Team 4

Kari Cieslak, Jakob Wulf-Eck, Austin Irvine, Alex Crane, Dylan Vondracek

Project SoundAround

Contents

Figures

Synopsis

AR headset music synthesis application, allowing users to create music by 'wiring up' nearby objects, which produce sounds and effects based on their physical properties.

Description

This project is being undertaken to explore the rapidly-developing augmented reality space and its potential intersection with music production and real-word/tech interface interaction. Once AR technology is widely adopted, tools like SoundAround could allow consumers who own AR devices to experiment with making music with very little barrier to entry. Tools like SoundAround could also serve as an example of interfacing the real-world environment with applications in general.

AR can provide musicians with a highly-intuitive user interface which is easier to learn and interact with than existing PC-based music production software. AR music production tools would remove the cost and physical space overhead of purchasing dedicated equipment -synthesizers, MIDI keyboards, instruments -enabling users to make music using full-featured tools through a device they already own. The incorporation of physical objects into the application -as 'modules' which will have sometimes-unexpected effects on a user's creation -adds elements of fun and exploration to music creation.

Additionally, it is an embodiment of one of the most fascinating aspects of AR -the way in which, unlike video games or VR, AR encourages its users to interact with the real-world environment around them. Rather than obscuring or replacing reality, it enhances reality and allows users to see it and interact with it in new ways.

SoundAround will use the AR device’s inbuilt object recognition (and supplemental SDKs if necessary) to discern between different objects, and will then determine what appropriate sound is created or played based on that object’s properties. The application will apply diegetic user interface principles in order to create a 3D user interface that blends in with the environment and real space, as opposed to a standard user interface that would be imposed onto a flat plane on the screen.

Ultimately, our goal is for this project to result in a robust, fascinating, and accessible music creation tool. With intriguing visuals, an intuitive interface, and an immersive concept this instrument will allow our customers to transform the world around them into music.

Milestones	Target Completion Date
Fall Semester	
1. Produce Formal Proposal and Demonstration Video 2. Train With Software and AR Headset (Microsoft HoloLens) 3. Integrate MIDI with HoloLens and Unity 4. AR 3D Module Design (Recognize Object Properties) 5. Audio Module Design (Sequencer and Synthesis) 6. Create Unity project, recognize limited set of objects	1. 10/29/2018 2. 11/11/2018 3. 11/12/2018 4. 12/08/2018 5. 12/08/2018 6. 12/12/2018
Spring Semester	
1. Build Algorithms For Object and Sound Association 2. Build Algorithm For Sound Effects with Texture 3. Incorporate Flexible User Interface 4. Successfully “Play” Test Room, Able to Demonstrate 5. Expand Catalogue of Assets, Incorporate Assets 6. Final Report and Promotion	1. 2/17/2019 2. 2/10/2019 3. 3/02/2019 4. 3/08/2019 5. 4/05/2019 6. 4/15/2019

Budget/Materials

- Microsoft HoloLens Development Kit (MHDK) - Loaned From Dr. Brian McClendon
    - Head-mounted AR device with head movement and gestural control
    - Supports application development with Unity, Vuforia, Visual Studio
- Windows Development Machines meeting the following specifications for HoloLens emulator (Machines meeting these specs are likely available in the capstone lab.):
    - 64-bit Windows 10 Pro, Enterprise, or Education w/10 April 2018 update
    - 64-bit, 4+ core CPU
    - 8 GB+ RAM
    - BIOS/motherboard supporting:
        - Hardware-assisted virtualization
        - Second Level Address Translation (SLAT)
        - Hardware-based Data Execution Prevention (DEP)
    - GPU supporting:
        - DirectX 11.0 or later
        - WDDM 1.2 driver or later
    - Necessary software
        - Visual Studio 2017
        - Unity 3D LTS (Long-Term Support) release
        - HoloLens Emulator
- Korg MicroKey Air 25 MIDI Keyboard (Bluetooth) - $115
    - Entry-level 25-key MIDI keyboard for synthesizer control
    - Bluetooth support necessary to interface with HoloLens, but uncommon - a previous AR music research project used this keyboard
- Project Management
    - Trello - issue tracking
    - Slack - team communication
    - Git/GitHub - version control
- Training
    - Tutorials available for free
        - Unity 3D documentation
        - HoloLens API documentation
- Useful APIs
    - Object Recognition
        - Vuforia Engine - AR computer vision library supported by HoloLens/Unity
        - OpenCV - More general computer vision library
    - Music Synthesis
        - Pure Data - Visual synthesizer programming language w/Unity plugin available
        - Blunt - Unity procedural audio library
    - AR

