

EXPLORING THE SINONASAL CAVITY IN THREE DIMENSIONS:  
TEACHING OTOLARYNGOLOGY SURGICAL TRAINEES CLINICAL  
ANATOMY USING A WEB-BASED LEARNING RESOURCE

by  
Shirley Li

A thesis submitted to Johns Hopkins University  
in conformity with the requirements for the degree of  
Master of Arts

Baltimore, Maryland  
March, 2022

## **Abstract**

Endoscopic sinus surgery is the most common procedure to restore normal sinus function in chronic sinus disease and is performed in a small complex anatomical space intimate to critical structures (i.e. eyes, anterior cranial fossa, and major head and neck vessels). Otolaryngology surgical trainees must therefore have a thorough three-dimensional (3D) understanding of the sinonasal space to avoid disorientation when surgically navigating through a limited endoscopic view.

However, learning of this anatomical region is hampered by the complexity of anatomy as well as limitations in current teaching resources. Commonly used two-dimensional (2D) visualizations include static illustrations and radiological CT imaging that poorly convey the 3D nature of the space. The author could not find any existing educational resources employing 3D visualization of the sinonasal space.

Web-based multimedia resources, such as online sinus surgery videos are widely used as a learning tool with novel clinical training modules developed to facilitate correlation of anatomical knowledge in radiological visualization and endoscopic surgical view. However, these resources lack correlation among the different types of media and the spatial relationships of clinically relevant structures in 3D space is not fully correlated to static 2D visualizations.

In this project, we propose creating a web-based interactive resource offering a comprehensive and multidimensional visualization of the sinonasal cavity. This resource will consist of two learning modes: i) In Explore Mode, fully interactable 3D schematic and CT-segmented models are presented alongside 2D axial and coronal CT image series, allowing users to navigate the sinonasal cavity in 3D space and bridge the gap between 3D and 2D

visualizations. ii) In Clinical Mode, surgical video clips are featured in addition to a schematic 3D model and CT image series, improving spatial orientation during surgery by correlating 3D and 2D visualizations of the sinonasal cavity from a clinical perspective.

The authors of this research postulate such a resource can improve clinical training outcomes among otolaryngology surgical trainees.

**Shirley Li**

### **Chairpersons of the Supervisory Committee**

**Nicholas Rowan, M.D.**, Preceptor

Assistant Professor of Otolaryngology - Head and Neck Surgery

The Johns Hopkins University School of Medicine

**Anirudh Saraswathula, M.D., M.S.**, Content Advisor

Resident Physician in Otolaryngology – Head and Neck Surgery

The Johns Hopkins University School of Medicine

**Juan R. García, M.A., C.C.A.**, Department Advisor

Associate Professor of Art as Applied to Medicine

Johns Hopkins University School of Medicine

## Acknowledgements

This thesis project would not have been possible without the guidance and support of many individuals. I would like to extend my sincere appreciation to:

**Juan Garcia**, my advisor, and Associate Professor in the Department of Art as Applied to Medicine for guiding me through the planning and development of this project. Thank you for your continuous support, patience, and valuable feedback throughout this project. You have encouraged me to come up with creative solutions and inspired me with your expertise. It was my pleasure to work with you.

**Dr. Nicholas Rowan** and **Dr. Ani Saraswathula**, my preceptor and content advisor, for proposing this exciting thesis topic and providing resources for this project. I would like to thank both of you for your enthusiasm throughout this research and sharing your extensive knowledge with me, providing feedback, and dedicating your time to many elucidating discussions. I am grateful for the opportunity to work with you on this project.

**Patrick McCarthy**, who is a game development instructor, for your instruction and advice on developing the interactive aspects of my project using Unity. Thank you for contributing your time and expertise with patience that made this project possible.

**The Faculty and Staff of the Department of Art as Applied to Medicine**, for introducing me to the wonderful world of medical illustration and biological communication. I am enormously proud to be a part of this enlightening community. Thank you for your wisdom, constant inspiration, and positive energies that helped me to flourish as a medical artist.

My amazing cohorts and friends, **Jennifer Wang, Kelly Lim, Gaby Rivera- Del Río, Tina Wang, Jason Brady,** and **Emily Simpson**, for your emotional support and inspiration over the span of the program. I enjoyed walking this path with all of you talented individuals.

**Adrian Matthews**, who is a technical support analyst at Johns Hopkins Tech Hub, for helping me build my first customized PC. I cannot say enough thank you for the technical advice and numerous troubleshooting you have done for me.

My parents: **Annie Gao** and **William Li**, for your love and support throughout my life. Thank you for giving me courage and opportunities to chase my dream.

Finally, my beloved sister, **Killy Li**, and my feline friend, **Chickadee**, for your companion and comfort with love on countless late nights.

## Table of Contents

<b>Abstract .....</b>	<b>ii</b>
<b>Acknowledgements .....</b>	<b>iv</b>
<b>Table of Contents.....</b>	<b>vi</b>
<b>List of Tables.....</b>	<b>viii</b>
<b>List of Figures.....</b>	<b>ix</b>
<b>Introduction.....</b>	<b>1</b>
An Overview of the Sinonasal Cavity .....	1
Existing Teaching Resources and Limitations .....	5
Computed Tomography Imaging.....	8
Web-based Interactive 3D Models as a Learning Resource.....	8
Research Objectives .....	9
Educational Goals and Target Audience .....	10
<b>Materials and Methods.....</b>	<b>11</b>
Software Overview .....	12
Acquisition of Radiological Imaging Data .....	13
Stage 1: Development of 2D and 3D Assets .....	15
Development of 3D Schematic Model.....	16
Development of 3D Reconstructed Model .....	21
Preparation of CT Image Series .....	31
Stage 2: Development of Interactive Content.....	33
User Interface and Interactive Design.....	33
Setting up User Interface .....	34
Creating Rounded Border for Each Panel.....	38
Creating Full Screen Button .....	40
Importing 3D Models into Project Scene.....	43
Synchronized Rotation and Panning of 3D Models.....	44
Zooming of 3D Models .....	47
Creating Scrollable Radiological Image Series.....	48
Incorporating CT Slice Orientation Planes in 3D Reconstructed Model .....	50
Creating Toggles to Hide Structures .....	52

Highlighting a Structure by Selecting an Item from Navigation Panel.....	53
Switching between Explore and Clinical Modes and Incorporating Video Components .....	54
<b>Results.....</b>	<b>56</b>
Stage 1: 3D Models .....	56
Stage 2: Interactive Elements .....	60
Access to Assets Resulting from this Thesis .....	65
<b>Discussion .....</b>	<b>66</b>
Project Goal .....	66
Considerations of Project Design.....	66
Limitations .....	68
Project Innovations .....	69
Future Directions.....	70
<b>Conclusion .....</b>	<b>71</b>
<b>Appendices .....</b>	<b>72</b>
Appendix A: “FullScreen” Script .....	72
Appendix B: “StructureControl2” Script for Concerted Manipulation of Models .....	73
Appendix C: “CameraZoom” Script for Zooming Action .....	75
Appendix D: “CoronalCT” Scrollable Image Series Script.....	77
Appendix E: “Toggle” Script for Visibility Control of Structures .....	78
Appendix F: “Highlighting” Script.....	79
Appendix G: “ModeControl” Mode Switching Script.....	80
<b>References.....</b>	<b>82</b>
<b>Vita.....</b>	<b>84</b>

## List of Tables

<b>Table 1.</b> Structures created for 3D schematic model using Cinema 4D .....	<b>16</b>
---	-----------

