Joshua Nowlen Webb
CS 461
Fall 2020

Milestone 1

Introduction:

This paper covers three capstone projects that are of interest to me. The coverage of these projects includes a summary of what the project is about, a segment detailing the engineering requirements of the project, a place to discuss the constraints of this particular project, and lastly a rough timeline of the development of each of these projects. The projects to be discussed are 1) Smartphone-Based Manual and Repair App, 2) Where's my stuff? A voice interactive inventory system, 3) 3D Rhythm game for iOS of Android.

Smartphone-Based Manual and Repair App #1:

- Currently, repair guides require the reader to constantly direct their attention from the guide back to what they're currently doing. These breaks in attention can distract the user and sometimes take them away from what they're supposed to be doing or repairing. Since modern smartphones have cameras, displays, and processors that are sophisticated enough to process images to a large extent, we can make use of these components to identify pieces of images and overlay them with instructions for manuals. For example, if we were trying to put together a table then we will be able to glide over the pieces and they would highlight in the order that we intend to put them together and add images that aid the user in putting together the table. This tool would be a great one to help users in putting things together. For this project, the partner is Scott Fairbanks and there are no other listed stakeholders.
- The requirements for this project are that the system works on both iOS, Android, and any other mobile operating systems. It is also necessary that it operates in a way that processes the images quickly and allows for a swift and seamless user experience. It also must be able to image track and follow the particular components that need to be part of the project. In this case, we must ensure that the processing is something that is not only quick, but it also must be lightweight. We need to store the 3d models of the object in a way that keeps the size of the particular application and instructions small enough to be practical for our user to be able to download and use quickly and efficiently.
- Some constraints that affect the project is that the computing capabilities of the mobile devices. This could greatly limit the platforms that we are able to deploy our project. We may be limited to only the newest smartphones and tablets. This is unfortunate because we may be limiting the pool of people that we can use our application to only those that can afford the newest and latest technology. We are also limited in the storage capacity that we can have for our instruction sets. We might not be able to use an entire instruction set if it happens to be too long and the images and instructions re too complicated. In this case, it would necessitate that we break the instruction sets into smaller compact parts that allow us to use

the full capabilities of our application. This could be mediated if we choose to use lower resolution images and use lower-fidelity models for the 3d overlays. This could potentially solve the problem that we have of not being able to fit the entire instruction set onto the devices.

- I recommend that first, we decide which devices we would like to deploy this software on. In that case, it could always be scaled up into something larger or more complicated but it may be difficult to go the other way around and scale a large application down. In that case, accuracy and clarity may suffer if we haven't tested the lower fidelity models out. We would need mobile development software and tools, image-recognition software, and a few decent sets of instructions to model our project on. A rough timeline would look like 2 months used to create the models, instructions, and animations. The first 2 months could be used to wireframe a decent interface, the next 2 months could be used for image tracking and mapping the animations to the AR environment. The final months of the project ideally could be used to polish and scale it to different platforms. The personnel required for this project would be the team members, experts on AR that could be contacted for advice, and of course the project partner to provide guidance and feedback on team progress.

Where's my stuff? A voice interactive inventory system. #2:

- It's well known that even the most organized person can sometimes misplace things. Therefore, a great idea would be to have a fully automated and voice-activated system created for keeping track of where things are located. This system could be a great way for a person to free up their thought process to do other things allowing them one less thing to worry about. The voice interpretation could be done via a cloud-based system whereas the interpretation of the words and the synthesis of instruction could be done locally on the device. This system would also be able to interface with a database and store locally where all of the user's objects would be located
- The requirements for this project would first be that the team has a decent understanding of voice processing and the API that the particular voice processing service uses so that they could make the best use of the queries. Next, they would need a polished system that could make sense of the written requests that the person uses. A lot of thought would need to go into the interface and the types of commands that the system would accept from the users. Next, they would need a way to manage the database of information that they've collected so that the user could make the most of it. They would need to do all of this while having in-depth knowledge of the system that they're developing and it's capabilities.
- Some constraints that could affect this project would be the capability of the voice processing service that's being used. There may be monetary costs to it or they may limit the number of queries that can be submitted to the service. The storage capabilities of the application would need to be considered so that there is sufficient space to store the database while also keeping it small and compact enough to be worthwhile. Lastly, the interface would need to be efficient enough

that the user isn't overwhelmed with a stream of information that they can't use or understand. All of this needs to be taken into consideration for the project to be fully successful.