    - Microsoft Mixed Reality Toolkit - Utility library adding additional components/scripts to Unity for AR development

- **Total Estimated Cost -- $115**

General Work Plan

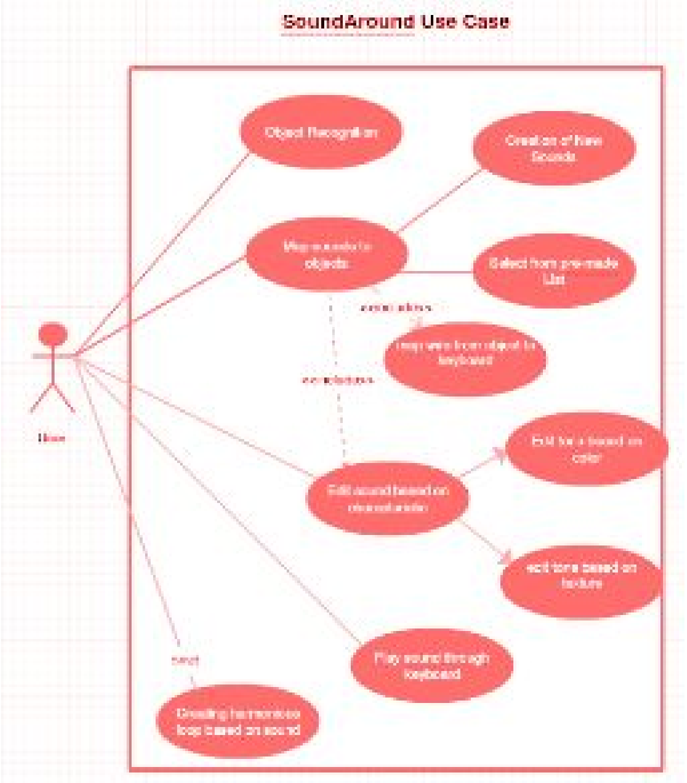
Jakob: Image Recognition Development, Audio Synth Design
Alex: Unity User Interface Development, Image Recognition Development
Austin: Graphics/Modeling, Audio Synth and Sequencer Design
Kari: Unity User Interface Design, Audio, Documentation/Deliverables
Dylan: Image Recognition Development, Audio Algorithm Design
Note: All members will contribute to production of project deliverables, design decisions, and feature testing.

See Gantt Chart Further Down For A Detailed Outline of Work

Preliminary Project Design & Design Constraints

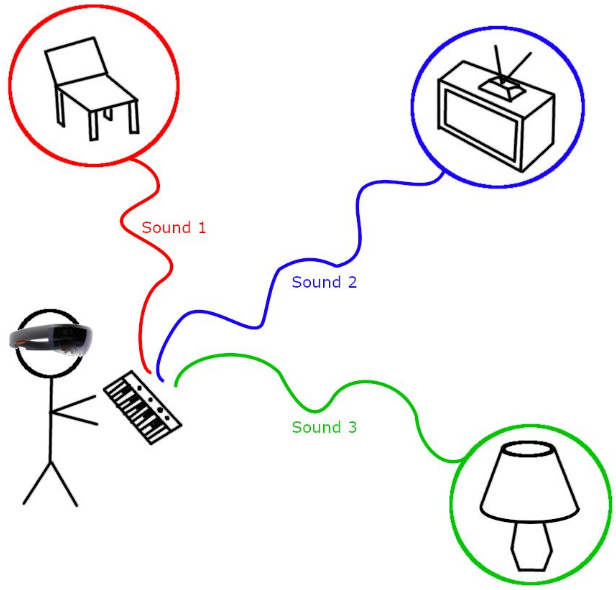
SoundAround's primary goal is to provide a unique experience that allows the user to interact with their environment in an unfamiliar way, interfacing music creation with real-life objects. This starts with first recognizing real-life objects in order for the software to create/assign sounds to the environment.