## List of Figures

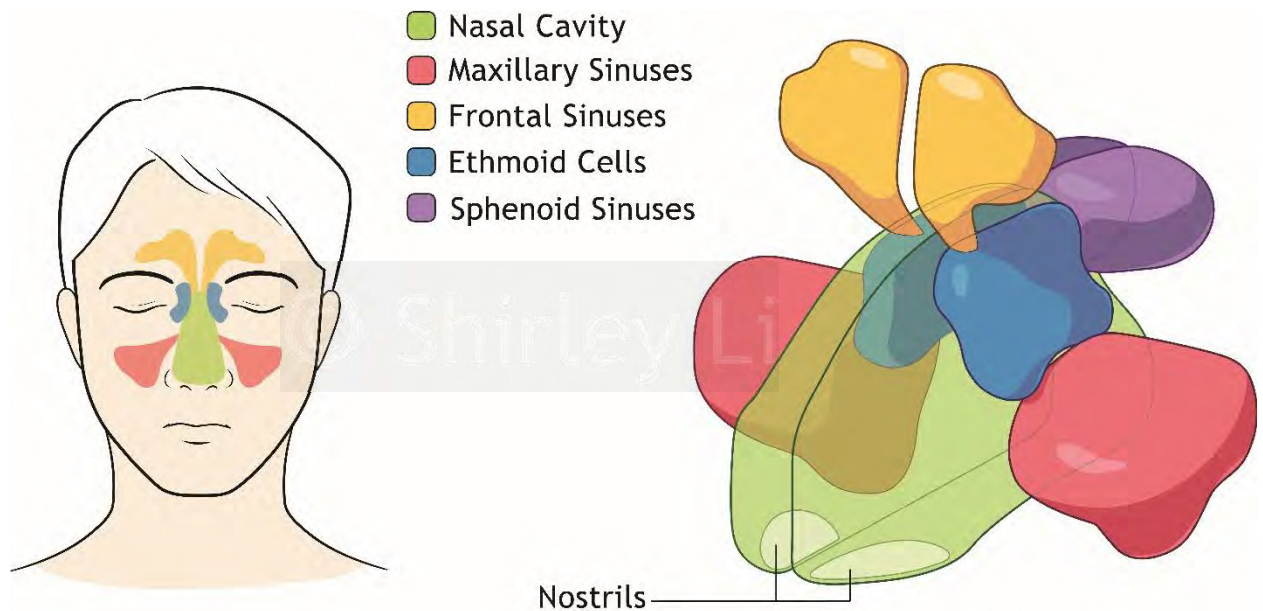
<b>Figure 1.</b> Basic anatomy of the sinonasal cavity .....	<b>1</b>
<b>Figure 2.</b> Functional regions within the nasal cavity .....	<b>3</b>
<b>Figure 3.</b> The five embryological ridges along the lateral nasal wall .....	<b>5</b>
<b>Figure 4.</b> Clinical training module created by Bailey et al. (2021).....	<b>7</b>
<b>Figure 5.</b> Planned layout of interactive resource .....	<b>12</b>
<b>Figure 6.</b> Segmentation of CT head data using D2P .....	<b>14</b>
<b>Figure 7.</b> Schematic construction of the ethmoid sinus by Wise S.K. et al. (2012) .....	<b>15</b>
<b>Figure 8.</b> Creation of uncinate process and lateral nasal wall using Cinema 4D .....	<b>17</b>
<b>Figure 9.</b> Creation of middle turbinate using Cinema 4D .....	<b>18</b>
<b>Figure 10.</b> Creation of internal carotid artery and vidian nerve using Cinema 4D.....	<b>19</b>
<b>Figure 11.</b> Creation of lateral wall of sphenoid sinus.....	<b>20</b>
<b>Figure 12.</b> Cropping of skull and mucosa mesh in ZBrush .....	<b>21</b>
<b>Figure 13.</b> Reconstruction of internal carotid artery using ZBrush.....	<b>23</b>
<b>Figure 14.</b> Reduction of polygon count of sinonasal structures using ZRemesher.....	<b>25</b>
<b>Figure 15.</b> Reduction of polygon count of cranium using Decimation Master .....	<b>26</b>
<b>Figure 16.</b> Dividing mucosa mesh into different polygroups using ZBrush .....	<b>28</b>
<b>Figure 17.</b> Reconstruction of reference cranium using ZBrush.....	<b>30</b>
<b>Figure 18.</b> Batch-processing of CT images.....	<b>32</b>
<b>Figure 19.</b> Adjusting aspect ratio using Unity .....	<b>35</b>
<b>Figure 20.</b> Setting up 3D Schematic panel through camera viewport .....	<b>37</b>
<b>Figure 21.</b> Adjusting the Viewport Rect .....	<b>38</b>
<b>Figure 22.</b> Importing and processing of sprite for rounded border .....	<b>39</b>

<b>Figure 23.</b> Applying rounded border to panels .....	<b>40</b>
<b>Figure 24.</b> Establishing full screen function.....	<b>41</b>
<b>Figure 25.</b> Duplicated first camera viewport for completed basic UI set up .....	<b>42</b>
<b>Figure 26.</b> Before and after incorporating multiple light sources for 3D models .....	<b>44</b>
<b>Figure 27.</b> Script for synchronized rotation and panning of models.....	<b>46</b>
<b>Figure 28.</b> Script for synchronized zooming of models.....	<b>47</b>
<b>Figure 29.</b> Setting up scrollable CT image series.....	<b>49</b>
<b>Figure 30.</b> Script for moving CT planes in 3D Reconstructed view .....	<b>51</b>
<b>Figure 31.</b> Attaching “Toggle” script and assigning function .....	<b>52</b>
<b>Figure 32.</b> Attaching “Highlighting” script and assigning function .....	<b>53</b>
<b>Figure 33.</b> Screenshot of “ModeControl” script and Inspector window.....	<b>55</b>
<b>Figure 34.</b> Rendered schematic model of the sinonasal cavity .....	<b>56</b>
<b>Figure 35.</b> Rendered CT-reconstructed model of sinonasal cavity.....	<b>57</b>
<b>Figure 36.</b> Rendered CT-reconstructed model of partial cranium .....	<b>58</b>
<b>Figure 37.</b> Combined cranium and sinonasal cavity models .....	<b>59</b>
<b>Figure 38.</b> Rendered orienting cranium model.....	<b>60</b>
<b>Figure 39.</b> Screenshot of Explore Mode within Unity .....	<b>61</b>
<b>Figure 40.</b> Screenshot of Clinical Mode within Unity .....	<b>61</b>
<b>Figure 41.</b> Screenshot showing interactions with 3D models.....	<b>62</b>
<b>Figure 42.</b> Screenshot showing activation of CT slice orientation planes .....	<b>63</b>
<b>Figure 43.</b> Screenshot showing full screen function .....	<b>64</b>

## Introduction

### *An Overview of the Sinonasal Cavity*

The sinonasal cavity is part of the uppermost respiratory tract. It consists of four pairs of paranasal sinuses surrounding the nasal cavities: the frontal, ethmoid, maxillary, and sphenoid sinuses, making up an interconnected network of air flow and drainage passages (**Figure 1**). The nasal cavities and paired paranasal sinuses directly communicate with each other through ostia. They function as a unit to remove foreign particles and warm and humidify inhaled air before it reaches the nasopharynx.

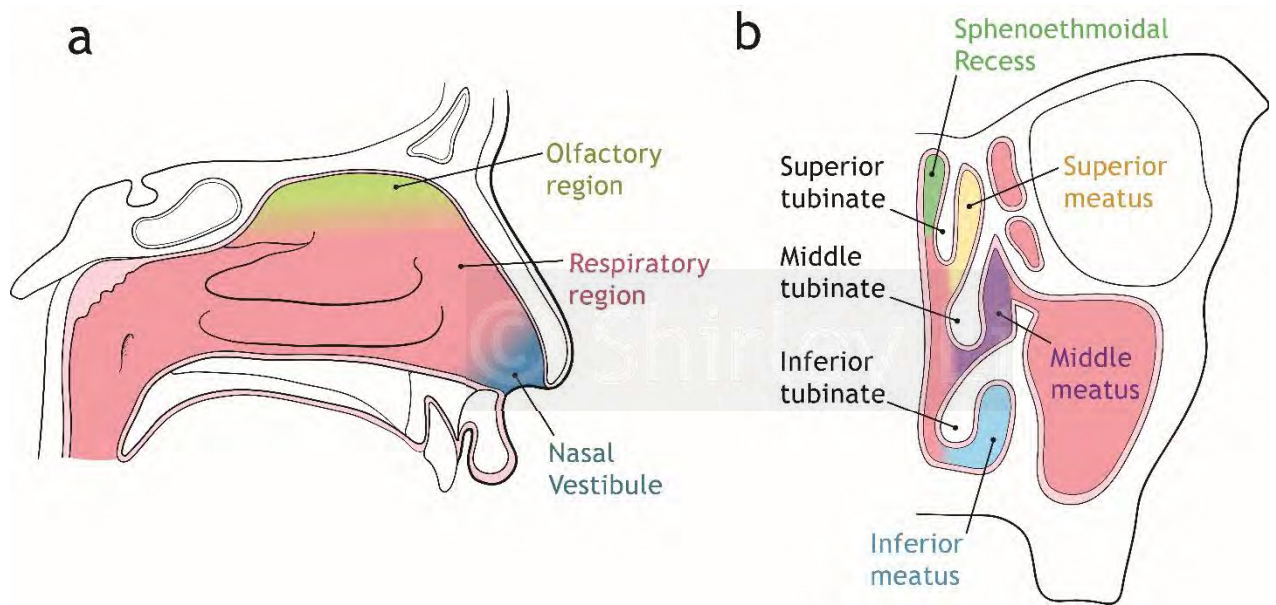


**Figure 1.** Basic anatomy of the sinonasal cavity. Image adapted from *Atlas of the Nasal Cavity and Paranasal Sinuses Anatomy*, Michear A. & Hoa D. (2021)

The nasal cavity is divided in two by the nasal septum and has three regions: the nasal vestibule, olfactory region, and respiratory region (**Figure 2a**). The nasal vestibule is the entrance of the nasal cavity and is lined with stratified squamous epithelium and

contains hair follicles. The olfactory region consists of the olfactory cleft, which resides medial to the middle turbinate on the roof of nasal cavity. This region is covered with olfactory epithelium that contains olfactory receptors. The nerve fibers project superiorly and cross the cribriform plate of the ethmoid bone towards the olfactory bulb, through which nerve signals are transmitted to the brain. The respiratory region of the sinonasal cavity is the nasal airway. It is covered with respiratory epithelium and mucous cells, which functions to warm, humidify, and filter inhaled air and regulates nasal airflow under sympathetic control (Sobiesk J. L. & Munakomi S. 2022).

Within nasal cavities, three sets of turbinates can normally be found along the lateral nasal walls, which include the superior, middle, and inferior turbinates. Occasionally, a fourth turbinate—the supreme turbinate can be observed as a normal anatomical variant. Turbinates are medially protruding scroll-shaped bones covered with mucosa that augment surface area of the nasal cavity and create air turbulence. The horizontal space beneath the attachment of each turbinate to the lateral nasal wall is referred to as a meatus, where secretion from paranasal sinuses or orbits drains into (**Figure 2b**).



