first need to know and understand some theory behind how the instrument is supposed to be used. Place your fingers over the holes and release them in order from bottom and up to produce a normal musical scale. Second you need practice to master breathing and blowing techniques and good finger control to produce clean and pleasant sound from the instrument. And third, practice and skill needs to be combined in how to create combinations of these sounds to melodies that sound like music as we know it today. Everybody who have had flute classes at their primary school know that this does not come easy and the first musical results does not sound nice.

The ability to master an instrument and together with others play music is said to be a joyful experience. Apart from the entertainment it has also been shown that music may have positive effects on health (Ruud, 2009). Music can be used to reduce the amount of pain experienced, but may also provide ways of communication for children or elders with disabilities (Robertson, 2010). Robertson (2010) describes a case where a blind child with autism finds a way through music to help him communicate. When asked of what he thinks of his mother he sits down by the piano and play *Twinkle, twinkle, little star*. Since most traditional instrument requires much practice and skill, this may become too much of a challenge for some disabled children, and the feeling of mastery and the social collaboration of music is lost. Inspired by the Rhyme project we want to use universal design in a way that makes co-creation of music possible, without the need for practice or specific skills.

When dedicated treatment tools based on information technology is designed and produced, it may be dependent on dedicated hardware made especially for the system. These may be controllers, sensors, or the main device itself. The use situation may require special needs from the hardware that makes it necessary to make dedicated solutions that supports those needs. The Orfi prototypes from the RHYME project had sensors, speakers and transmitters built into pillows. By tangible interaction with these pillows users controlled sounds and musical expressions set in a musical theme controlled by one theme pillow. The designers did not want to put limitations on the use of these pillows, and therefore wanted to make the hardware sturdy enough to withstand being thrown into walls or to support the weight of adult users using the pillows to sit (Birgitta Cappelen, private conversation, March 2011).

But dedicated hardware comes with a cost. To distribute the final product you are dependent on a manufacturing line of the dedicated hardware, distributors who are willing to buy a stock of your product and places that sell and share the knowledge of your product to the end customer. In the last years, with the introduction of smart phones on the market, the use of “apps” (small applications that run on a mobile device) has exploded. One of the factors of success for these applications is the centralized marketplace for each OS that gives a short and easy distribution line from developers to end users. Final products are suddenly available to all users of that platform without any extra cost for the developer. We wanted to experiment with an open platform to see how the hardware in today's smart phones can be utilized for this kind of systems. Even though the hardware may not be ideal in all cases, we believe the availability of the possible final product will outweigh the limitations.

Music applications made for smart phones are usually dependent on the touch screen interface as the main source of interaction. The small screens require good motoric precision to operate and normal sight to distinguish the options and the appropriate action to take. Aligning this to the Universal Design paradigm, means that a final product should consider communicating sound combined with touch, to facilitate our design for users with impaired vision.

Based on our review of current music applications for mobile devices we will state that they seldom supports co-creation. Music may be shared and possibly adapted and recreated, but systems that supports live co-creation of music have we yet not found. By playing the controlled music out loud from each device, co-creation is often possible, but it is much more dependent on the skill of the users and there may be some situations where it is almost impossible to let the loops sync and get everybody to play in the same key. A system designed for co-creation is needed.

Goals

The RHYME project has stated two main goals at their homepage:

> ***1.*** *Radically improve the health and well-being of people with severe disabilities, through a new treatment paradigm based upon collaborative, tangible, net based, "smart objects", with multimedia capabilities*
> ***2.*** *Contribute to the emerging body of knowledge in interface design, specifically within the area of networked social co-creation, based on multimodal interaction through the Internet of Things.*

(About RHYME, 2011):

These goals coincide with the underlying goals of our related project, and we hope to contribute to some of them. Based on the universal design principles we want to let our system be usable by the general public, but also include users with disabilities and special limitations. To narrow the scope we have decided to use open application frameworks for smart phones, currently the Google Android platform. We also focus on live co-creation of music through gesture interaction, where we use the movement of the device itself as the main form of interaction. Our main research question in this project can be summed up to:

- How can a mobile application running on the Android OS be designed to enable and inspire co-creation of music through movement and gestures?

Users

As well as defining specific users is valuable in regards to defining the scope of what is to be designed, this can potentially be forcing unwanted limitations on a project before you know your target group well enough. Since music is such a universal language of communication, we seek to avoid limitations not just to reach a broad audience, but also to make a user experience possible for users commercial products normally don't take into consideration. Our approach to the users will thus be based on the principles of the discipline known as universal design. "Universal design means simply designing all products, buildings and exterior spaces to be usable by all people to the greatest extent possible" (Mace et.al 1991:2).

Based on this, we need to make sure our application will be usable for people with disabilities as well as people without disabilities. Potential users is basically only limited to people that can benefit from communicating, experimenting and playing with music, and wish to do so. One good example here, is that an application should be fun to use for a father as well as his kid. But when it comes to user testing on potential users, we aim to meet the least common multiple in order to follow the principles of universal design. This means in practice that testing should be done on users with special needs, i.e. kids and people with disabilities. User testing on people with disabilities was unfortunately not an option in this project and as a result of this, the first workshop conducted in this project included normal children and their parents, as well as fellow students.

Psychographics and demographics

Without a critical mass of users or sufficient data to analyze from similar research, it seemed difficult to conduct a demographic profile. As we pointed out earlier, there is not much to compare with related to other co-creative music apps on the market. Psychographic profiling on the other hand, is a helpful tool in interaction design to describe attitudes and perceptions that users have about the world, and/or about the subject matter at hand (Garrett 2011:44). Psychographic profiling can in this way be a useful tool to ensure if we are working in the right direction according to our users cognitive understanding of our design.

The scope of users:

- Like/enjoy or have an interest in music
- Have a positive attitude towards technology
- Are willing to experiment and perform
- Are willing to perform within a group (co-creation)
- Are be able to hear sound, or at least feel it
- Have access to a mobile device with a compatible operating system
- Have access to a wi-fi network

Another important issue that will need to be addressed, is that the co-creative aspect itself means that our users can be both expert, casual, frequent and novice at the same time (I.e. a father using the application with an authentic child). If we are to follow basic principles of universal design, user testing need to consider issues like this, as well as on specific impairments in regards to requirements and usability issues.

At the bottom of our approach towards users lies a thought of equality. For us the goal will therefore be to design based on common trends, using common devices etc, for the sole purpose to avoid designing for an idea of disability, of which, could be a reason to reject the design (Plos & Buisine, 2006).

Music and Health

Most of us are familiar with evidence from research showing a positive effect between music and health. Indications in this field of research suggest that our understanding of music, as a means of therapy, should relate to health as something more than just biology. In the article “The Health Term – the self and the cell” (Ruud, 2009 [our translation]), Edvin Schei states that biomechanics is an aspect of human health, and that it is not possible to detach this from being interwoven with other things of importance in human life. Such as: Love, how we organise our society, education and music; all has a deeper meaning, also for the life of human cells. (Ruud, 2009). By taking a closer look at health from the two opposite perspectives phenomenological philosophy and psychoneuroimmunological research, Schei points to the fact that health in many ways are related to the question of what is good for the human being.