In regards to this requirement, the Hololens has the ability recognize and map the surfaces and textures of objects within the room. We plan to use this to create an object class that is defined by the properties of the objects that the user selects. Once the user selects the item it will then assign a unique/modified sound to this specific object. Each different unique object will have a 3D-rendered "wire" that is mapped from the object to the location of the keyboard itself. This can show the user the wide array of objects that are mapped to the keyboard, in which they can harmonize those items to create a melody or song unique to their location. We will implement a baseline color/texture-recognition functionality, and then use this to assign sounds. The Hololens itself has a speaker that will be sufficient in regards to playing sounds for our application. One constraint will be the speaker on the device, which is only played out of one portion of the headset, which will result in us not being able to create a surround sound based


SoundAround Use Case
Object Recognition
Creation of New Sounds
Map sounds to objects
Select from pre-made List
map wire from object to keyboard
Edit sound based on characteristic
Edit tone based on color
edit tone based on texture
Play sound through keyboard
Creating harmonious loop based on sound
User

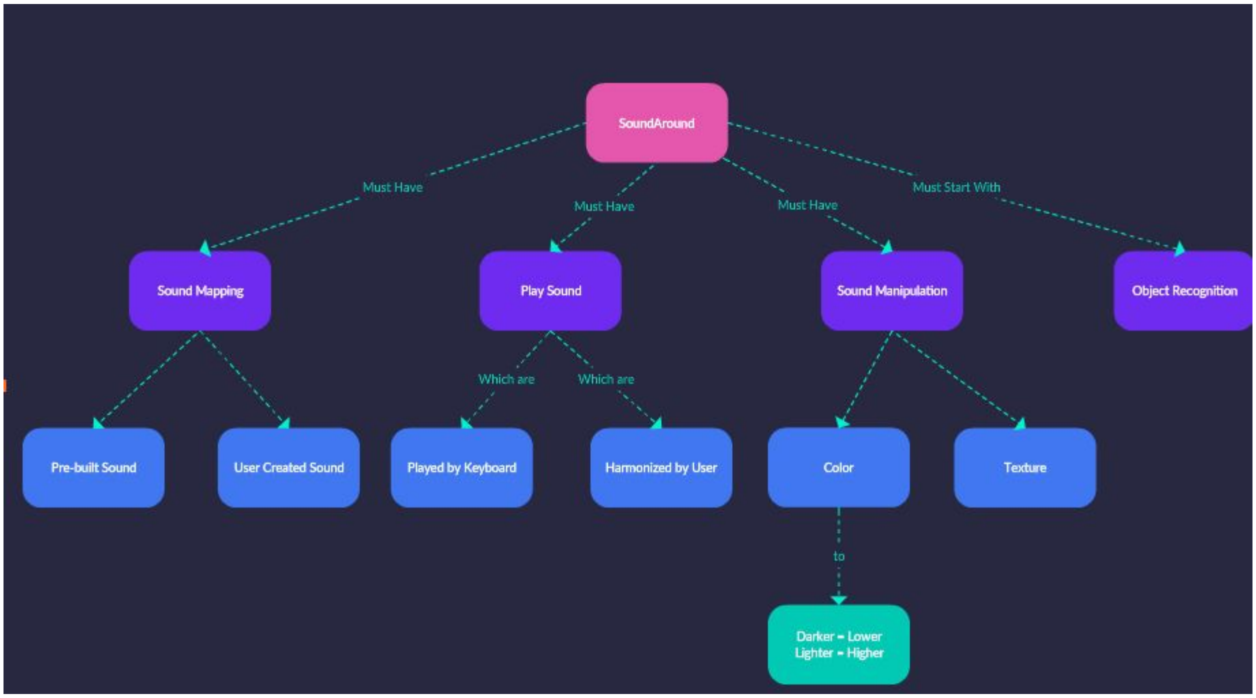
setting. This will limit the amount of immersive sounds in which the user can experience, as well as the quality of the sounds.