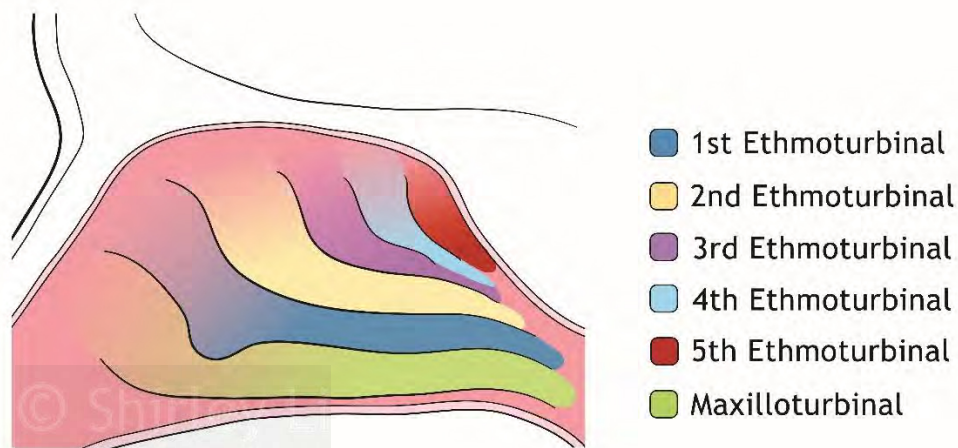
**Figure 2.** (a) Functional regions within the nasal cavity and (b) a coronal section showing nasal turbinates and meatuses; only half of the nasal cavity is shown.

Paranasal sinuses are paired air-filled cavities that are named for the skull bones that they pneumatize. They reduce the weight of the skull and act as resonating chambers. All four pairs of sinuses are lined with respiratory epithelium and covered with mucus that traps airborne particles. Secretions from the sinuses drain into the nasal cavities through a series of well-defined pathways.

The maxillary, anterior ethmoid, and oftentimes the frontal sinuses drain into nasal cavities through the osteomeatal complex (OMC). The OMC is composed of the maxillary ostium, infundibulum, ethmoid bulla, uncinate process, hiatus semilunaris, and middle meatus. The ethmoid bulla is the largest ethmoid cell and the uncinate process is a hook-like process that parallels the shape of the anterior surface of ethmoid bulla and extends inferiorly. The infundibulum is the curved channel lateral to the uncinate process whereas the hiatus semilunaris is the two-dimensional (2D) opening medial to the infundibulum between the free edge of the uncinate process and ethmoid bulla. Although the OMC is one

of the most important clinically relevant anatomical landmarks, the spatial relationships between structures are poorly appreciated using 2D media. The superior meatus is the region for drainage from posterior ethmoid sinuses, while the sphenoethmoidal recess, which represents the region superior and posterior to the superior turbinate, receives the drainage from sphenoid ostia (the opening of the sphenoid sinuses lateral to the nasal septum). Finally, secretions from the lacrimal sac drain through the nasolacrimal duct and into the inferior meatus.

During embryological development of the sinonasal cavity, five ridges of cartilaginous and soft tissues start to form along the lateral nasal wall at around 8 weeks of gestation (**Figure 3**). These ridges are referred to as the ethmoturbinals, which eventually develop into turbinates and related structures on the lateral nasal wall. The first ethmoturbinal ridge gives rise to the agger nasi cells, a pneumatized area at the anterior portion of the lateral nasal wall, and uncinate process. The second ethmoturbinal gives rise to the ethmoid bulla. The middle turbinate develops from the second and third ethmoturbinals, whereas the superior turbinate comes from the third ethmoturbinals. When present, the supreme turbinate comes from the fourth and fifth ethmoturbinals. The ethmoturbinals are considered to originate from the ethmoid bone; therefore, both middle and superior turbinates derive from an ethmoid bone precursor. The inferior turbinate, however, develops from an independent ridge, the maxilloturbinal, with its origin involving the cartilaginous nasal capsule and the bone of the maxilla (Wise S. K. et al. 2012). In comparison, the development pattern of paranasal sinuses is more variable and unpredictable. Pneumatization of sinuses begins as early as the 10<sup>th</sup> week prenatally, however, the growth of sinuses continues into adolescence.



**Figure 3.** The five embryological ridges along the lateral nasal wall. During embryological development, these give rise to paranasal wall structures.

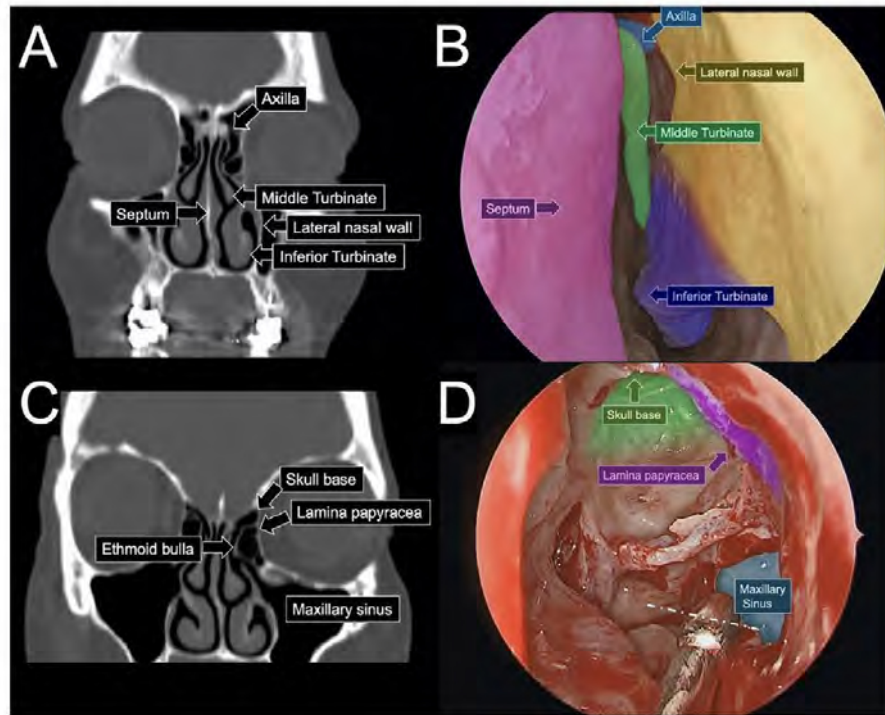
### *Existing Teaching Resources and Limitations*

The anatomy of the sinonasal cavity is highly complex and morphologically variable. This complex anatomical space is also intimate to critical structures such as eyes, the anterior cranial fossa, and major head and neck vessels. Therefore, it is important for surgeons to have a thorough anatomic understanding of this region and note any pre-existing anatomical variations by studying radiological imaging prior to performing safe sinus surgeries. In recent years, endoscopic/endonasal approaches had become the most common type of technique for sinus and skull base surgeries. Despite the improved operative efficiency and surgical outcomes with this minimally invasive approach, endoscopic surgery requires surgeons to navigate within narrow and complex nasal spaces from an endoscopic perspective, operate through a limited endoscopic surgical view, and minimize neural and vascular manipulation. Unfamiliarity with three-dimensional (3D) spatial relationships between structures and the failure to identify relevant anatomical landmarks can lead to intraoperative disorientation, which poses significant risk for patients.

Understanding spatial nuances of this anatomical region is hampered by limitations of teaching methods used. During medical school, sinonasal anatomy is primarily taught using traditional 2D textbook visualizations, which poorly communicates 3D relationships within this anatomical region. In addition, the sinonasal cavity is often an overlooked region during typical cadaveric dissection lab because its location deep in the skull renders it inaccessible compared to other anatomical regions of the body.

During residency, otolaryngology surgical trainees mostly learn through firsthand experiences and observations made in the operating room. Utilization of web-based multimedia resources, such as online sinus surgery videos to improve clinical knowledge and skills beyond the duration of formal surgical training, has become popular among otolaryngology surgical trainees (Koch G. K. et al. 2017). One example is [www.SinusVideos.com](http://www.SinusVideos.com), a website that hosts high-quality instructive sinus surgery videos for surgical trainees. Further, investigation shows that YouTube.com is also a popular educational video source for surgical preparation (Rapp A. K. et al. 2016). However, there are no didactic visualization and supplementary materials associated with these videos to provide greater anatomical context of the surgical field and facilitate learning the anatomy. In a recent study by Bailey et al. (2021), a self-directed multimedia curriculum on sinonasal surgical anatomy was created for clinical training. This curriculum features narrative videos with clinically relevant structures labelled on static radiographic imaging and endoscopic surgical video stills (**Figure 4**). However, despite a somewhat successful attempt at applying anatomical knowledge to clinically relevant considerations, a knowledge gap exists between correlating the 2D radiological images to the complex 3D space seen during surgery. Furthermore, this multimedia curriculum lacks correlation between different types of

visualization: the spatial relationships between important structures in 3D space is not fully correlated to the static 2D radiological visualizations.



**FIGURE 1** Comparison of radiologic and endoscopic video stills. (A, C) Radiographic images of sinonasal anatomy were used in Video 1. (B, D) Labeled endoscopic surgery video stills added in Video 2.

**Figure 4.** Clinical training module created by Bailey et al. (2021) featuring static radiographic imaging and endoscopic surgical video stills. Text not intended to be read.

Other teaching resources for otolaryngology surgical trainees include simulations using physical 3D models and virtual reality (Lee A. Y. et al. 2017). Surgical simulators have been widely used in a variety of fields to improve clinical performance by allowing surgical trainees to improve their technical skills by practicing. However, the development of a simulator can be expensive and time- and effort-costly; the availability of equipment and materials can also be limited, which makes this option less desirable.

### *Computed Tomography Imaging*

Computed tomography (CT) is the primary imaging modality to assess sinonasal disease processes and help with intraoperative navigation. CT imaging provides an anatomical “road map” to performing complete and safe surgeries in the sinonasal area by providing precise anatomical information of a patient. As noted earlier, the anatomy of the sinonasal cavity is highly variable. Therefore, prior to any surgical approach, studying the radiological imaging of a patient thoroughly is crucial to identify any preexisting anatomical variants and minimize complications. Typically, CT imaging in both axial and coronal planes need to be acquired to comprehend the anatomical variation and extent of disease.