If we look at health from a sociological perspective, like Prof. Gunnar Mæland (*Ruud, 2009*), health could be understood as both a dynamic and subjective phenomena:

> *What has subjectivity got to do with health? The way health is understood in this way of thinking, is as a prerequisite for subjectivity, health is amongst others visible in it's ability to perform as a subject - with integrity and self-respect, in committing and nourishing relations. At the same time, health manifests it self through the experience of being a subject. Damaged health means that the ability to be a self, a person, is damaged. Or, on the other side - if the ability to be a person integrated in life is weakened, then your health is damaged. In this way health is connected by being the centre of your own existence, to create integrity, and to establish rich connections to others. (Ruud, 2009 - our translation).*

Music and health is according to this about self-esteem, relations, communication, ability to master and being part of a community on an individual level. Through The Music Impro App every participant are able to control individual instruments, which then makes it natural to imagine this as a tool for improved health as long as the subject in question is able to master the situated context of communication. The report "Music in psychic health work with young people and children" (Ruud, 2009 [our translation]) suggests a connection between use and control, where the users perception of actually controlling the music seem to have a positive effect. Our project try to build on this by giving the user control over which instrument they wish to play, and how they want to play it. Being designed for use in co-creative settings, the Music Impro App also makes it possible for each individual to create integrity by creating musical expressions, and be part of a rich productive relationship to other participants.

As mentioned in the introduction, one of the goals for the RHYME project is to use music as a tool for communication to prevent human isolation. Communication and the use of wireless technology opens up a realm of possibilities that challenges our notions about traditional place and space. Our project suggest that by building on principles from Universal Design, it is possible to adapt this technology to include users with disabilities, and thus seek to build a bridge between isolated individuals through the use of wireless technology. Building on the above, one of our goals for The Music Impro App is to play as a subject of research in this area with a focus on health and well being for users.

Music therapist Karette Stenset working with the RHYME project, has pointed out the necessity of bridging the gap between therapy contexts, and life outside the therapy context. (NRK Kulturnytt 2011). Our application is so far only tested in a LAN/WI-FI-setting by using mobile phones and a PC, but this technology is in principle open for use over larger networks like the Internet. This again opens up a new space and the possibility for co-creation on an asynchronous and synchronous level, pointing in the direction of a possible contributor bridging the gap between therapeutic and non-therapeutic situations.

Ethical issues

Adaption and implementation of new technology in the field of health research, inevitably raises ethical issues. As ubiquitous computing is becoming more and more integrated with human life on a global scale, it can possibly make it more difficult for us to see *the forest for the trees*, hence we should be aware of the necessity to sometimes take a step back for a better overview of what we actually are trying to achieve. It's seems obvious that technology isn't about what it *is*; it's about what it *does*. So how can we create the perspective needed to help us see the greater picture? If we follow Shceis definition of health as "what is good for the human being", it connects the more philosophical aspect of what would be the right thing to do. How do we think about implementation of technology in terms of cause and effect? This is obviously important when practicing Universal Design connected to health issues.

In the article "Look Who's Talking", Howard Rheingold (2001) makes a study of the Amish settlements in the USA. The Amish is well known for its refusal of modern technology, living a life that still resembles in many ways 19th century mid-western life. This community is interesting due to the fact that it is not a part of what we can call the Information society, thus it also seem to be an interesting case study as discovered by Rheingold. In the community's strive to keep its culture and integrity intact, the main question they raise towards new technology is as simple as; "does it bring us together, or does it bring us apart?" (Rheingold, 2001). Mobile technology is in many ways a great way of communicating, but on the other side, it is important to consider what implications this has for physical social relations. In regards to music and health, it will be important to actually know what implications lies within this realm of communication. Do we actually bring people out of isolation, or do we encourage less physical contact by introducing applications like The Music Impro App?

Review of literature

There has been done a great deal of research on music applications ,and how the mobile can be used by amateurs to create music. In a project called 'Mobile Gesture Music Instrument' Dekel, Amnon, Dekel and Gilly (2008) explore how the accelerometer axis can be used as a new dimension of music generation. They focused on using the phone as an easy to use musical instrument that amateurs can use to make music. In the same way that this project wanted to explore interaction methods that can be learned in a very short time, we in our project also want the users to be able to understand how it works by just playing with the application for a couple of minutes. In their project they test out two different applications. One of them is letting the users control the loops, tempo and effects, and the other application is letting the users use pre-recorded samples which are played depending on how they move their phone. We want to try to explore both of these aspects in our project. In their user study they found out that mostly the users enjoyed the first application best because it gave the users most feedback and control over what they did (Dekel et.al, 2008).

Another project that has explored the use of mobile devices for music creation is MoPhO (Mobile Phone Orchestra of CCRMA) (Essl et al, 2008). Their research is located around the term generic design, meaning a platform that is not designed for a specific performance but designs that are flexible and variable in use. Platforms that exists simultaneously with multiple and repetitive tasks and also universal to the extend that it allows a range of artistic possibilities. One of the challenges they mention to the generic use of mobile phones as a musical instrument is flexible local and remote networking for performances (Essl et al, 2008). In our project we want to explore the possibility of several users connecting their mobile phones through a wireless network for generic music synchronisation with the use of accelerometer and orientation sensors. We know that this will be our biggest challenge, because of the lack of existing material to work with.

Mobile music technology has been a growing research field within the last decade and because of the increasing popularity of context-aware systems and social networking applications the use of mobile phones as a music platform is growing. Embodied music cognition is a term that explores the idea where the bodily movement is an essential part of our understanding and experience of music. Jensenius (2008) in his article about ‘Music and Movement in Mobile Music Technology’ opens up some challenges related to music and movement in mobile technology. He talks about the difference between mobile (large external space) and immobile (small external space) and how this influence our understanding of action-sound relationships. What he means by this is that we mentally simulate how a sound was produced while percepting it. This relationship between action and sound has emerged as a very important research topic in HCI (Human Computer Interaction) and the exploration of new digital music – instruments. This is an important aspect in our own research project where we have to develop an application that both take into consideration the physical space (mobile-large external space) of the musical performance, and the response of the action performed by the user and thus the sound generated by the action.

Review of existing technologies

In the past few years the number of music applications available for smart phones has increased enormously. The development of new technologies in mobile services is opening for new possibilities for exploratory music-making. We are here going to do a short review of some of the most exciting music applications out there today.

Reactable *(iPhone, iPad ,Multi touch-table)*

Reactable is a new musical instrument that allows you to see the music while playing with it. Its unique combination of sound generation, manipulation, visuals and tangible interaction creates new expressive possibilities of traditional instruments for those who are working with new technologies. Reactable is a result of many years of research within the field of Human-Computer Interaction, and because the software and hardware components are designed together it differs from all the other existing technologies out there. The reactable interface is intuitive, tangible (physical blocks are moved around the multitouch-table), visual and collaborative. This multitouch-table allows several people to create music together using the same interface (Reactable, 2011). The way reactable opens the possibilities for starting loops, and adding different manipulative sound modules when collaborating, makes this an interesting application for inspiration for our own project.

Beatmaker *(iPhone, iPad)*

While reactable is an interactive instrument that could be used in a live-setting, beatmaker is more of an advanced home studio application. The application allows you to create beats and melodies using a range of different instruments, and audio effects. Beatmaker is structured much like similar recording applications like Logic and Cubase for stationary PC´s, but is easier to use for amateur musicians. Beatmaker allows the user to quickly record a beat using different samples, and easily record a new beat or a synth on top of that. It differs from reactable because of the user´s possibility of editing the recordings and the way interaction is conducted (http://www.intua.net/products/beatmaker2).

Gyrosynth *(iPhone)*

Gyrosynth is an music application that uses the gyroscope on the iPhone 4. The gyroscope is motion sensors that are build into the phone. It detects the rotation of the phone and opens up new interactive possibilities both for development of music and game applications on your mobile device. The gyrosynth uses these sensors to allow the user to control the frequency and modulation of the chosen synthesiser. One of the advantages of the gyrosynth is the immediate feedback to the user, but there are also several limitations of the application:

- Few available synthesisers
- Does not support co-creation
- Abstract sounds
- No synchronisation between phones
- Does not give the user the feeling of mastering and creating actual music

But because the application uses these motion sensors build inside the phone, the application is interesting for our project. We also want to build an application that uses the movement of the phone, but adding a new dimension by opening for co-creation between several people.