- Some key skills that would be needed are that someone on the team needs to have a good understanding of Human-Computer Interaction so that the interface can be designed in a way that is the most useful to the people that are going to be using the service. Also, the prompts need to be clear so that the user can know what information they need to be providing for it to be of best use to them. Also, there needs to be a good understanding of how voice recognition services work and how to parse written information to interpret it into meaningful commands. Also good knowledge of databases and not only how to store and retrieve information from them, but also how to structure them in a way for quick and efficient retrieval of information. Access to the machine and the voice recognition service would be needed in order for the project to be completed. As well as tools for diagramming the project and the things needed for creating a working user interface. A good timeline for this project would be for the first term to be spent designing what a good user interface for the project would be like and what prompts would the user be expected to respond to. The next term could be used for developing the interface and also getting the prompts and text processing to work, as well as developing the database that is going to be used to store the items and their locations. Lastly, the final part of the year could be dedicated to user testing and polishing the product until it's in a state that it is ready to be used.

3D Rhythm game for iOS and Android #3:

- This project involved creating a game in the rhythm tapping genre in Unity. The game involved the user tapping on a screen in the rhythm of either a musical track or to icons that slide across the screen (this is similar to how the Guitar Hero games play). The game should be modeled in 3D which presents a spin on how some entries in the genres appear. The particular visual style of the game is free to be artistically decided by the teammembers that are placed on the team. Unity is a powerful game engine that has been used to create passion projects, small indie studio titles, and also large AAA games that sell millions of copies. Unity has also been used in movie creation and on scientific projects.
- Some of the requirements involve knowing the capabilities of the system that you're developing for. The team must be aware of which devices they intend to develop the game for and how those are going to react to the Unity engine. Furthermore, it's going to be necessary to synchronize the music track or visuals to the actual tapping input that's going to be gathered from the device. Some other requirements are the the models that are used for the game are not taking away from the performance of the product and are not causing graphical glitches or framerate issues. Furthermore, the user interface that's going to be designed for the game must be presented in a way that is going to aid the users in learning how the game functions and the overall goals of the game. This shouldn't be too

difficult but it should be done in a way that is not the typically tutorial section so to not distract from a seamless user experience of gameplay.

- Some constrains that could affect the development of this game could be complication with implementing the Unity engine in the development. Although Unity is a great tool for creating a game, it could be difficult to first learn. Also, it's important to make sure that the 3D models that are being used for the game are able to be used within the game while ensuring that the users have smooth and seamless gameplay without any deviations. It's important that there are methods to ensure that the creation of the game is done in a way that encourages users to come back to the game such as high scores and streaks. If the team developing the game is not familiar with video game design language them it might be a complication to deviate too far from the typical ways that developers signal things to consumers. It perhaps could detract from the experience if the user needs to familiarize themselves with a new method of communication.
- Some key skills is that the team needs to be familiar with video games and the fundamentals of their design. This will allow the team to be able to communicate with their users in a way that can give them swift and smooth feedback to aid in their gaming experience. Knowledge of mobile application development is going to be highly important to the development of the game. Because it's being created for iOS and Android, it is going to be essential to know how these systems work and how to create great experiences on the platform. Knowledge of Unity is going to help since that's the engine that the game is going to be using. Experience with 3D modeling is going to help in the creation of the game since it is going to be the perspective and the style of the game. Furthermore, knowledge of graphic design and rhythm is going to be of great importance to developing to the game to know where to place the interactive beat that the player is going to be watching for. I believe that a good brief timeline for this project would be to allow the first segment of capstone to be for planning the type of game, doing research on ther games of the genre, and deciding a direction to take the user interface in. The second term could be used to gather songs, create the basic assets that are going to be in the game, and start to lay out some of the demo levels. The last term could be used to add in some of the higher level features such as high scores, streaks and other visuals that could aid in the user experience with the game.