Another major constraint of the project is the limited field of view within the Microsoft Hololens itself. When wearing the headset there is a noticeable difference in what can be seen in the real-world environment and what the Hololens itself will display through the glasses. While the Hololens does render objects in the real-world space, which appear fully 3D to the user, the user still must view these objects through a small “window” in front of them. Our initial proposal came with the idea that the user can use a 3D-rendered keyboard the user would be able to play through gesture recognition. After our first experience with the Hololens, the group has decided to instead use a bluetooth keyboard for the implementation. Constantly having to move the user’s head to play the sound would ruin a major amount of immersion that Augmented reality would present for our project. With the keyboard the user just has to press a certain button and the sound tied to that object will be played. Another constraint that also led to this decision was the sensitivity of the gesture recognition. While the Hololens has a robust (albeit small) ability to display 3D “holograms”, its gesture recognition is known to have problems. After sufficient testing, we have decided the device would not be able to recognize the fine details of the users’ movement, which would be necessary for playing a digitally-rendered keyboard.



Another initial design constraint comes with how we ultimately decide to visually display the object mapping to the user. Among other things, we are considering a solid floating wire that runs from our keyboard to the object itself, a floating “aura” around selected objects, or even a pulsing wave that originates from objects as they are played. An issue with this is not obscuring the users vision too much, and not creating too much visual clutter.

In addition, there are a significant amount of technical constraints involved in designing and implementing our project. To develop for the Hololens, we will be using the Unity game engine. Unity will be the easiest in terms of creating an application in which we can download to our Hololens. We will be implementing using C# as our primary language for this application. Unity is a new platform, and C# is a new language, for many of us. In addition, developing in an AR space is brand new to the team. There are several APIs that we are looking at that will allow for easier implementation of features for application, an example being a tool for better object recognition. Using pre-existing APIs will allow for us to spend more time implementing things for sound creation and mapping and not spending time in creating the mapping for objects. Some of these API's include Vuforia Engine and OpenCV for object recognition, and Pure Data and Blunt for music synthesis. All of these APIs are open source, so there are no prior costs for us to use them in our implementation of our application.



Ethical and Intellectual Property Issues

Ethical:

People may use SoundAround in places that they ought not to. People may trespass on private property in order to find some new and unique sounds. This is an issue because it violates the property rights of those who own or reside on the land our users may trespass on. It also violates the privacy of said landowners or proprietors. There is also a concern that people may bring their headsets to sacred places such as churches or graveyards. This is disrespectful towards people who hold reverence for those places. Another place that would generate interesting and exciting sounds is a museum. It may ruin the experience of other museum goers if there are people using SoundAround. It could take away from people being able to admire the artwork or historic artifacts. If our users are standing in front of the people trying to admire the museum's items, they may not be able to see and their experiences would therefore be diminished.