RjDj *(iPhone, iPad, Android under development)*

RjDj is an innovative application that introduces the concept of reactive music. It reacts to the sounds around you and creates new interesting sounds based on this. The app takes input from the microphone and analyzes and processes these sounds through different filters. It lets users dynamically create sounds and music from their environment. The sounds can be described as constantly evolving, atmospheric and rhythmic patterns.
When you use RjDj, you listen to what is called a “scene”, which is a collection of filters, processing and even sounds, that shapes the sound coming into the microphone. There are several different scenes included in the application, and more can be downloaded from the net. You can even create your own scenes through a desktop program called “RJC1000”.
This application relies heavily on voice/sound input, so it is easy to use for everyone. Even several people can together shape the sound, but it can be complicated for everyone to listen at the same time, as the app relies on the use of headphones. Another limitation of RjDj is that it is not able to generate other musical sounds than the one you make yourself, as it has no synthesizer built in.

Aura flux *(iPhone, iPad)*

Aura Flux is an ambient music creating application that lets the user dynamically create looping sounds by touching the screen. The user can draw lines and figures on the screen to shape the music. Sound is produced by different kinds of “nodes” that can be connected together. Nodes can either generate sound (generator node) or react to pulses (reactor node) coming from other nodes, and the nodes can be moved around the screen and be interconnected in many different ways. There are a total of 48 individual sounds to choose from, and the overall mood of a scene can be changed. Each node can be edited, and there are lots of parameters to edit.
Aura Flux has a lot in common with Reactable when it comes to the graphical interface and how you interact with the application. Nodes can be created and interconnected, and how they are placed on the screen has an impact on how they interact with each other. This is quite similar to how Reactable is designed. This application can be really inspiring to use, it is always easy to come up with something new, and the sounds and also the visual feedback is nice. However, there is no support for co-creation through connecting several devices and using them at once.

Seline/Seline HD *(iPhone, iPad)*

Seline is a musical instrument application for the iPhone and the iPad. It basically lets you play an instrument sound (like flute, cello, violin, synthesizer, etc) via the touch interface on the device´s screen. The interface is structured into different squares, representing the notes, and the user can choose a scale to play over. In this way, the instrument is easier to play, as it gets impossible to play “false” or “wrong” notes. This application is used by “The iPad Orchestra”, which features four musicians each playing one classical instrument on their iPad running Seline HD.

Bloom *(iPhone, iPod touch)*

Bloom is created in collaboration with ambient music pioneer Brian Eno. It features a simple interface that lets users tap the screen to create notes. The notes will play back in a looping manner, showing growing and fading dots on the screen as they play. Notes are organised into scales, and this can be selected from several predefined ones. The pitch of the notes is laid out on the screen going from deep notes on one side, to higher notes towards the other side. There are a lot of ambient effects on the notes you play, making it sound really pleasing and calm. The app also supports sensors; you can shake the phone to erase the notes.

Ocarina, Leaf Trombone (*iPhone, iPod touch*)

To play these two instruments, the user can blow into the device´s microphone and put the fingers on the screen to play different notes. The Ocarina is a flute instrument and the Leaf Trombone is, as you might have guessed, a trombone. In this way, controlling the instrument is done almost like if you were playing the real instrument. The interesting thing about these instruments is that users can connect to something called the “World Stage” and share their performances with others. Other users can rate performances, even in real time when another user is playing his instrument. This can be looked at as an interesting and innovative way of stimulating co-creation between users around the globe.

As this review of existing technologies show, the most exciting musical applications for smart phones are built on the iOS platform; iPhone, iPad, and iPod touch. When it comes to Android, the applications available are either based on the idea of creating a small “pocket studio”, a simple sequencer where you can put together different pieces of pre-recorded samples, or simple instruments like theremin or piano. So the selection of applications for music creation on Android is rather limited. After searching a bit around the web on this topic, one reason for this seems to be that there has been lacking a good development toolkit (SDK) for sound programming on Android. Also, there seems to be quite high latency on the sound output on the Android OS, and this makes it difficult to create good and responsive music applications. Nevertheless, there are several applications under development on this platform, and the guys at RjDj have some interesting things going on with their software that opens up for users programming sound patches for use on Android. In our project we have settled on developing for the Android platform due to practical reasons, since we had most experience with the Java programming environment. Due to its limitations compared to the iOS platform we found it suitable enough for concept prototyping.

Conceptual Model

In order to describe what the system is going to be like for the user, we’ll here try to conceptualize the design space. One approach to this is to perform an abstraction that gives a general idea of what the users can do with our application, and which concepts are involved to understand how to interact with it (Preece et al., 2007).

To understand what we intend to design, one must think of each device as a node in a network. The idea of co-creation will demand at least two or more nodes connected at the same time. All nodes are synced, and the system configuration does allow a node to make output in a different key than the rest of the nodes. Through the applications user interface a user should be able to choose an improvisation instrument. The user control their instrument and the sound it produces by movement of the device itself.

One node in the group will play the role as a master-node. The master node will typically be the rhythm section, with options related to musical categories, i.e. funky drum loop, jazzy beat etc. A traditional graphical user interface (GUI) will be supporting this node.

Interactivity within the group depends on co-creation in similar ways of a musical group or a band. One way to picture how the conceptual model is to be understood, can be done by imagining a scenario:

Just like in a band, a user have the option to be the rhythm section (controlling the master node), or control a guitar, piano or flue etc. What part you’d like to play in the band will be possible to access through the user interface or which device you choose.

Let’s say one user initiate a groove on the master node by choosing the funk category and starting a drum track. Then we have a loop with a funky rhythm going. Then the same person initiates a second loop, a bass ostinato chosen from the same funk category that plays together with the drums. A second user, holding a regular improvisation node, choose an instrument (like something sounding like a guitar with distortion) can then start to improvise upon this by moving the phone with the music and control the pitch and beats of the instrument by movement. The instrument sound is synced with the rhythm automatically. Then a third person, maybe choosing a different instrument, might want to move his phone around and create improvised sound upon the groove produced by the rest of the “band”. The instrument might play tones up and down a pre-defined scale (set by the system) and i.e. play the pentatonic scale upwards or downwards, depending on the movements from the person generating the output. The pentatonic scale playing will be matched to the basic chord progression running in the background, so that no matter how you move the phone, the scale will be playing in harmony with the rhythm section and the other improvisation instruments.

The conceptual model as described above is just to give a general idea of the concept and mental model behind our application, and what could be done with it by potential users. User testing with prototypes is part of the project to further investigate how well this concept worked with real users.

Design process

Based on an initial vision this project started with loose brainstorming sessions and technical reviews of what technologies and applications that existed. A large part of our group members had experience in playing music, but we had little experience in programming it. Instead of starting to work on concepts and paper mock-ups we felt it necessary to begin research which technologies existed and how music creation could most easily be conducted on mobile platforms.

Expert opinions

This first phase of the project has been mostly technical related. We have made technology reviews and developed a prototype framework that can be used to test ideas on users directly. We have talked with and got in contact with a lot of people in the field with expertise in music programming and mobile development. This has been teachers and professors, members of the RHYME project team, developers of relevant distributed applications, as well as fellow master students with the expertise we needed. Without all the great feedback and help the project would have wound up with a slow start. We did not meet anyone who had done music programming on the Android platform itself but some referenced to technologies like Max/MSP and Processing for programming on traditional computers. We were also recommended using Arduino as an early stage of hardware prototyping if sensor data served to be difficult to get from the Android platform. We had also help with the providing of example applications that showed us how we could get the different technologies to talk together. As soon as we had some understanding of the possibilities and limitations of the technology and a working framework for prototyping, we could start working on concepts, scenarios and interaction techniques and do user testing.

