Project Context:

3D image segmentation is a significant process in scientific and medical operations. It is the process of separating a digital image into multiple segments. The goal is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. The formation of a reliable algorithm for automatic segmentation is eagerly sought for. This process is hindered by the ability of these softwares to produce quality segmentations due to ambiguous boundaries and noise. Manual segmentation is still the primary protocol for segmentation process which poses a problem because it can be tiresome, time consuming and prone to errors.

The volume viewer (VV) is a user friendly software that allows for experts and non-experts to segment complex structures. It uses three windows on the interface with one being the selected view of the data chosen. It uses a manual contour drawing tool to form segments which includes deleting, editing and intersection points for previous contour recognition. The second including the cubecam widget that allows you to manipulate 2D slices of the 3-Dimensional image. The tools on the interface include scrolling, rotating, zooming, etc. It also helps you identify your exact position in the data which served as a problem for traditional practices. The third window displays the pieces of the data that you have already segmented in a gray mesh form. The volume viewer is a great choice for segmenting and contains many features that will assure consistency, efficiency, quality and accuracy. The project is to make the volume viewer compatible for tablet devices.

Terms you should know:

1. **Native Development Kit**: a toolset that allows you to implement parts of your app using native-code languages such as C and C++.

2. **Java native Interface**: is a programming framework that enables Java code running in a Java Virtual Machine (JVM) to call and be called by native applications and libraries written in other languages such as C, C++ and assembly.
### Background:

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Do Experts Look while doing 3d image segmentation</td>
<td>Anahita Sanandaji, Cindy Grimm, Ruth West, Jeremy Deutsh, Max Parola, and Meghan Kajihara</td>
<td>4 experienced participants in segmenting were selected to perform and replicate their normal segmenting processes. While they were performing they were equipped with an eye tracker that allowed the researchers to record their mental models and low-level perceptions including how long the user’s eye gazed at the data, tools, boundaries, regions etc.</td>
</tr>
<tr>
<td>A Guided Approach To Segmentation of Volumetric Data</td>
<td>Michelle Holloway, Cindy Grimm, Ruth West, Ross Sowell</td>
<td>Participants used a guided approach to 3D segmentation. They were tested for consistency, quality, accuracy and efficiency. The researchers utilized the similarity of repeated segmentation task to address the key difficulties and make it possible for non-experts to segment relatively complex structures.</td>
</tr>
<tr>
<td>CubeCam: A screen-space camera manipulation tool</td>
<td>Nisha Sudarsanam, Cindy Grimm, Karan Singh</td>
<td>The cubecam was displayed to explain how this image spaced camera manipulation widget uses a cube to present the camera’s position in regards to the scene and as a convenient tool to modify the camera.</td>
</tr>
<tr>
<td>HCC: Medium: Collaborative Research: Developing conceptual model for navigation, marking, and inspection in the context of 3D image segmentation</td>
<td>Cindy Grimm</td>
<td>Domain experts participated in a study to research the low-level and high-level aspects of the segmentation process. The info will be used to define the correct way to design and evaluate complex images; furthermore, helping to create a software for automatic segmentation with expert behavior.</td>
</tr>
</tbody>
</table>
Background Summary:

Many studies have been carried out to evaluate the 3D segmentation process to define ways to improve it. Participants were recorded on the low-level and high-level constraints of segmentation. Low-level aspects of segmentation include visual cues, delineation of structures by marks, and local accuracy or quality criteria. Higher-level constraints include the documentation of different segmentation behaviors throughout the process and the creation of algorithms that can perform these techniques. Participants were graded on consistency, efficiency, repeatability and quality.

The specific techniques of segmentation were also observed. Participants were tested on their ability to render reliable segmentations on simple to complex structures using non-parallel and parallel segmentation routines. Data was recorded based on their low-level aspects including eye gaze location, global navigation, marking action, etc. Graphs were made based on their time, frequencies and averages to conclude whether novice segmenters can create reliable segmentations that are tantamount to experienced ones.

