Product Requirements Document

KeyboARd

Chieh (Josh) Chang - iamjoshchang@gmail.com
Seokmin Kim - Seokmin.Kim95@gmail.com
Hayoun Oh - hayounW@gmail.com
Matthew Robinson - MattLRobinson7@gmail.com

Apr 12, 2016

Summary

Learning how to play the piano is hard. There are tons of skills and knowledge one has to learn to become a musician, such as hand-eye coordination, music theory, sight reading, and a way to combine all those together at the same time. No wonder why there are so few people who can play piano well. Many whom have tried to learn the piano fail. However, we might see a huge change in a near future. Introducing: Keybo**AR**d, the first ever augmented reality app that puts a virtual layer of keys onto a real piano and teaches all-level players how to play their favorite songs. By providing visual aids in the form of music flowing down to your keyboard and your keys lighting up, players will be able to practice the songs that they want to play while enjoying it at the same time.

Deliverables

By the end of the project our goal is to be able to create an AR application that allows users to sit at a keyboard and use our app to practice any of the songs that we have provided for them and/or learn how to play the piano by going through our beginners' tutorial.

In the demo, you will be able to 1) get initiated into the art of piano by following the beginners' tutorial, 2) enjoy playing multiple songs with an intuitive, game-like interface, and 3) have overall better experience playing piano, inspired by your augmented surroundings.

Critical Features

- **Music Sheet to Text Conversion** - Using ABC standard notation, we aim to create a parser to read the ABC Notation and play any song we add to the application. This will make future updates easier and faster.

- **Correct Piano Mapping -** With our application, users will have an additional layer of virtual keys on top of the actual keyboard they own. In order to let the virtual keys with varied color serve as an instructions for users to play a song with the piano, we will find a way to overlay the virtual keyboard in the correct place and in a proper scale. Our ultimate goal is to have an automatically scalable mapping system, but for the current purpose, manually manipulated mapping between a virtual and physical keyboard is more than enough.

- **Raining Notes** - As a visual cue on what notes are coming next in the music, we want to show notes that would 'rain down' from the top of the interface down to the keyboard, to notify players when and which key they should hit.
    - There might be problems with the hololens and its Field of Vision not being large enough, however we can try to rectify the solution by having the notes come from on top of the users vision and they can easily and naturally follow where the notes go

- **Keyboard Highlighting Signals -** To provide further visual aid to users when hitting the keyboard, we want to highlight the specific key on the piano that is pressed, or to be pressed.

- **Tutorial Hand Modeling** - To teach users the proper hand shape and fingerings when playing the piano, we intend to adopt a hand model to demonstrate the proper way to position hands to hit certain keys.

- **Time Adjustment -** Due to lack of time to implement finger motion tracking, we adjusted our goal to get input from a MIDI keyboard. We will also allow players to slow down the speed of the music for themselves.

Performance Metrics

We will evaluate our demo by its:

- **Accuracy** - How precise is the placement of the 'notes' and 'virtual keys'? Does it well match the placement of the physical keyboard?
- **Stability** - How well do the 'notes' and 'virtual keys' stay in place? How do they react to the sudden head movements of a user? How long can a user play our application without experiencing motion-sickness?
- **Effectiveness** - How effective is our application in helping the learning process of beginners and intermediate players? How enjoyable is our application?

Milestones

- Week 1: Collect ideas and finalize the main concept of the product
- Week 2: Write a project proposal and present the idea to experts in order to discuss possible risks and get advices
- Week 3: Test feasibility of piano app and test all possible limitations that could cause the app to be abandoned; Setup development environment; Make a specific list of features to implement