User Centered Design

We planned to use a user-centered perspective in the further development of our prototypes. Since we are dealing with sophisticated interaction methods without any clear design reference, we have found it beneficial to make high fidelity prototypes as working Android applications instead of starting with low fidelity paper prototypes (Kangas and Kinnunen, 2005). By using more realistic prototypes we hoped to gain more from user testing and get a clearer discussion around concepts and ideas that can be made more tangible and concrete than without. Since the development of the prototype was challenging, we only had time to conduct one workshop with users. In further development of this project a higher rate of user testing should be used.

Prototypes

In our initial meetings we talked about creating a prototype with Arduino to make a proof of concept, but when we found out how easy it is to get simple results with android, we bypassed Arduino and went straight on with android programming. The concept we had in mind required a phone app where the user got the option of choosing an instrument to play. This found app then needed to register sensor data and send this data over a WiFi network to a computer that produce the actual music. We decided on using Max/MSP as the basis for music programming.

We created two different applications. One application is intended to run on pads, or other large devices. This app is supposed to control the background music, i.e. the main beat, effects, tempo and other things that affect the overall experience of the music. The second app is the essential one. This is the one that allows the users to improvise their own music by using one of the predefined instruments and then moving their phone using different gestures. We developed both apps through several iterations with a different focus for each iteration. We will first take you through the development process of the improvisation app and the different directions we tried to take it. Thereafter we will take you through the process of developing the control tablet app. And thereafter the Max/MSP patch that generates all the music that are played while using our apps. As a working title we named the improvisation app “Crazy Impro App” and the app with control over the background music for “Crazy Impro App Boss”.

Crazy Impro App

Our first prototype was a quick and dirty application that just read out sensor data (Fig 2.), it was programmed in Googles AppInventor (Fig 1.) and didn’t really give us anything to bring further into the development process since AppInventor doesn’t provide source code.

To get started with the second prototype we got a little help from an American programmer through e-mail. He showed us how to make the app communicate with Max/MSP. We then wrote code that managed to read data values out from the accelerometer, the orientation sensor and the magnetic compass. These values were then sent over WiFi to the Max/MSP patch using code inspired by the American guy. We made it possible to choose from a range of instruments and set the IP-address of the computer running the Max/MSP patch. Our first functional prototype was made and it showed a technical concept that worked (Fig 3.).

In the next iteration of the application we did some changes under the hood, in the GUI and settings part. We had a problem with the last prototype that it utilized too much system resources. This was solved simply by complying to the Android programming Fundamentals (Android Fundamentals 2011). For the setting part, we added the possibility to obtain the IP-address of the Max/MSP host automatically by communication with a server script. We also grouped the screen elements together in a logical way, this was also done with the main window in the app (Fig 4.) Under the hood we mainly limited how often the application sends data over the network, and we did some work trying to detect certain movements with the accelerometer.

In the next iteration of the app we implemented a lot more internal magic and a better graphical interface. Internally we added a way to detect shake events, we implemented an algorithm which can detect shaking of the phone each 100 millisecond, and this enabled us to give the users several different opportunities to play music with our app. The user interface also got a completely new lock and feel. We added six icons to represent the instruments. When one instrument is chosen the background color changes accordingly, and the chosen icon wiggles on the screen to give running feedback on the chosen option and indicate what the user itself should do with the phone (Fig 5.).

One prototype version that we later decided not to use did not depend on GUI for choosing instruments. The screen just had one of six different colors and the color represented the current instrument (Fig 6.). Following our Universal Design approach we wanted to make a concept that did not rely on screen interaction at all, to include even users with vision disabilities. To change the instrument you only had to hold the phone close to another electronic device, like a phone or a television. If two people wished to change instruments they only needed to “bump” their phones together, and the app would manage the switch through communication with a server script. But using the magnetic field sensor in this way proved to be rather error prone. Users did not seem to like to be given a random instrument and it could easily change instruments several times by itself if you were in a room with a lot of electric wires. Using RFID technology for this approach may have been a better approach, but since this is not yet supported on most smart phones we decided to abandon the idea.

Final Prototype

In what we can call the last prototype of the Music Impro App, we did go in different directions on the layout, but let´s start out with the internal changes first. At this stage it was important to us to have different ways to play the different instruments so you had to do different things with the phone to play the different instruments. For example, to play the egg shaker it is sufficient to shake your phone to get the egg shaker sound. To play the Violin you have to change the speed of the phone moving. This sounds strange, but it enables you to use the phone as a bow that you use while playing the violin. For the Xylophone we used a combination of the compass and the accelerometer. The direction the phone is facing chooses the tone, and you can play the tone by using the phone as a stick “hitting” a xylophone. The R2D2 instrument generates a noise that becomes higher or lower depending on the phones x-axis, and a beeping sound that affects the noise depending on the phones y-axis. The piano would play continuously, but change tone depending on the phones position along the x-axis. The last instrument affects the other playing sounds when you move the phone around.

In the GUI we went with an advise that we got from RHYME, namely to get rid of the instrument representation in the icons. By using these cultural references we also have to expect that the users tries to map and evaluate the sound produced and the interaction used with the original instrument. This may give us unnecessary limitations on how the instruments best should be made and used. We tried to avoid this by representing the different instruments with color and graphical arrows that symbolised how you may move the phone in order to play the corresponding instrument (Fig 7.). In this way both identity and guidance were represented in the same icon. When pressed the main background changes color to match the chosen instruments and the instrument icon wiggled to indicate movement. After feedback from a user that thought the arrows represented touch gestures we added a drawing of a mobile phone on the arrows in one version (Fig 8.). We hoped this would make it clear that it is the phone that is supposed to be moved like the arrows.

The app we made could be installed on any device running the Android Platform 1.6 or later just by downloading an installation file.

Crazy Impro App Boss

In addition to this application we also developed an application that controlled the background music. The application was developed for tablets sized devices and relying on the graphical user interface (GUI). For testing purposes we borrowed a Samsung Galaxy Tab through the RHYME project. This tablet had a perfect screen size for our concept and used the Android platform. This app also went through several iterations during the development process. The pad application can as mentioned earlier control which loops that can be played, how fast they are to play and switch effects on and of. The first version had a simple GUI and were developed just to get the functionality in place (we don't have a screenshot of this one). After this initial version we didn't do much with the functionality. In the second iteration we added the possibility to sync all of the phones running the Music Impro App. This were done so that they would play synchronized. We later added a restriction to the Max/MSP patch which made this solution redundant and not needed. In the GUI we added grouping of elements and a lot of nice colors (Fig 9.). The last version of this app had a completely different layout of the GUI, instead of choosing a loop from a dropdown list and then adjusting effects, you chose a tab depending on which groove type you want, and then you get a page containing all the loops you can toggle and all the effects you can control for that groove. This enabled us to have completely different sets of effects to each groove type (Fig 10.).

Max/MSP

Max/MSP is a music programming environment that runs on both Windows and MacOS computers, created by software development company Cycling ´74. The Max/MSP patch for our prototype had an incremental development path following the apps. First off we were satisfied with just getting it to receive data from the mobile apps, then we expanded and made it possible to receive from ten different users simultaneously. It didn't take long before we had added a midi piano which played notes depending on the data it got in from the phones accelerometer x-axis. After this, the ball started rolling. And very soon the Max/MSP patch had become quite complex

(Fig 11.). It was soon able to play four different loops, and six different instruments. The way the different instruments were played was at that time quite simple, only playing notes over the whole musical keyboard range. But we intended to make it able to play chords and loops, and perhaps more sophisticated variations over the notes in each controlled chord to make it feel more like improvisation. Also, the patch synchronized the input signals from the phones with the playing loop, so it basically eliminated the use of a synchronization function on the phones.