Being able to interpret radiological imaging correctly is an essential skill among otolaryngology physicians. It is important for surgical trainees to be familiarized with radiological visualization of the sinonasal cavity and identify clinically significant findings.

### *Web-based Interactive 3D Models as a Learning Resource*

In the last 10 years, the use of interactive digital 3D medical models has become an effective tool in supporting anatomical and medical education. Aside from the effectiveness of a digital 3D model in communicating 3D spatial knowledge of an anatomical region, studies suggested that an interactive 3D model that allows manipulation over the model enables better identification of anatomical features from different orientations, compared to passive observational learning of fixed views (Stull A. T. et al., 2009).

Per our research, there is no existing teaching tool featuring interactive 3D visualization of the sinonasal cavity. In this research, we will explore how interactive 3D visualizations with correlations to 2D CT-based visualization can improve comprehension of

the anatomy and spatial orientation of the sinonasal cavity and subsequently improve surgical training outcomes in otolaryngology.

### *Research Objectives*

The main objective of this research is to develop a web-based interactive resource that will offer a comprehensive and multidimensional visualization of the sinonasal cavity. This resource features two learning modes that contain different components:

The **Explore Mode** will consist of four panels featuring: 1) a 3D schematic representation of the nasal cavity with clinically relevant landmarks; 2) a 3D reconstructed model from CT data that provides a realistic representation of the sinonasal cavity within the context of the skull base, and 3) a radiological presentation including a series of CT images in coronal and axial planes. In this mode, all components are fully interactive. Both 3D models can be manipulated in synchronized motion and individual structures can be highlighted, upon selecting labels to turn on and off. The CT image series can be scrolled through, with corresponding planes shown in the 3D reconstructed model to indicate location of the slice. The 3D representations are correlated with the radiological representation in a way that highlighted structures from 3D representations will be reflected on 2D CT images. The Explore Mode allows surgical trainees to navigate the sinonasal cavity in 3D space and bridge the gap between 3D (using schematized and realistic 3D models) and 2D (using axial and coronal CT images) visualizations.

The **Clinical Mode** will consist of four panels featuring: 1) a 3D schematic representation of the nasal cavity with clinically relevant landmarks; 2) a short endoscopic sinus surgery video segment with relevant landmarks labelled, and finally, 3) the

radiological presentation including a series of CT images in coronal and axial planes. The interactive aspect is similar to that in Explore Mode, except for the range of rotation of 3D model will be restrained to a limited range that is consistent with the range of motion of endoscopic camera to mimic the surgical visual field. This mode helps to improve spatial orientation during surgery and correlation between 3D/2D visualization of the sinonasal cavity and clinical perspective.

### *Educational Goals and Target Audience*

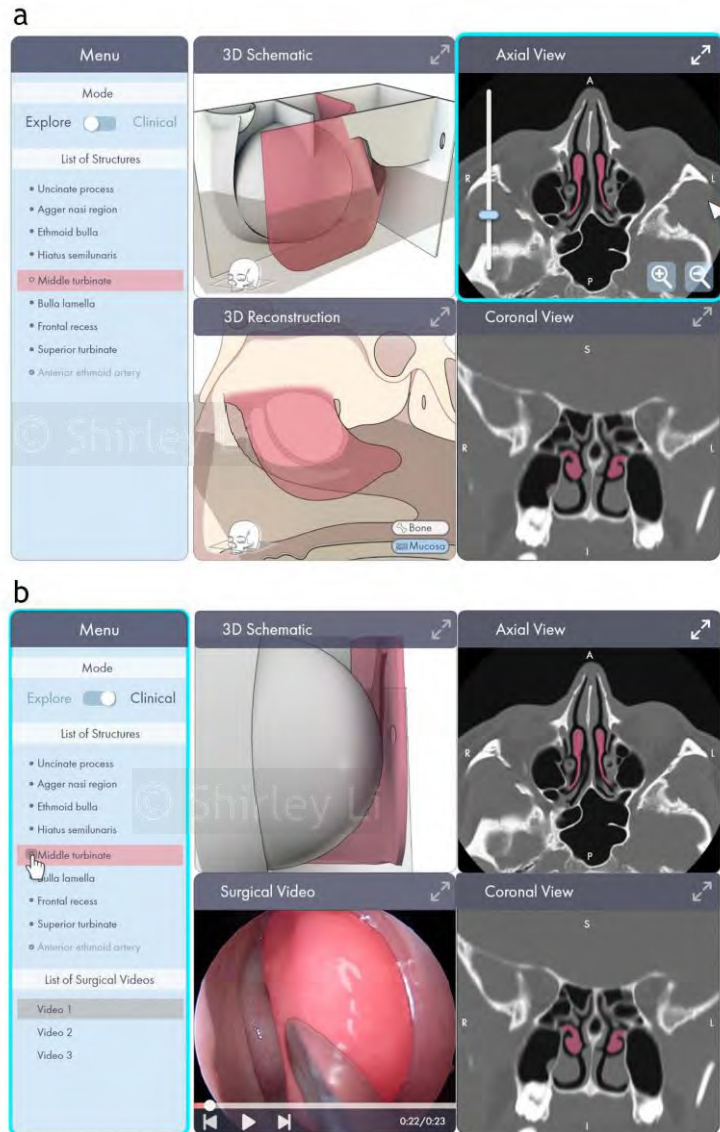
The primary goal of this research is to create an educational tool that can serve to improve surgical training outcomes in otolaryngology by helping surgical trainees to develop a strong foundation in basic and detailed anatomical knowledge of the sinonasal cavity. We hypothesize that surgical trainees will benefit from such a tool incorporating interactive 3D visualization with correlations to different types of media to augment anatomical knowledge of the sinonasal cavity across 3D space and improve spatial orientation under clinically relevant surgical exposure perspectives.

Our primary audience is otolaryngology surgical residents, especially junior residents who need to systematically study and understand the intricacies of 3D sinonasal anatomy and be able to identify structures across radiographic images and during surgery. Our secondary audience includes medical students preparing for otolaryngology residency rotation and medical professionals who are interested in understanding anatomy of the sinonasal cavity.

## Materials and Methods

After meetings and discussions with project preceptors, the basic plan for the layout of this web-based interactive resource was prototyped using Adobe XD (**Figure 5**). This resource would be composed of four main panels featuring different types of visualization and media alongside a navigation panel, where modes can be switched, and visibility of structures can be modified. When switching from Explore Mode to Clinical Mode, the 3D Reconstructed panel would be replaced with the Surgical Video panel and the range of rotation angle in the 3D Schematic panel will become restrained by that seen in the surgical video.

The development of this project involved two stages. In Stage 1, 2D and 3D assets were developed. This included the development of a 3D schematic model and a 3D reconstructed model derived from CT data and rendering of the 2D radiological image series using various software. In Stage 2, the focus was placed on the development of the web-based interactive and creating the interface using Unity. It was agreed upon to prioritize the development of the interactive aspect of the Explore Mode, which is the main teaching module in this project. Afterwards, the Clinical Mode and user interface design were to be developed as time allowed.



**Figure 5.** Planned layout of the interactive resource; **(a)** shows the Explore Mode and **(b)** shows the Clinical Mode. Text not intended to be read.

### Software Overview

A variety of software tools were used to develop the assets and the web-based interactive. 3D Systems DICOM To Print (D2P) (Rock Hill, SC) was used to perform segmentation of CT imaging datasets. Pixmeo OsiriX MD (Geneva, Switzerland) was used to export CT imaging series in both coronal and axial planes. Maxon Cinema 4D R23 (Friedrichsdorf,

Germany) was used to create the 3D schematic representation of the sinonasal cavity. Pixologic ZBrush v.2021.5.1 (Los Angeles, CA) was used to refine and modify the segmented mesh into a reconstructed 3D model with divided structures that could be individually manipulated under interactive setting. Adobe Photoshop CC2021 (San Jose, CA) was used to crop and label structures on the CT image series. Unity by Technologies Unity v.2020.3.24f1 (San Francisco, CA) was used to add interactivity to 3D models and 2D media and build the interface of the web-based interactive resource in this thesis project.

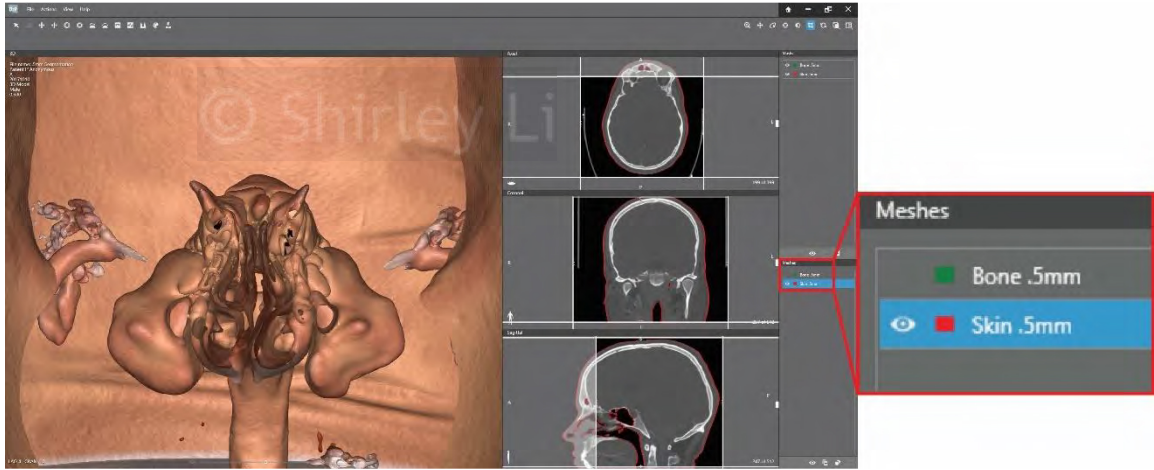
### *Acquisition of Radiological Imaging Data*

A CT angiogram dataset of an anonymous adult head was obtained following an IRB-approved protocol for this project and used in creating a CT-derived 3D reconstructed model of the sinonasal cavity as well as scrollable CT image series in axial and coronal planes. This CT data did not have significant anatomical abnormalities in the sinonasal space and was therefore deemed suitable for the purposes of this project. The CT data was provided by the project preceptor, Dr. Nicholas Rowan, and Dr. Anirudh Saraswathula, from the Department of Otolaryngology-Head and Neck Surgery at the Johns Hopkins University School of Medicine. CT data and surgical video segments used in this project was approved by the Institutional Review Board (IRB) (#CIR00074183) at the Johns Hopkins University School of Medicine.