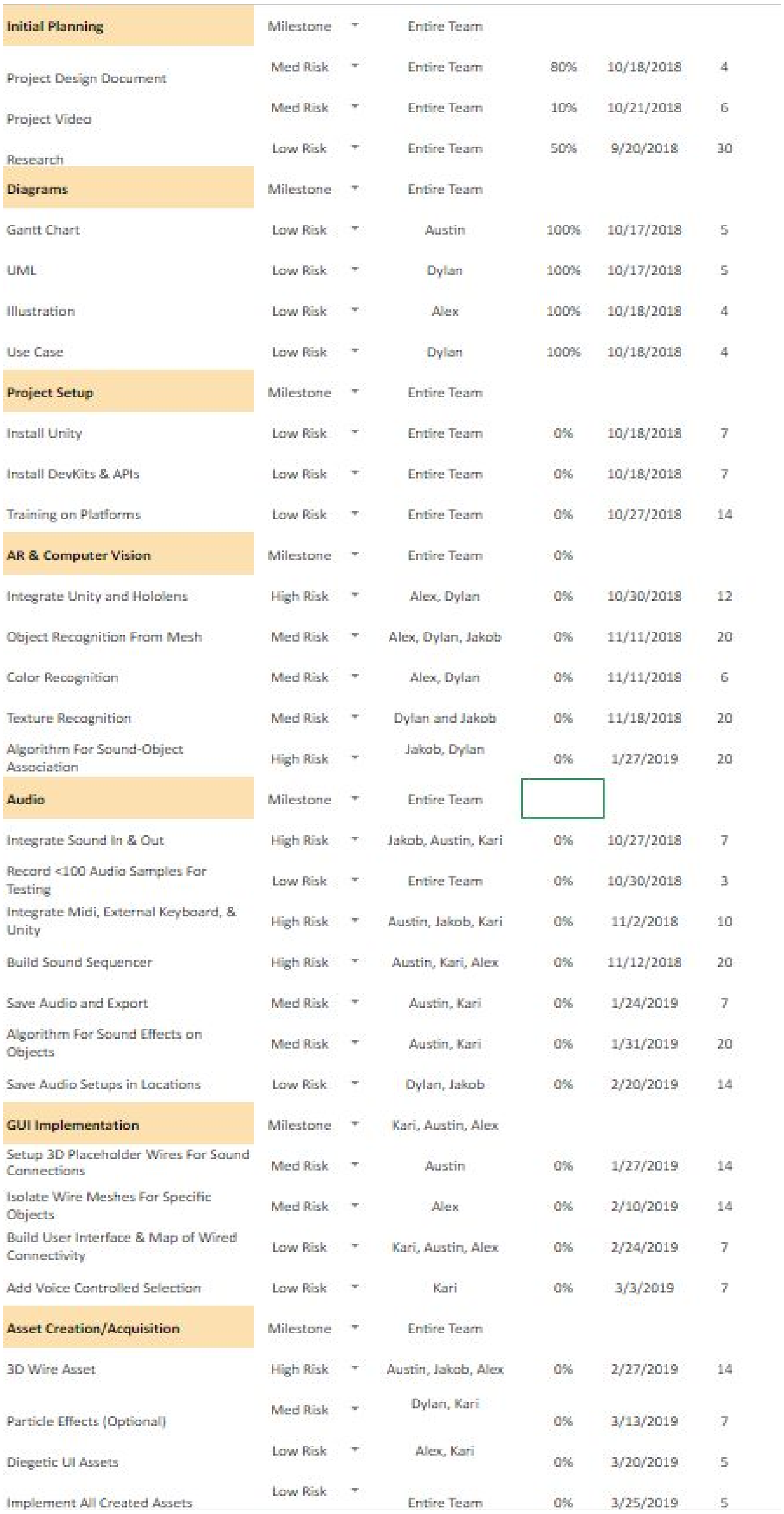
Intellectual:

We need to be careful about using other people's creations. We will be using Unity to help us generate graphics. Our group will most likely not be doing our own graphics from scratch so we need to make sure we use only open source graphics or get the appropriate permissions before using content created by other programmers. In addition to graphics, we also need to be careful about where we get our sound samples from. One way we may deal with this issue is by creating our own sounds. We were planning on getting together to record our own sound samples. Users would then be able to modify those sounds using filters based on objects in their environments. There is a possibility that users may use SoundAround to recreate their favorite songs, but that is outside the realm of our control and they would then be the ones to violate copyright, not us. If we are unable to make our own sounds, we will be sure to use open source audio, or otherwise license the use of the needed sounds.