The phones are outputting sensor data at a given rate, often quite fast, and this rate is not influenced by how the user move the phone when playing an instrument. To make the triggering of notes in Max/MSP more musical and not too chaotic, we made Max/MSP trigger the notes at quarter note intervals of the chosen beat. This was just a way of simplifying the synchronization of notes to the beat, as there are possibly more interesting ways of doing this.

For sound generation, Max/MSP was at first just using the computer´s built-in midi device, which contains a basic collection of musical instrument sounds. The quality of these sounds is unfortunately not very good, but for an early prototyping version it did suffice. One way of improving these sounds could be to connect a better sound module/midi synthesizer to a midi port on the computer.

Towards the final version of our application, we made a lot of changes in the Max/MSP patch. First, we changed all the instruments from just playing on the built-in midi device in the computer, to using either high quality samples that sounded much more realistic, or using sound programming in Max/MSP to create a synthesizer-like sound. This made the sound much more appealing and interesting.

Second, we changed the structure of the loop-playback mechanism. Instead of just being able to choose one out of several different loops at a time, we wanted the user to be able to choose between two different musical genres. So we made two different genres to choose from: funk and etno. After having chosen one of these genres, the user could now start a drum loop. In addition to this, we made three additional loops available to add to the already chosen drum loop; a bass groove, a guitar/sitar riff, and an electric piano groove (a percussive loop in the etno version). The user would be able to control these loops and the individual volumes in the application running on the Galaxy Tab. We made it so that these loops would always play in perfect sync, and that they sounded musically appealing when played together. We also added a filtering effect to the drum loop, to give the user a bit more control over the sound. All the loop tracks were made by using Apple´s Logic.

We also added a musical scale to the notes triggered by the different instruments, so that the notes played would always match a given scale and always sound “right”. This helped to make it easier to control the instruments, and gave a more musical result.

At last, we added an echo effect to some of the instrument sounds, to polish the sound even more and make it more appealing.

Ways of interaction - gestures and control

When developing our Music Impro Application, a question that often came up was: how can we control the different instruments in the best way? One of the things that we considered was the way a normal instrument is usually played. Most instruments relies on two controllers, or two parameters to control the sound. One being the pitch of the note being played, which can be either divided into steps of half notes, like the keys on a piano or the mallets of a xylophone, or there can be no such steps and a completely floating pitch, like on a violin or a cello.
The second parameter is the triggering of notes, like hitting a key on the piano or striking a string on the guitar. This parameter consists of a value indicating the strength of the note and, on some instruments, a time element indicating the length of the note until it is released.

When designing our application, we wanted it to be easy to learn and to play with. The users should not be forced to practice for a long time to master the instruments. For this to be possible, we had to simplify the way a user controls an instrument, and make some constraints to what can be done with the instrument. One way of accomplishing this, was to make the instruments play over a predefined, musical scale. So regardless of how “badly” the user controlled an instrument, the result would be no “false” notes and it would always sound tonally “right”.

On some of our instruments, we made a simplification to the triggering of notes so that the notes would not appear at random times and sound “chaotic”. We simply made the notes trigger in time with the beat, on each quarter or eighth note. On these instruments, the user simply controlled the pitch of the notes being played. After having discovered that constantly triggering notes sounded rather annoying in the long run, we implemented the possibility of stopping the notes from being triggered by holding the phone still for a moment.

For the xylophone instrument, we chose a more exciting way of controlling it. We thought of a xylophone player hitting the mallets on the instrument, and tried to transfer this type of movement to a gesture that could be recognized by the phone. We thought of the striking of a note as a shake of the phone, and the direction the phone is held could indicate the pitch of the note played. As when playing a real xylophone, the low notes are to the players left, and the higher notes are towards his right hand side. We implemented this by using the compass in the phone. Similarly, for the violin instrument, we thought of a player striking the bow on the strings back and forth, and implemented a shake recognition for this kind of movement.

In this way, we implemented various types of gestures and control of the instruments, and then we conducted a user test to gain some feedback on our application and how different people respond to playing with the instruments. The results from this user test would give us valuable insight to what elements of the application is functioning well, and what needs to be improved or even redesigned or discarded.

User Testing

We conducted one user-workshop based on our last prototype to try out our application on a number of different users. In total 14 people participated in the workshop. Because we want to follow the principle of universal design we invited users that consisted of two children (3 years old), their parents and students to get a broad range of feedback on our application. The user-testing was divided into two groups with the children and parents in one group, and the students in another. Each group was given approximately 30 minutes to test the application with as little instructions as possible. The people in our group had specific tasks during the testing. While some were in charge of observing the users, others were instructing and asking the users questions along the way. We used external computer-speakers to get the volume as high as possible and to be able to spread the sound into different directions in the room. Some of the questions we wanted answered in the user testing was:

- Does our application invite the users to co-creation, or do they prefer to play with the application by themselves?
- Does the users feel in control of the music being created?
- Do they get bored with the application or do they want to keep on playing?
- How do they respond to the user-interface? Do they prefer the touch interface on the pad-application or do they prefer the sensor-interface?
- How does the users respond to the graphical user interface with the “arrow-buttons”? Are they trying to replicate the gestures, or are they moving the phone randomly?

Group 1 - consisting of two children and their fathers:

Two mobile phones with the sensor-application was given to the children´s parents, and they was asked to invite the children to try the application (Fig 12.). At first the children was very shy, and did not want to try it out. But after watching their parents try it out for a couple of minutes, they started touching the buttons on the phone. It was clear that the kids liked to touch the buttons, instead of moving the phone, even though their parents instructed them to move the phone. When they did move the phone, the gestures were only short shakes. One of the parents said after the workshop: ”He’s used to hearing ‘you have to be careful with the expensive cellphone’ at home. So he did not dare. I could tell from just looking at him” (father, our translation). After a while we gave the parents two mobile phones as well. When both the kids and the parents had one phone each things got more interesting. The children tried to imitate what their parents did. They chose the same “instrument” as their fathers, and they started to communicate more. The focus of the children was still on touching the buttons, while the parents tried different types of gestures. It was hard to tell if the children understood what happened with the music when they moved the phone, or if they just touched the buttons because it was fun. At the end of the workshop we gave one of the children the Samsung Galaxy Tab to try out. This was obviously more fun. Because the GUI consisted of buttons and sliders the kid looked more comfortable and tried different things. He also agreed with that it was easier to understand what happened with the music when he used the pad.

Group 2 - consisting of 10 students:

The students that participated in the user testing was students at the department of informatics at the University in Oslo. Because of poorly given instructions from our group, all of the students came at the same time and everyone started to download the application and try it out simultaneously. This made the whole session a bit chaotic, but we managed to get some results anyway. When observing the students using the application it seemed like everyone was trying to figure it out by themselves. They all stood in front of the speakers and tested the different “instruments”, and tried to figure out what sounds they were controlling, and how they controlled it. Because every sound came out of the same speakers it was hard to distinguish them. If they tried the application without anyone else playing, it was easier to understand what sounds they were making and how they controlled it. The sounds that the users thought was the easiest to distinguish was the “rhythm egg” and the “droid”. This was something we had expected.