The CT dataset consisted of a CT head angiogram acquired with and without IV contrast, with a slice thickness of 0.75mm and was saved in DICOM format. Segmentation of this CT data was performed by a research collaborator using D2P for the development of 3D reconstructed model. The soft tissue (containing skin and mucosa) and skull bone were

segmented into two separate 3D meshes (**Figure 6**). These segmented meshes were exported as .obj files to be imported and further processed using ZBrush.

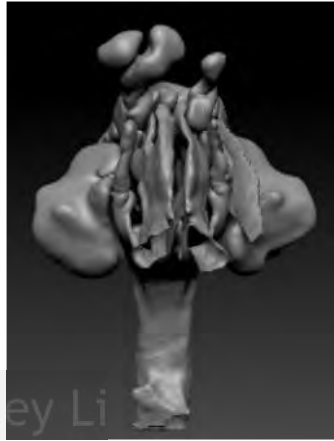
**a**



**b**



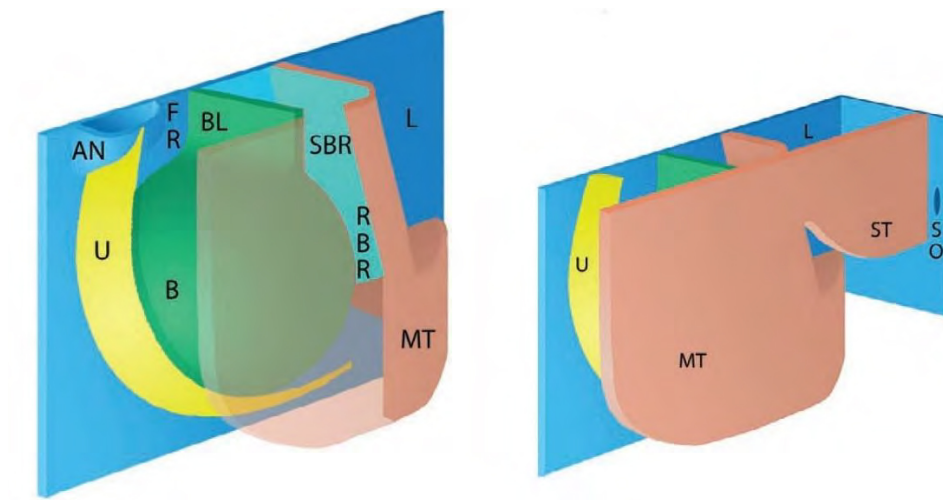
**c**



**Figure 6.** (a) Segmentation of CT head data using D2P and the resulting segmented (b) bone mesh, (c) the negative air space of the sinonasal cavity isolated from the skin mesh in ZBrush. Text not intended to be read.

### *Stage 1: Development of 2D and 3D Assets*

To fulfill the objectives of this web-based interactive resource, it was agreed with the project preceptor that a simplified 3D schematic model is ideal to communicate the complex spatial relationship between clinically relevant structures of the sinonasal cavity. A published article on sinonasal development and anatomy by Wise S. K. et al. (2012) demonstrated a stepwise schematic construction of the ethmoid sinus, highlighting the 3D relationship between critical structures (**Figure 7**). Visualizations from this article served as a blueprint for the 3D schematic model used in this project. In addition, a 3D reconstructed model was to be developed to provide more context around the sinonasal area and to promote correlation between schematic and realistic representation of this anatomical region. To improve correlation between 3D and 2D media, critical structures were highlighted in the 2D radiological image series, so that when a structure is highlighted in the 3D models, the corresponding structure will show up in the scrollable CT image panels.



**Figure 7.** Visualizations on the schematic construction of the ethmoid sinus in a published article by Wise S. K. et al. (2012) Text not intended to be read.

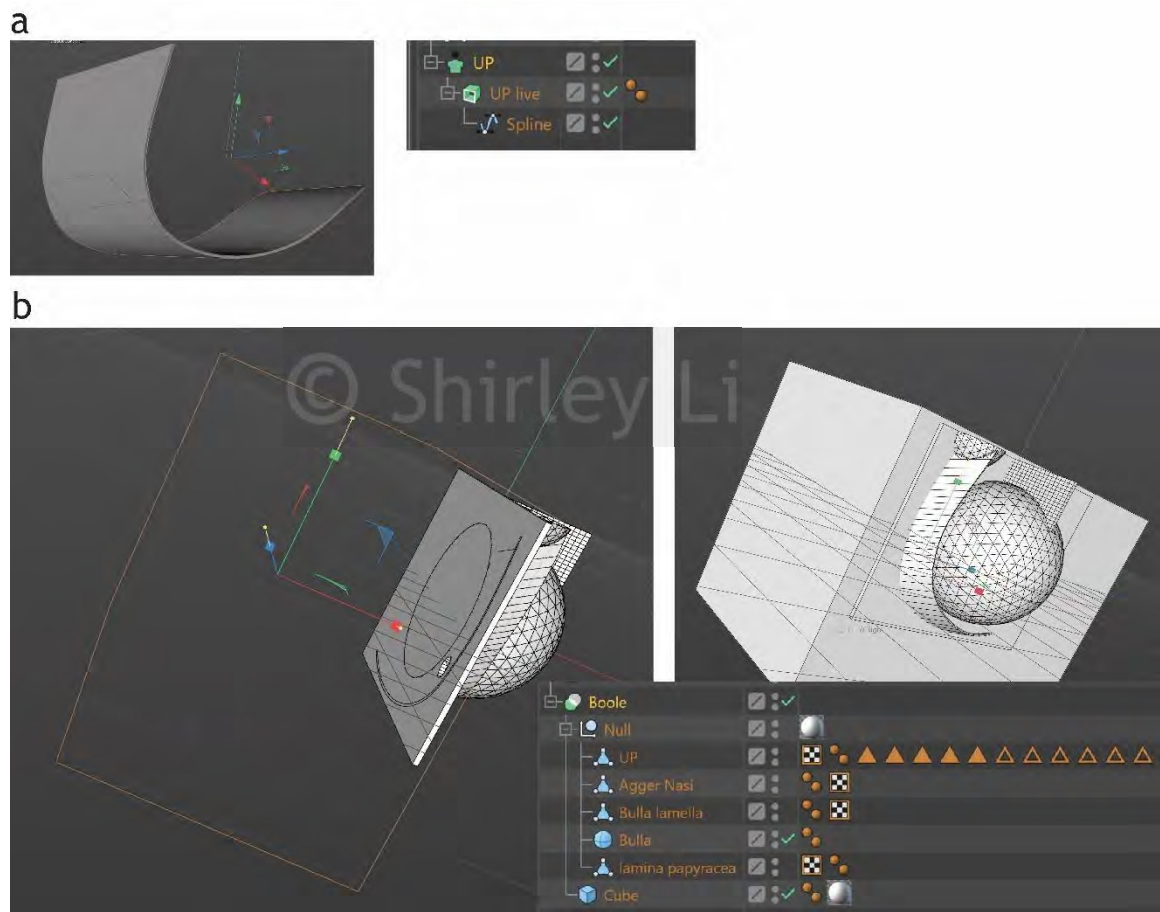
### *Development of 3D Schematic Model*

Cinema 4D was used to create the 3D schematic model. The ethmoid sinus, sphenoid sinus, and critical neurovascular structures were developed. **Table 1** presents a complete list of structures created using Cinema 4D. For each structure, the scale, orientation, and placement relative to other structures were discussed with the project content advisor to ensure the accuracy of the schematic model in conveying spatial relationship between structures.