A complete description of the many features of the cubcam were reviewed, tested and explained. The zoom, scrolling, rotating and volume rendering tools were described comprehensively. The cube cam is the tool used inside the volume viewer program and is also essential to the software’s success. Participants were researched on the use of the volume viewer. Simple and complex structures were used along with experienced and inexperienced users. A guided approach was done utilizing automatic navigation paths to non-parallel cross sections of the 3D volume, images of example marks from similar data sets, and text instructions. The feasibility of this system of this system with expert users and the usability of this system with non-expert users was tested.

Specific Goals:

The objective is to run VV as an application on the tablet device. The protocols to manipulate the 3D space will be done using touch gestures in lieu of mouse clicks. By using a tablet that’s GPU is capable of 3D rendering we can transfer the volume viewer source code to the tablet. We will use a 3D rendering application source code to ensure the display of 3D data objects. We will use the same three window interface and default tablet gestures to enable the camera manipulation and contour drawing features.

We must first find a tablet that is capable of rendering 3D volumes while obtaining an open source application that already does. Then we will be able to insert a portion of the VV
source code inside the chosen source code which will allow us to display the interface. Assuming we are successful, we can then start creating the gestures to replace the previous tools and cube cam behaviors. The gestures must be written in the programming language the volume viewer supports and then implemented inside the code. Lastly, the code will be run through a software that will translate the new VV source code to the tablet to run the full version of the application.

Team Deliverables:

- Collection of relevant papers
- Knowledge of the necessary programming
- Discovery of 3D rendering tablet
- Choice of open source tablet application with 3D rendering
- Utilize Native Development Kit (NDK)
- Creation of gestures through Java
- Construct gesture codes through Java Native Interface (JNI)
- Enable a portion of VV on code on tablet device
- Run full VV source code on tablet device

Specific Approach:

We must first familiarize ourselves with the VV software and source code. We will learn the C++ programming language (if unknown) preparing yourself to be able to add code or edit it if needed. This can be done by obtaining and reviewing the VV source code and performing the tutorial. Also, using the various online programs such as codeacademy.com or khanacademy.com that teaches you the C++ programming language. The tablet and source code will be found by surfing the internet and probing the various github repositories. An existing 3D rendering application for tablet device’s source code will be used instead of having to create one from scratch. We will also use source codes from colleagues that are available to us. The JNI and NDK will be used to translate the gestures and edited VV source code to the selected tablet. Only a small fraction of the VV source code will be used the first time. If successful, we will continue to adding the remaining code with the touch screen tool capabilities.

Techniques:

- Translation of gestures through JNI
- Translation of source code through NDK
- Tests for transfer of VV on the tablet
- Preferred Tablet: Samsung Galaxy Tab 2
Unknown or trouble points:
- NDK may not translate program properly
- NDK is only applicable for androids (not iOS)
- 3D source code may not display volume viewer along with 3D images
- VV code may be too complex
- Translation of source code may have to be done manually

Timeline:

Week 1: Background information and preparation
Week 2: Looking for compatible devices and source codes (tablet, 3D rendering applications)
Week 3 & 4: Execution of a portion of the VV source code on the tablet, troubleshooting problems, editing code, and change of device if necessary
Week 5: Creation of gesture codes/addition to VV source code/implementation on tablet device
Week 6, 7, 8, and 9: Any changes or edits to the new code, device or plan to ensure complete success/full application of VV
Week 10: Finalization of project

CheckPoints:
- Portion of Volume Viewer source code runs on tablet device successfully
- Gestures are successfully made and added to VV source code
- Gestures run perfectly on tablet device
- Full VV software runs on tablet device perfectly

Backup Plan:
- Alternative tablet device (iOS, Nexus)
- Different application source code
- Different procedure, example: NDK plan is obsolete and 3D rendering code must be written from scratch
- Code including VV, gestures, 3D rendering must be written in an alternative programming language