While some of the students wanted more control over the music, one student said: “More control but not at the expense of how easy it is” (male student, 23, our translation). Some students that tried the pad-application liked it better than the sensor-application because they felt more in control. When asked if they wanted to use the application after the user testing, some said that they would not use it if they were alone but maybe in a social setting with some of their friends. One user said: “Would have been cool to connect to someone on the other side of the world and hear their beats on top of what you yourself played. Kinda like world music - It would be cool!” (female student, 28, our translation). This comment supports our vision of making an application that invites people to play music together without necessarily being in the same room. Most of the users thought the “arrow-interface” looked good and gave them instructions on how to use the application, but they tried to follow the drawings without really understanding how to do it. Some of them thought something specific would happened if they managed to replicate the gesture correct and they therefor tried to repeated the gesture over and over again in different ways.

One of our goals with the user testing was to look at how they responded to the different types of interactions (gestures and buttons). It was obvious that there was a big difference between how the kids interacted with the application and how the students interacted. The children was more in favor of touching buttons, while the students seemed to enjoy the use of gestures even if they didn’t quite understand how the gestures affected the music. We chose to develop the application using the sensors inside the mobile phones to use a more exploratory design approach and at the same time was easy to use. The challenge is “not to only develop new technology and interaction techniques, but to be able to understand possibilities generated by these techniques” (Enquist et al., 2008). After having the user tests we see great possibilities and advantages using sensor technology to control sound for some of the users, but our challenge is to improve the accuracy and feedback generated by the system to further expand the feeling of user control. As also mentioned in the “Memory Stone” project, the participants using the gesture controlled system used a high diversity of movements. One of our challenges for further development is to use a flexible interpretation of how to interpret a gesture or perhaps use a combination of both touch and gestures (Enquist et al., 2008).

Discussion

Constraints on Mobility

To be able to make a working prototype in the short time frame, we settled on using a system depending on a computer that generated the music and presented it trough speakers. The mobile terminals would then be used as the main form of interaction with the system and communicated with the computer trough the wireless network. If you look at a computer with its sound system as stationary artifacts, the mobility of the concept is restricted by a unnecessary linkage between the architecture and the activity since the context required to do the activity is based on stationary devices (Agre, 2001). If the users want to do the activity of playing music with our system they now need to be on a specific place with a specific hardware to do it. You can argue that a laptop is mobile and a mobile network could be used for communication, but as of now we think that if our concept should become useful in everyday life and practice it should not be dependent on other hardware than the mobile devices themselves. We feel the activity of music co-creation is better linked with an institution, as the family, friends, or a client-therapist relationship. These forms of institutions are often as mobile as the cellphones they carry and the system should not be restricted by other contextual aspects.

An important aspect for future work would have been how to make this concept truly mobile, possibly by letting the mobile devices produce their own sounds and have the synchronisation of beat happen over networks like *Bluetooth* that does not require stationary hotspots other than the cellphones themselves.

A mobile app as a music instrument

We wanted to have the users to respond to an understand the application as a music instrument. This produced some challenges for us, because music instruments and mobile applications can be thought of coming from two different artifact genres with different aspects of use and different contextual roles. You may think of mobile applications as artifacts that are used individually, as information tools or entertainment, but seldom as a way of communicating with people that are present in the room. Music instruments as artifacts on the other hand may be understood as having a more collective use in the sense that they are used in collaboration with others or used alone but with an audience as target. By using terms from Brown and Duguid (1994) you may say that the use of a mobile application as a music instrument is a borderline issue. Since “aspects of an artifact regarded as part of the canonical center from one standpoint may be regarded as quite peripheral from another” (Brown & Duguid, 1994) we may anticipate problems if our users understands our application as something else than we intend. We use movement of the device as our main form of interaction, something that is not that common in mobile applications. How can we let our users anticipate this unorthodox way of interaction? Would it be easier to anticipate this if the application is thought of as a music instrument? In what ways may we present the application so the users will understand it as a music instrument?

Our first prototype tried to combat some of these borderline issues by using graphical illustrations of traditional music instruments as icons for the users different choices. In this way we hoped to make it apparent that the application was mimicking aspects of this instruments and that it should fulfill the same role. When selected the icons also wiggled from side to side to indicate movement.

This seemed to help tackling some of the borderline issues since our users seemed to understand the purpose of the app. But some did not see how this peculiar instrument was supposed to be played. Julie, a 7 years old test user, tried each instrument by wiggling the device from side to side and stated it as boring after 10 seconds. We also got complaints from users that thought the produced sounds did not sound like the instruments that were illustrated and were annoyed by this. Professors from the RHYME project pointed out that by using traditional instruments as metaphors for our own instruments we should also anticipate that the users would to move the device in ways that would mimic how the traditional instruments would be played. As mentioned in the prototype chapter we tried to avoid this by changing the instrument icons into arrows that visualized how this instrument might be played. This no longer made it apparent what purpose the app was for, but it seemed to help users understand in what way the device was supposed to be used. Once users understood the purpose of the app they did not seemed to miss the strict instrument metaphors the older icons provided.

Technical Future Work

We now see a lot of possible technical improvements that can be made to improve this concept. The most prominent improvement is to have the functionality of the Max/MSP patch embedded in the mobile application, so that the mobile device itself will be able to generate sound and not having to connect to a central base computer. In this way the system will be truly mobile. As far as we have discovered, there is no way of running a Max/MSP patch on Android, but it seems that running a patch from PD (Pure Data) is possible. It should be possible to convert the Max/MSP patch into a PD patch to accomplish this. Beyond the mobility improvement this will also make it possible to have the sound of one instrument being controlled coming out of the speaker of the phone that is playing that instrument. In this way, a person playing the xylophone will hear the xylophone sound coming from his own phone, and not from the base or the pad-application. This should increase the feeling of actually controlling one particular instrument. We would also have had to work on each “instrument” to get a more distinct sound on each one of them and improve user control. To make the grooves and beats more dynamic and interesting, and to give the user more control over the rhythmic elements, a feature could be added that enables triggering drum fills/breaks from the pad-application.

There are also a number of improvements and optimizations that could be done to the way the application handles gestures and movement. Although we have come a long way in optimizing the shake recognition and the implementation of the shaker instrument and the control of the xylophone, the response should possibly be made even quicker, and the control could be more accurate. Part of this delay will go away when we can put the sound generation on to the phones, because then we wont be affected by the delay caused by the network, but there are still possible to make better algorithms that can detect when a shake is about to happen. If you draw out a graph from your accelerometer, you will see that you get a negative spike when you start to move your phone. This spike can be used to calculate when it is most likely that the positive spike (the real shake) comes and we can start generating the sound before the actual shake happens. This will perhaps give a more precise feel to the app.

Conceptual Future Work

More research should be conducted to study how this concept may facilitate co-creation in a better way. Is there interactive elements or other aspects of the concept that may be altered or added to encourage communication, collaboration and co-creation between the participants? Should more of the control interface be dependent on interaction between users, like “bumping” phones to switch instruments? Or will the technical improvements specified over also give rise to a more collective use of the applications?

A broader look on which type of gestures and movement to support in the music creation would also be beneficial. Users carry a lot of different expectations that could be useful to detect to make ways of interaction that is experienced as intuitive. Which forms of gestures do users feel is the natural way of controlling pitch? Should it be up or down, rotation, speed or something else? What do people expect will happen if they play with stronger force or just “go wild” with an instrument? Should it produce more volume, more noise and bends, or more tones?

More user test will also reveal the different needs we have to design for. Some users will play with more gentle movements than others, and some users will have disabilities that prevent them from doing gestures others can. Should the device adapt to how it is used automatically or should it be possible to make different user categories to choose from?