<b><i>Ethmoid Sinus</i></b>	<b><i>Sphenoid Sinus and surrounding neurovascular structures</i></b>
Lateral nasal wall	Posterior wall of ethmoid sinus
Agger nasi	Sphenoid sinus os
Uncinate process	Clivus
Bulla ethmoidalis	Sella turcica
Bulla lamella	Pituitary gland and pituitary stalk
Middle turbinate	Carotid artery
Superior turbinate	Optic nerve
Maxillary sinus os	Vidian nerve

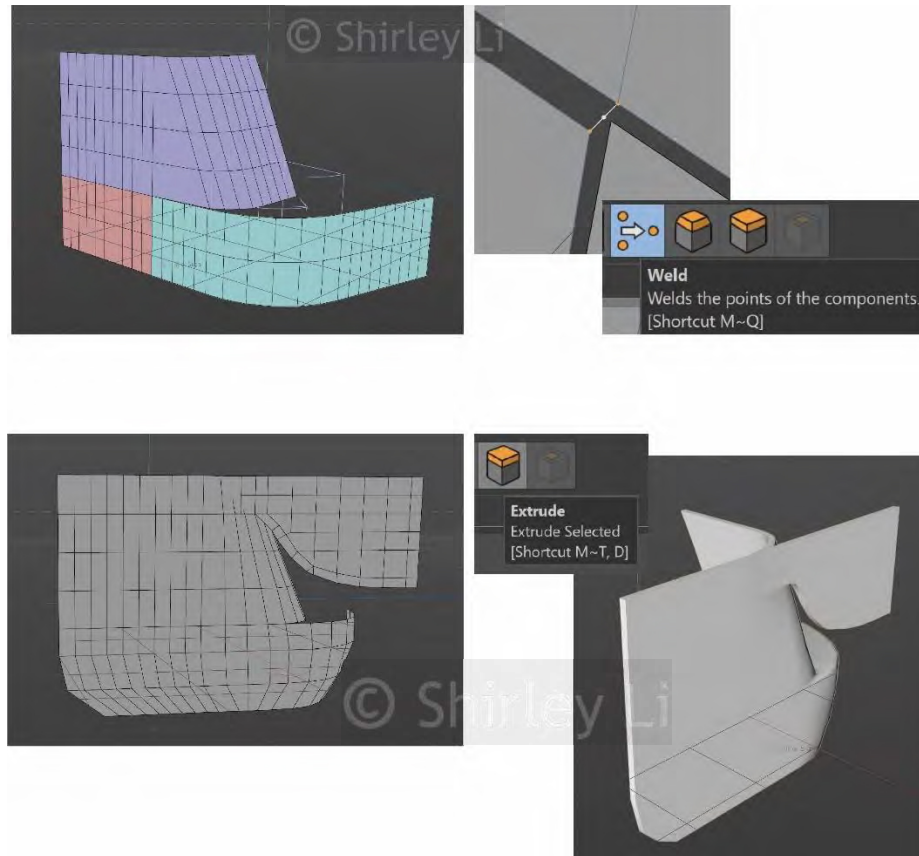
**Table 1.** Structures created for the 3D schematic model using Cinema 4D.

The lateral nasal wall and most associated structures, which include the agger nasi, bulla ethmoidalis, and bulla lamella, were made using primitive 3D objects in Cinema 4D. The uncinate process (UP) was made by creating a spline that follows the trajectory of the UP, which was placed under an Extrude generator. The thickness for the extruded plane was generated by adding a cloth surface, with a thickness of 2 cm. Finally, a smooth lateral surface of the lateral nasal wall was created via a Boolean operation (**Figure 8**). The maxillary os on the lateral nasal wall and the sphenoid sinus os on the posterior wall of ethmoid sinus were also made via Boolean operations using primitive cylinders.



**Figure 8.** (a) Creation of uncinat process and (b) boolean of the lateral nasal wall strucutres in Cinema 4D.

The middle turbinate (MT) and superior turbinate (ST) were developed from primitive planes (**Figure 9**). In the case of MT, three planes were created and were bent into desired shape using a Bend deformer. The three planes were then merged into one object by the Connect Objects + Delete command and the points were welded to form a continuous plane using the Weld tool. Finally, thickness was added by selecting all the surfaces and performing an Extrude operation. The development of ST and walls of the sphenoid sinus followed similar workflow.

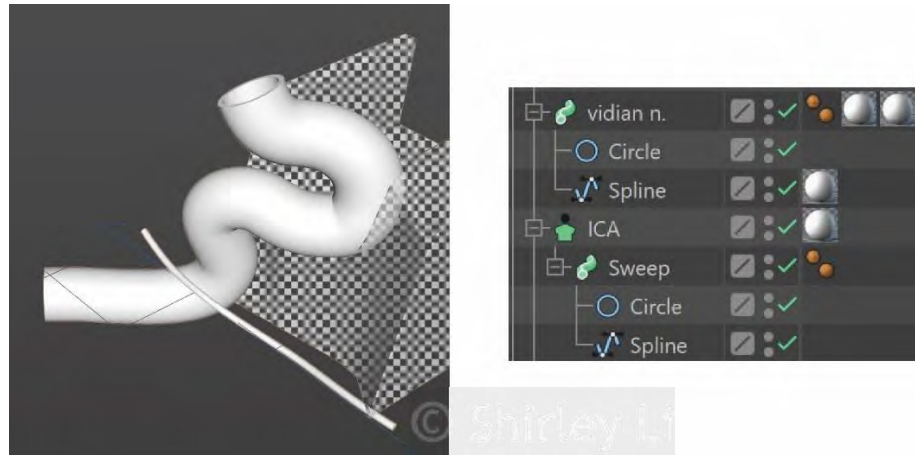


**Figure 9.** *Creation of MT in Cinema 4D: separate planes were combined into one object with points welded and the surfaces were extruded to add thickness.*

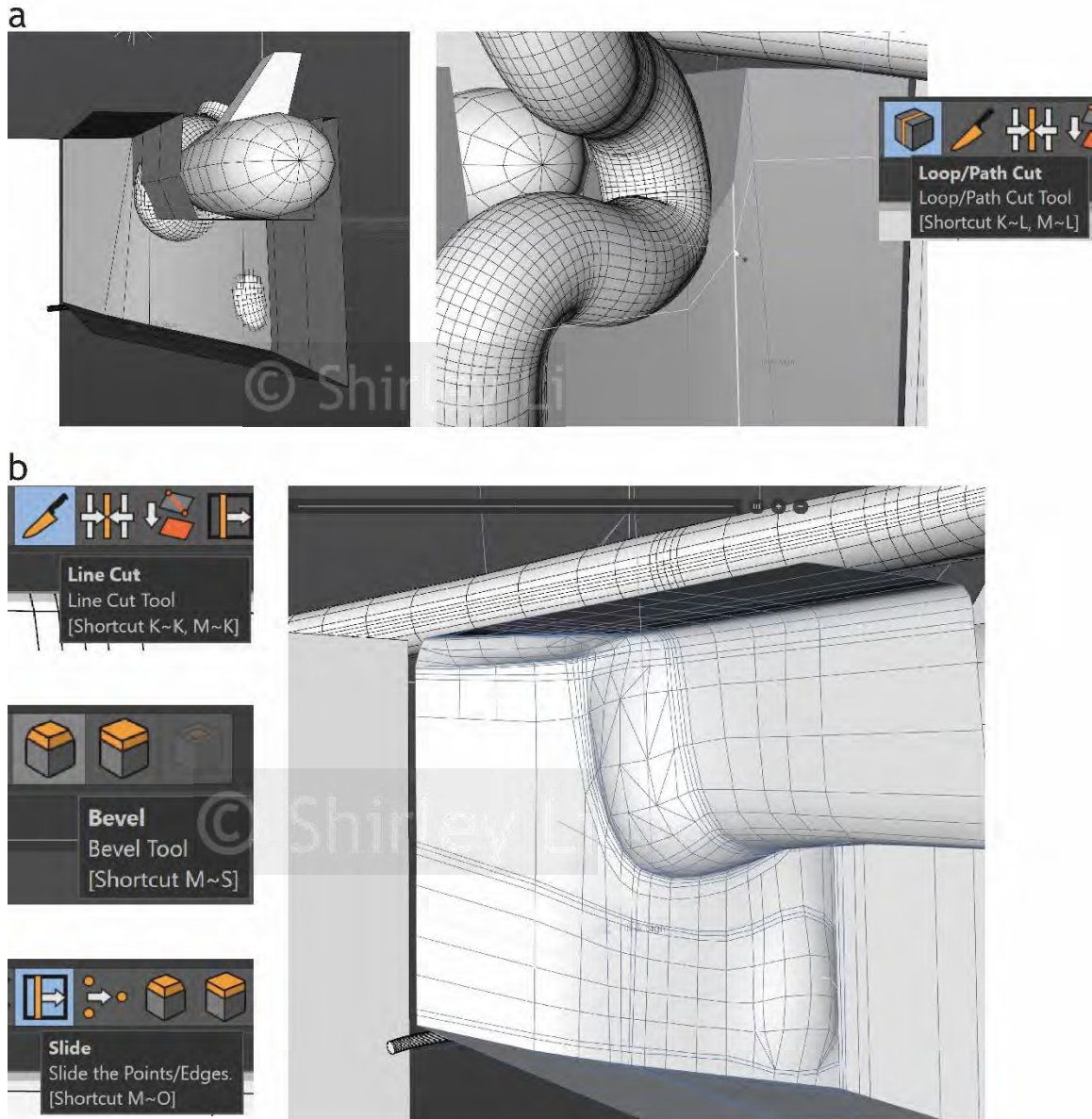
Neurovascular structures were made using the Sweep tool with circles and splines (**Figure 10**). A Cloth Surface was added to the internal carotid artery (ICA) to create thickness to the vessel wall. To create indentation of the ICA and optic nerve (ON) on the lateral wall of sphenoid sinus, the mesh of the lateral wall was first subdivided using Loop Cut and Line Cut tool to create edges around the indentation. The surfaces and points of the indentation areas were moved towards the sinus space to cover the spilling segments of ICA and ON. More edges were then added to the indentation areas using Line Cut tool to achieve a rounded surface and eliminate sharp edges. Finally, the Slide tool was used to

adjust the position of points to minimize odd geometry (**Figure 11**). Thickness was added by selecting all surfaces and performing Extrude operation.

Lastly, the completed model was exported as an .FBX file to import into Unity.



**Figure 10.** Creation of the internal carotid artery and vidian nerve in Cinema 4D.



**Figure 11.** Creation of lateral wall of sphenoid sinus using Cinema 4D. **(a)** Loop/Path Cut tool used to create shapes of indentation around structures. **(b)** Line Cut and Bevel tool used to add edges across the surface of indentation to eliminate sharp edges; final adjustment and polishing of topologies made using the Slide tool.