GitHub:

https://github.com/austinirvine/soundaround

SoundAround Gantt Chart



Initial Planning	Milestone	▾	Entire Team			
Project Design Document	Med Risk	▾	Entire Team	80%	10/18/2018	4
Project Video	Med Risk	▾	Entire Team	10%	10/21/2018	6
Research	Low Risk	▾	Entire Team	50%	9/20/2018	30
Diagrams	Milestone	▾	Entire Team			
Gantt Chart	Low Risk	▾	Austin	100%	10/17/2018	5
UML	Low Risk	▾	Dylan	100%	10/17/2018	5
Illustration	Low Risk	▾	Alex	100%	10/18/2018	4
Use Case	Low Risk	▾	Dylan	100%	10/18/2018	4
Project Setup	Milestone	▾	Entire Team			
Install Unity	Low Risk	▾	Entire Team	0%	10/18/2018	7
Install DevKits & APIs	Low Risk	▾	Entire Team	0%	10/18/2018	7
Training on Platforms	Low Risk	▾	Entire Team	0%	10/27/2018	14
AR & Computer Vision	Milestone	▾	Entire Team	0%		
Integrate Unity and Hololens	High Risk	▾	Alex, Dylan	0%	10/30/2018	12
Object Recognition From Mesh	Med Risk	▾	Alex, Dylan, Jakob	0%	11/11/2018	20
Color Recognition	Med Risk	▾	Alex, Dylan	0%	11/11/2018	6
Texture Recognition	Med Risk	▾	Dylan and Jakob	0%	11/18/2018	20
Algorithm For Sound-Object Association	High Risk	▾	Jakob, Dylan	0%	1/27/2019	20
Audio	Milestone	▾	Entire Team			
Integrate Sound In & Out	High Risk	▾	Jakob, Austin, Kari	0%	10/27/2018	7
Record <100 Audio Samples For Testing	Low Risk	▾	Entire Team	0%	10/30/2018	3
Integrate Midi, External Keyboard, & Unity	High Risk	▾	Austin, Jakob, Kari	0%	11/2/2018	10
Build Sound Sequencer	High Risk	▾	Austin, Kari, Alex	0%	11/12/2018	20
Save Audio and Export	Med Risk	▾	Austin, Kari	0%	1/24/2019	7
Algorithm For Sound Effects on Objects	Med Risk	▾	Austin, Kari	0%	1/31/2019	20
Save Audio Setups in Locations	Low Risk	▾	Dylan, Jakob	0%	2/20/2019	14
GUI Implementation	Milestone	▾	Kari, Austin, Alex			
Setup 3D Placeholder Wires For Sound Connections	Med Risk	▾	Austin	0%	1/27/2019	14
Isolate Wire Meshes For Specific Objects	Med Risk	▾	Alex	0%	2/10/2019	14
Build User Interface & Map of Wired Connectivity	Low Risk	▾	Kari, Austin, Alex	0%	2/24/2019	7
Add Voice Controlled Selection	Low Risk	▾	Kari	0%	3/3/2019	7
Asset Creation/Acquisition	Milestone	▾	Entire Team			
3D Wire Asset	High Risk	▾	Austin, Jakob, Alex	0%	2/27/2019	14
Particle Effects (Optional)	Med Risk	▾	Dylan, Kari	0%	3/13/2019	7
Diegetic UI Assets	Low Risk	▾	Alex, Kari	0%	3/20/2019	5
Implement All Created Assets	Low Risk	▾	Entire Team	0%	3/25/2019	5

Detailed Gantt Chart:

https://docs.google.com/spreadsheets/d/14ESgS9XI-9af41ORbzQtXL6KPoRWr8b4NazVAYNlrpE/edit?usp=sharing

Change Log

Section	Changed (Yes/No)
Project Description	No
Project Milestones/Gantt Chart	No
Project Budget	No
GitHub / Resources	No