    - Chieh (Josh) Chang - Work on PRD, Experimenting with the field of vision issue
    - Seokmin Kim - Creating a 3d keyboard using Unity
    - Hayoun Oh - Experimenting with 3d mapping in Hololens
    - Matthew Robinson - Experimenting with key highlighting; Editing the video clip for the team
- Week 4: Create parser to convert music notation to usable data and begin trying to map virtual keyboard to real keyboard
    - Chieh (Josh) Chang - Scripts for parsing simple text songs
    - Seokmin Kim - Tracks for musical note rain, look into vision APIs and begin vision work.
    - Hayoun Oh - Tracks for musical note rain, look into vision APIs and begin vision work.
    - Matthew Robinson - Scripts for parsing simple text songs
- Week 5: Finish keyboard mapping. Work on being able to generate music to flow down and highlight keyboards when the notes flowing down
    - Chieh (Josh) Chang - Start working on script for ABC Notation
    - Seokmin Kim - Work on Vision and calibration for piano
    - Hayoun Oh - Work on Vision and calibration for piano
    - Matthew Robinson - Finish Script for simple songs
- Week 6: Complete the visual aids. Depending on how far we are, create a system that would read music files and create songs that we can play
    - Chieh (Josh) Chang - Finish Script for ABC Notation
    - Seokmin Kim - Vision finish, Begin work on Visual Aids (Music Rain)
    - Hayoun Oh - Vision finish, Begin work on Visual Aids (Music Rain)
    - Matthew Robinson - Finish Script for ABC Notation, Begin work on Visual Aids (Music Rain)
- Week 7: Finish the music system. Work on hand model manipulation to manipulate a hand model to be able to give tutorials
    - Chieh (Josh) Chang - Begin working with hand model for tutorials
    - Seokmin Kim - Finish Visual Aids
    - Hayoun Oh - Finish Visual Aids
    - Matthew Robinson - Finish Visual Aids
- Week 8: Plan out basic piano lessons that we can teach and implement scenes in Unity that could go through the lessons. Finish Hand Model manipulation
    - Chieh (Josh) Chang - Hand Model for tutorials
    - Seokmin Kim - Begin Midi integration
    - Hayoun Oh - Work on tutorials
    - Matthew Robinson - Begin Midi integration
- Week 9: Integrate hand models to our lessons to show beginners the proper way of positioning their hands.
    - Chieh (Josh) Chang - Add feature to change speeds of songs to learn
    - Seokmin Kim - Midi Integration
    - Hayoun Oh - Add feature to change speeds of songs to learn
    - Matthew Robinson - Midi Integration
- Week 10: Bug fixes/prep for demo
    - Chieh (Josh) Chang - Bug fixes
    - Seokmin Kim - Bug fixes
    - Hayoun Oh - Bug fixes
    - Matthew Robinson - Bug fixes

Materials and outside help needed

- Hand models - To teach beginners how to position their hand
- A piano keyboard

Budget

For example to buy 3D assets, software, or hardware. Specify costs to be approved by the class staff (up to $1,000).

- A hand model for lessons($25) https://www.assetstore.unity3d.com/en/#!/content/14068
- Real Electronic keyboard (~$100-200) http://www.zzounds.com/item--CASCTK6200

Risks and how risks will be addressed

Our biggest risk is that our app is not possible to implement and will be limited by the hardware when in use. To address this problem we aim to test everything that can and could go wrong first so that we have enough time to try several different approaches.

Our risks are:

- Headaches caused by being too near the holograms
    - We can try and make it a standing keyboard instead
- The Field of Vision is too small for our piano
    - We can make the music flowing down down appear in the user's vision and then flow to where it should go, indicating the player to follow along
- We can't accurately map the virtual keyboard to the real keyboard
    - Automatic mapping might not be possible, but we can try to move our keyboard to where our virtual keyboard is located.

After we solve our feasibility problem the rest of the work has been already been proven to exist when looking at the history of video games. Thus, the only limitation is our own ability to implement.

Luckily our project builds on each other. Before we can create our piano tutorial we have to be able to already have music flowing down, before we can have music flowing down, we need to have a keyboard mapping.

In the event that we stumble on our tasks we can cut our tutorial feature and aim to focus on creating a better UX for playing songs possibly adding more features to enhance the playing experience.