### *Development of 3D Reconstructed Model*

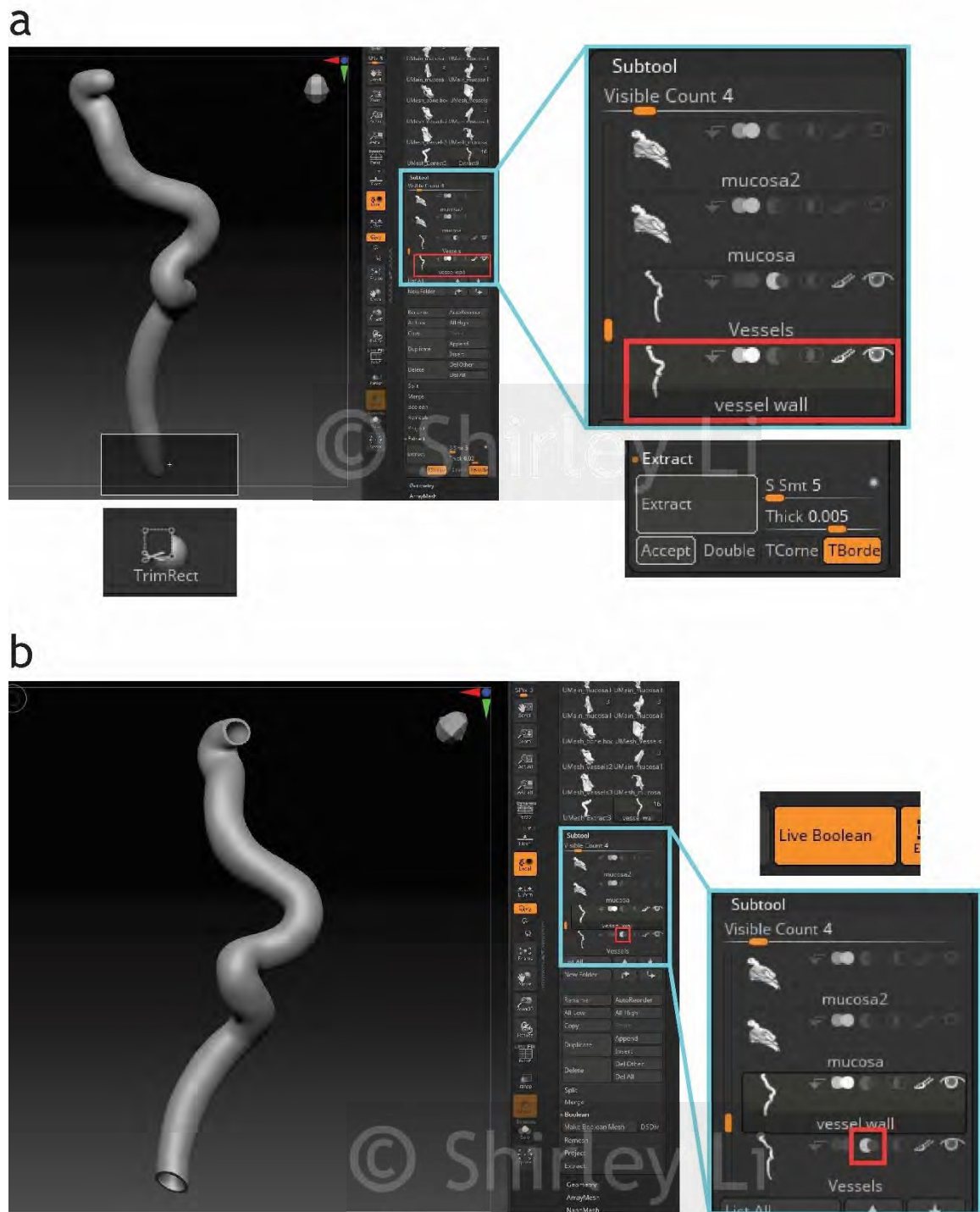
The segmented bone and skin meshes were imported into ZBrush as .OBJ files. The bone mesh contained the cranium while the skin mesh included the skin of the patient's head and the mucosa-lined negative air space of the sinonasal cavity in the center of the mesh (**Figure 12a**). Both meshes were trimmed along midline, preserving the right half of the skull and sinonasal cavity (**Figure 12b**). The cranium was trimmed via a Boolean operation to create cut surfaces, while the mucosa of the sinonasal cavity was trimmed using the TrimRect tool and Select, Hide and Del Hidden commands. The nasal septum was also removed to fully expose the nasal cavity. The major vascular structures of the cranium, which include the ICA and internal jugular vein, were isolated and split into a separate subtool.



**Figure 12.** (a) Cropping of the skull and mucosa mesh along midline in ZBrush and (b) isolation of vascular structures into independent subtool. The sinonasal mucosa mesh was polished in this Figure.

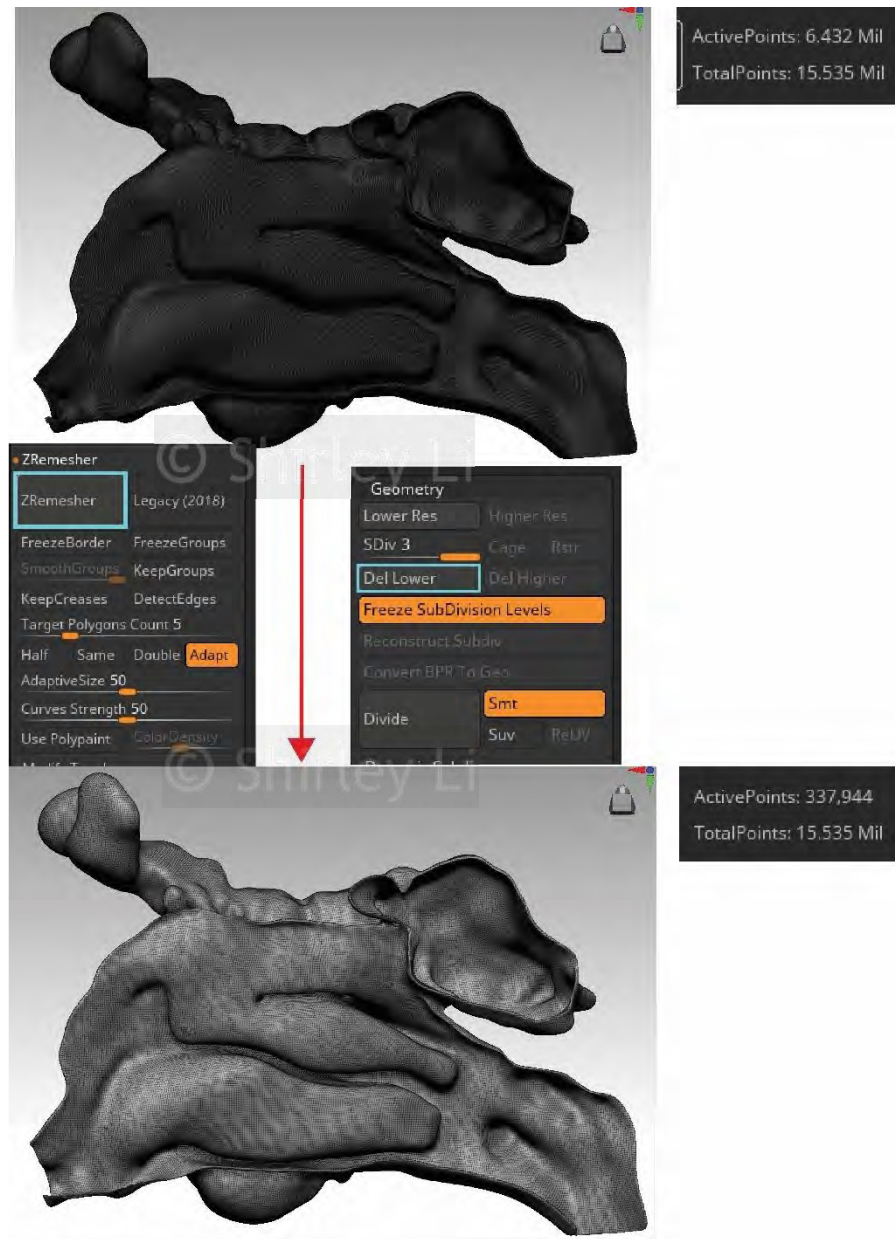
To further prepare the meshes for the project, the mucosa of the sinonasal cavity was first generated by running an extract function on the sinonasal mucosa mesh isolated from the skin mesh using Subtool>Extract, with Thickness set to 0.00178. Upon accepting the extraction, a new mesh with thickness was generated as a new subtool for further

polishing. This was performed using DynaMesh followed by ClayBuildup brush to adjust the geometry of structures. Any unwanted holes in geometry were closed by using a combination of Move Topological and Inflate brushes, followed by a re-Dynamesh operation. Next, the cranium was further cropped to appropriate size using Boolean operations and the surface was smoothed out using the Smooth brush. Gaps were filled in by appending and sculpting 3D cube primitives followed by a DynaMesh operation. The geometry of the cranium model was then further optimized using the ClayBuildup and Smooth brushes. Finally, the ICA was isolated by removing segments of internal jugular vein using the SelectLasso tool and Del Hidden command. After cleaning up the surface of the ICA to create a vessel lumen, the ICA was duplicated and the duplicated mesh was extracted using Subtool>Extract, with a Thickness of 0.005 (**Figure 13**). The cross section of the ICA was created by trimming off the blind ends using the TrimRect tool. With the original subtool sitting below the extracted subtool and switched to subtraction mode, a Boolean operation was performed to create the vessel wall of the ICA.