Concluding Remarks

One of our goals in designing a mobile music application using gestures and movement was to enable and inspire people to play music through co-creation. Through several rounds of iterations on the high fidelity prototype we managed to develop a working application based on the Android platform, using Max/MSP running on a computer for sound generation. Because of the time consuming and challenging aspects of prototype development, less time was devoted to user involvement in the design process. This has in many ways influenced the resulting prototype and identified many aspects that should be addressed in later development. But the high fidelity prototype have identified many technical limitations and also proved the concept to be technically feasible. Results from our user study shows positive feedback on the socializing aspect of the application, by inviting people to have fun together instead of creating music by themselves. Users seemed to like the concept and they appeared to approach the prototype with positive expectations. Many challenges met in the workshop conducted with users concerned the balance between user-control and the simplicity of the application. Some users did not fully understand the relationship between the interaction and the sound-output generated. This balance is something that should be thoroughly discussed and worked on if the application was to be further developed. We hope this project may inspire research and further development of this types of applications. We may have a couple of findings, but mostly a bunch of unsolved questions waiting to be studied further.

Thanks to

- *The RHYME group for support of hardware and software that we could not be without, and important advice and guidance on conceptual and technical challenges.*
- *Even Bekkedal and Hoang Viet Phi Uy on advice and guidance on music programming and sharing of patches for reference and inspiration*
- *The participants on our workshop for their time, interest and valuable feedback*

References

- *About RHYME (2011), http://rhyme.no/?page_id=10, Accessed online March 17. 2011*
- *Android Fundamentals (2011)http://developer.android.com/guide/topics/fundamentals.html Acceced online February 20. 2011*
- *Agre, Philip E. (2001) Changing Places: Contexts of Awareness in Computing, Human-Computer Interaction, Volume 16, pp. 177-192*
- Brown JS and Duguid P (1994) *Borderline issues: social and material aspects of design*, Interactions.
- *Dekel, Amnon, Dekel, Gilly(2008) MogMI:Mobile Gesture Music Instrument. Mobile Music Workshop´08.*
- *Enquist, Henrik, Tollmar, Konrad(2008) The Memory Stone - A personal ICT Device in Health Care. NordicCHI 2008.*
- *Essl, Georg, Rohs, Michael,Wang, Ge, (2008) Developments and Challenges turning Mobile Phones into Generic Music Perfomance Platforms. Mobile Music Workshop´08.*
- *Garrett, Jesse James (2011) The Elements of User Experience: User-Centered Design for the Web and Beyond. New Riders: Berkley.*
- *Jensenius, Alexander-Refsum(2008) Some Challenges Related to Music and Movement in Mobile Music Technology. Mobile Music Workshop´08.*
- *Kangas, E. , Kinnunen, T. (2005) Applying user-centered design to mobile application development, Communications of the ACM , Volume 48 Issue 7, July 2005 (http://portal.acm.org/citation.cfm?id=1070866)*
- *Mace, Ronald L, Graeme J. Hardy, Janie P Place (1991) Accessible Enivironments-Toward Universal Design. The Center for Universal Design.*
- NRK Kulturnytt 22.02.2011 Edited version produced by NRK (norwegian) http://dl.dropbox.com/u/15788242/2011-02-22-kulturnytt-rhyme.mp3 (Accessed online Mai 2011)
- *Plos, A., Buisine, S (2006) Universal design for mobile phones: a case study , ACM.*
- *Reactable (2011), www.reactable.com .Accessed online March 20. 2011*
- Rheingold: *Look who's talking*, 2001. Wired magazine 7-01.
- *Robertson, P. (2001) Music and health. In: A. Dilani, Editor, Design and Health—The Therapeutic Benefits of Design, Elanders Svenskt Tryck AB, Stockholm (2001).*
- *Ruud, Even (2009), Musikk i psykisk helsearbeid med barn og unge, NMH-publications, ISBN: 9788278530603*
- *Sharp, Helen, Preece, Jenny, Rogers, Yvonne(2007) Interaction Design- beyond human-computer interaction. John Wiley & Sons*

Appendix

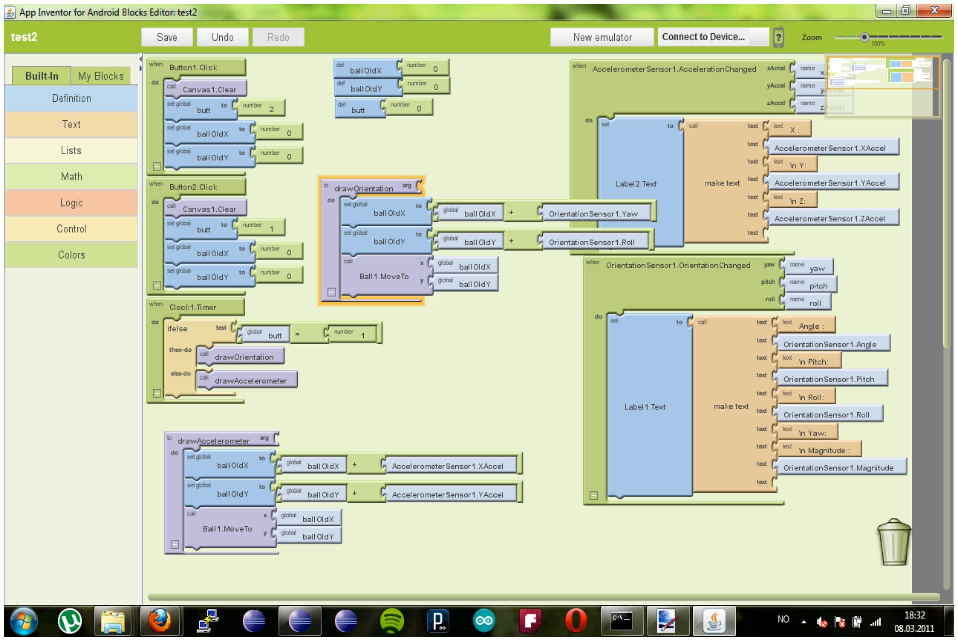

Fig 1. Screenshot from the AppInventor window

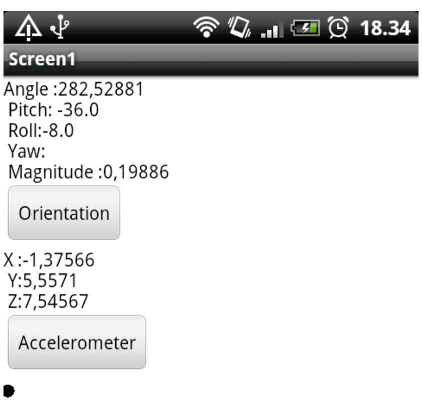

Fig 2. Screenshot from the first prototype based on AppInventor

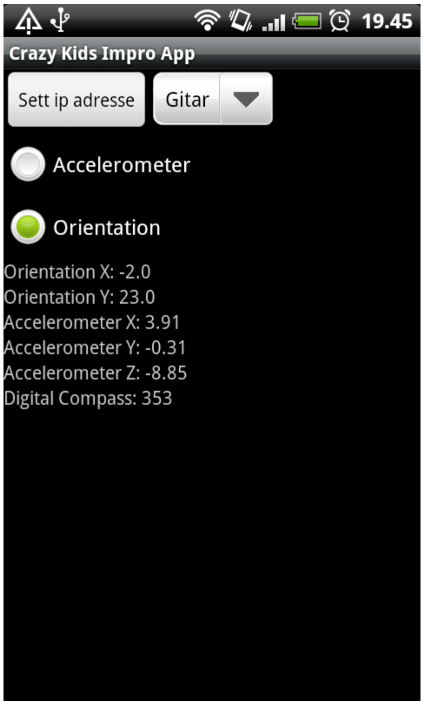

Fig 3. Our first functional prototype

Fig 4. Prototype with grouped layout elements

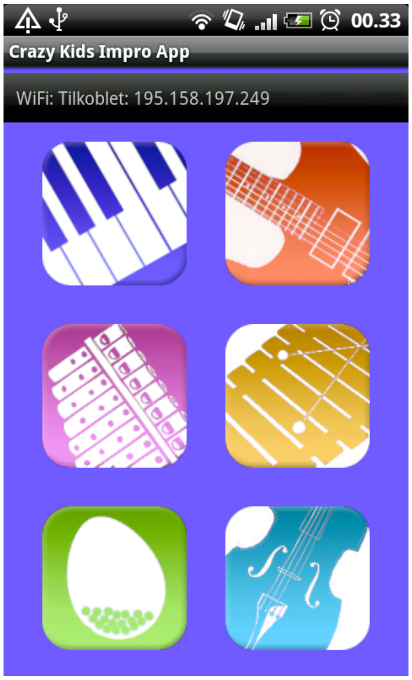

Fig 5. Screenshot of prototype with new icons

Fig 6. Screenshot of a prototype version without buttons

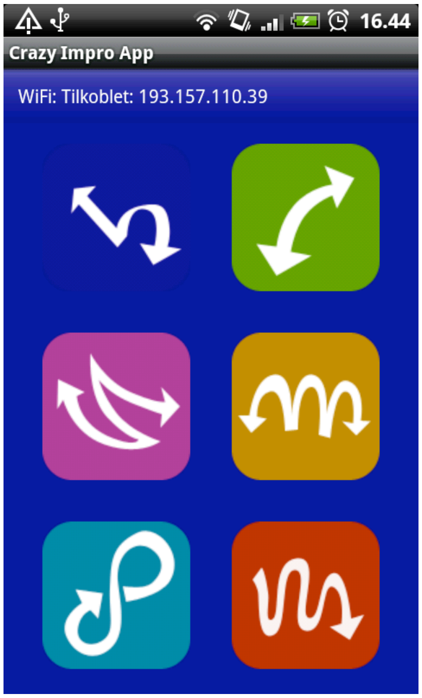

Fig 7. Screenshot of the Impro App with arrow representation

Fig 8. Screenshot of the Impro App with arrow and device representation

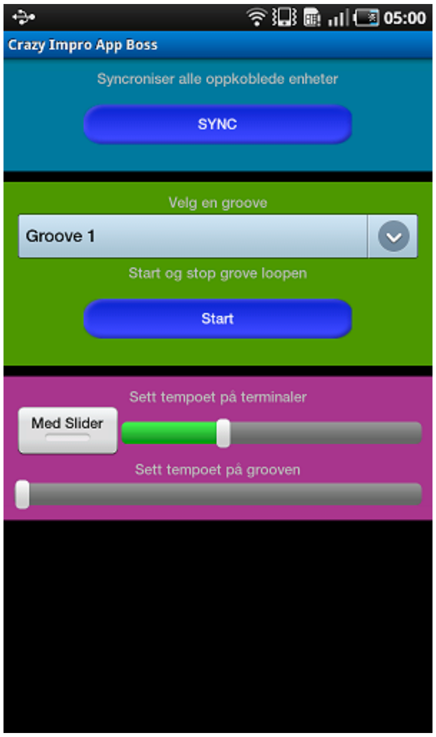

Fig 9. The “crazy impro app boss” with colors and grouped layout

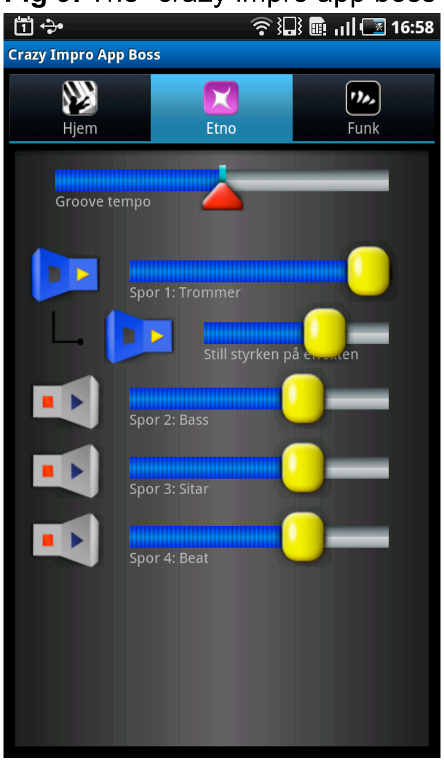

Fig 10. Crazy Impro App Boss with new design based on customized graphical elements

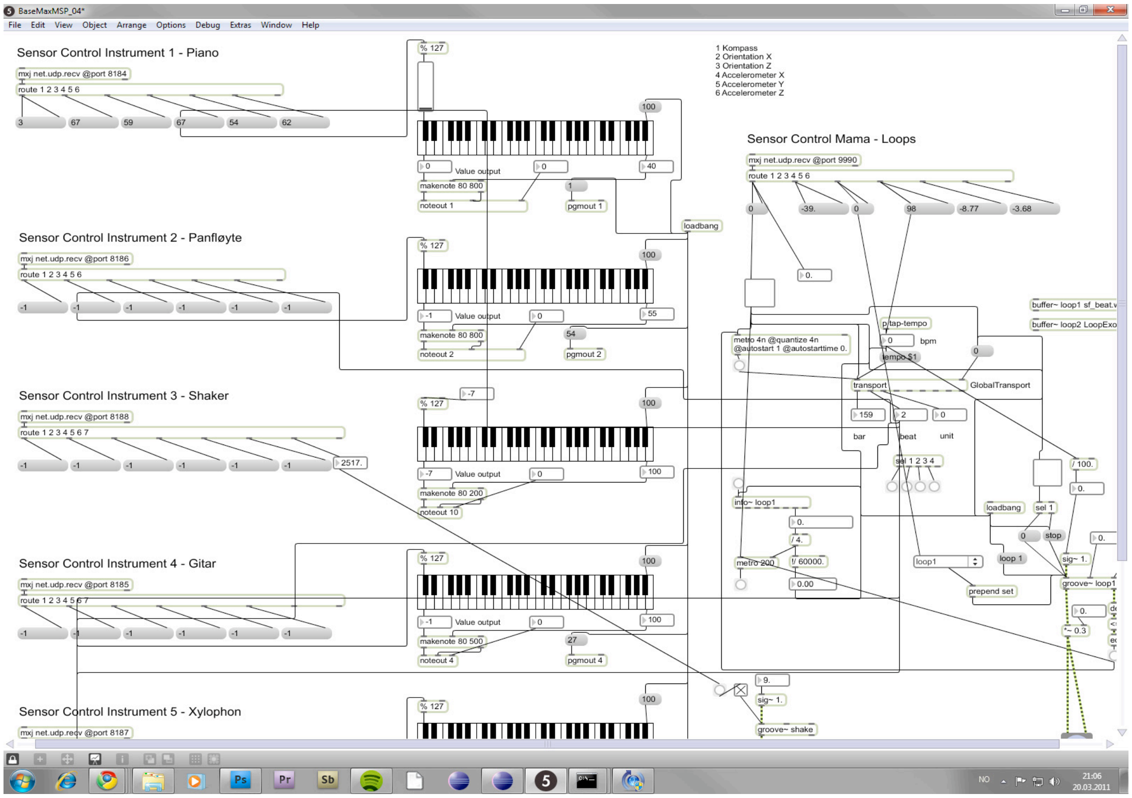

Fig 11. The messy layout of an early Max/MSP patch

Fig 12. A picture from our user-workshop with children and their fathers