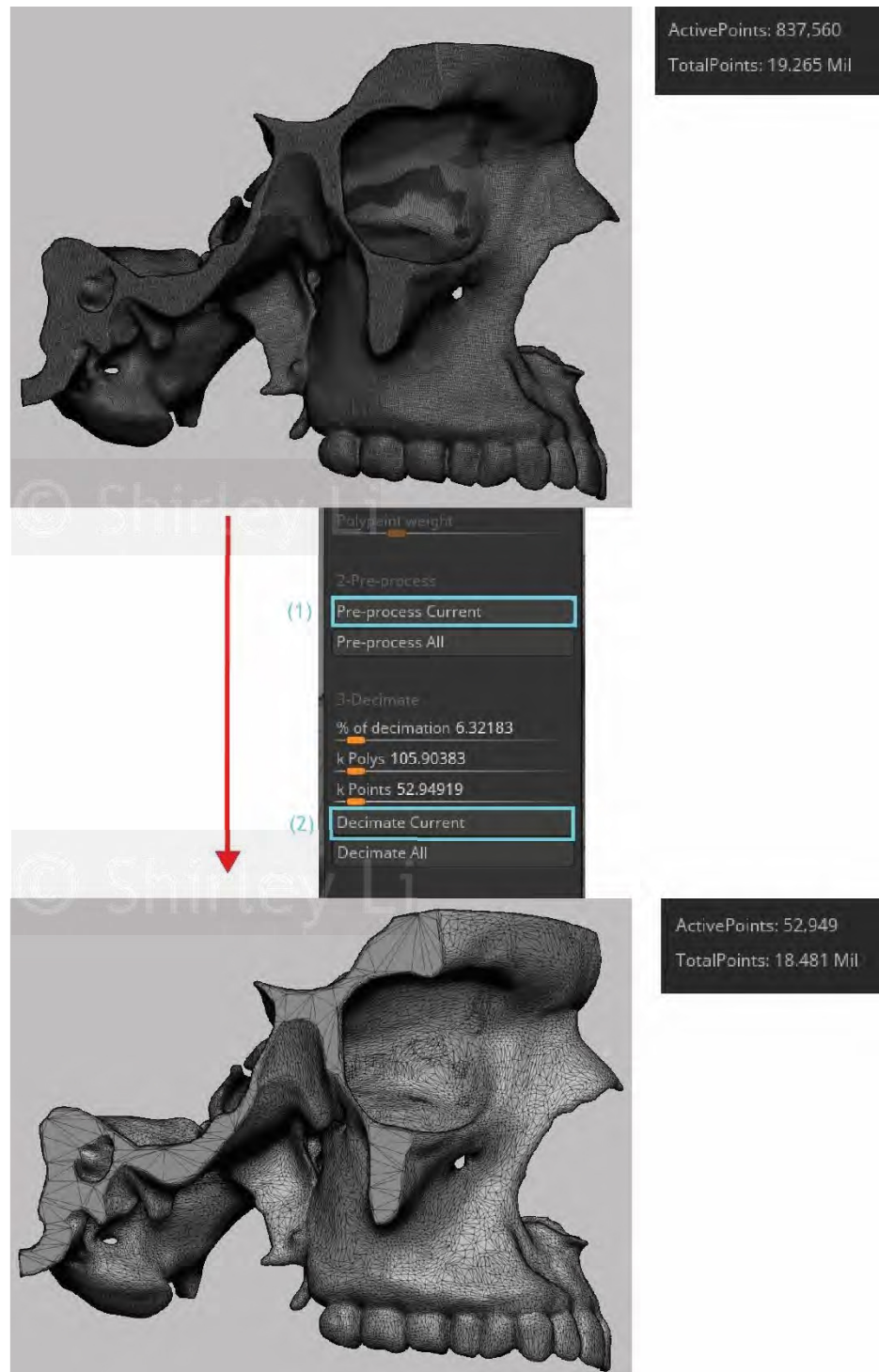


**Figure 13.** Reconstruction of the ICA in Zbrush. **(a)** Extraction of duplicated mesh followed by creation of cross section using TrimRect tool. **(b)** Boolean performed to create vessel lumen.

Reducing polygon count of generated 3D models was essential to improve render performance in interactive uses. Therefore, poly count was reduced in ZBrush for each created mesh. After finishing sculpting and adjusting geometry, the mucosa mesh was duplicated and re-meshed using Geometry>ZRemesher (**Figure 14**). Subdivision was then increased via Geometry>Divide several times (2 times in this case) until achieving optimal resolution. Finally, lower subdivision was deleted using Geometry>Del Lower command to obtain the final reduced poly mesh. However, to reduce polygon counts for the cranium mesh, the Decimation Master function was used since ZRemesher was not as effective in keeping details of some structures, such as teeth. Decimation master is efficient in reducing polygon count while preserving details and is located in Zplugin>Decimation Master menu. Upon completion of pre-processing, a percentage of decimation value of about 6.7 was sufficient to preserve all details while significantly reducing polygon count (**Figure 15**).

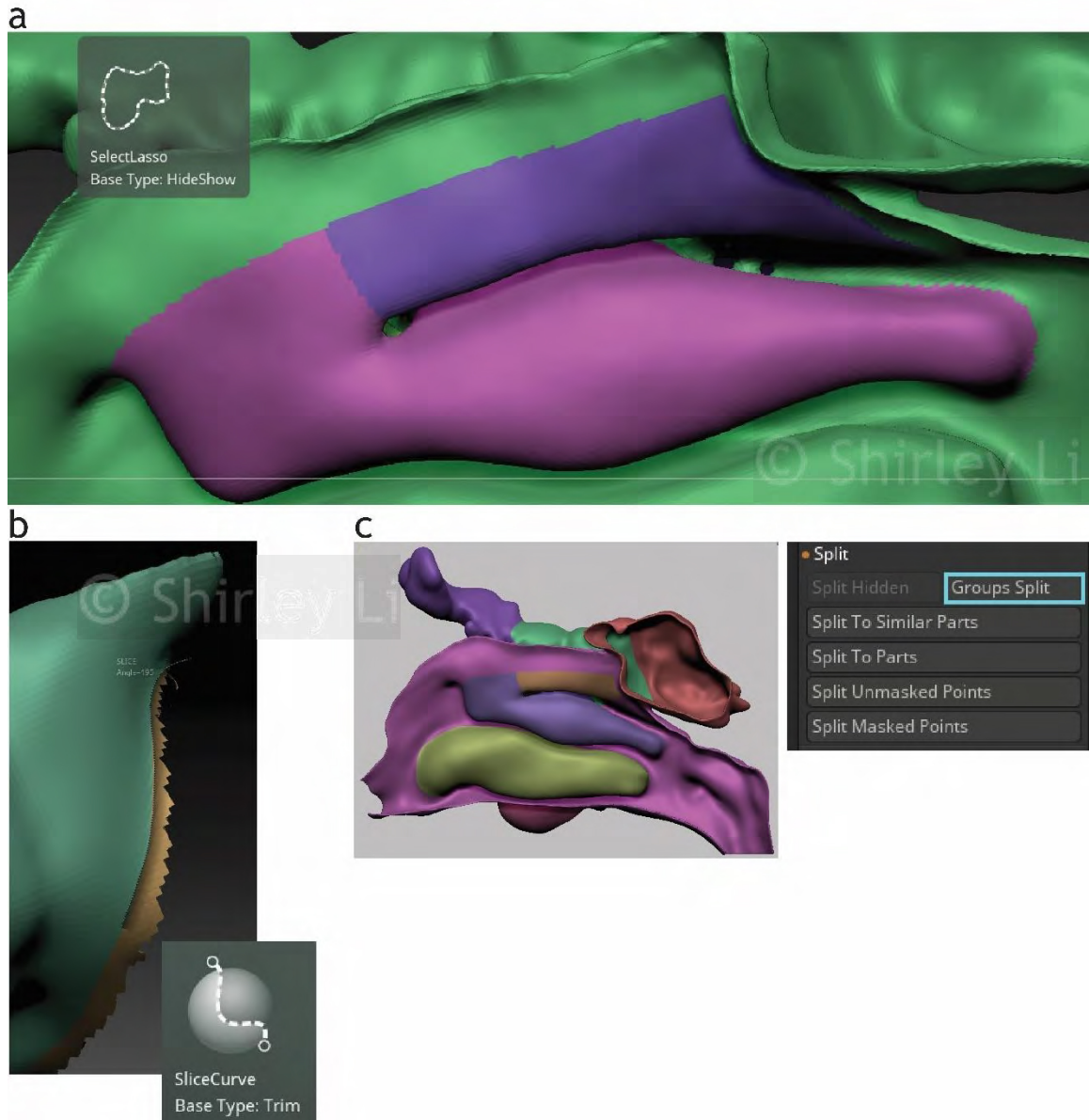


**Figure 14.** Reduction of polygon count of sinonasal structures within ZBrush using ZRemesher. Left and right images showing before and after polygon reduction, respectively.



**Figure 15.** Reduction of polygon count of cranium in ZBrush using Decimation Master. Left and right images showing before and after polygon reduction, respectively.

To allow for independent manipulation of structures in Unity, each structure of the sinonasal cavity needed to be divided into separate subtools and exported as separate models. To achieve this, different structures were first identified, and their boundaries were marked into different polygroups using the SelectLasso tool and make polygroup command. Each polygroup would then be represented by a different color across the mucosa mesh (**Figure 16**). After confirming correct identification of structures with the project preceptors, the boundaries were generated using the SliceCurve tool to obtain sharp clean edges. Upon completing structures divisions, each polygroup was split into a separate subtools using Subtool>Split>Groups Split command.

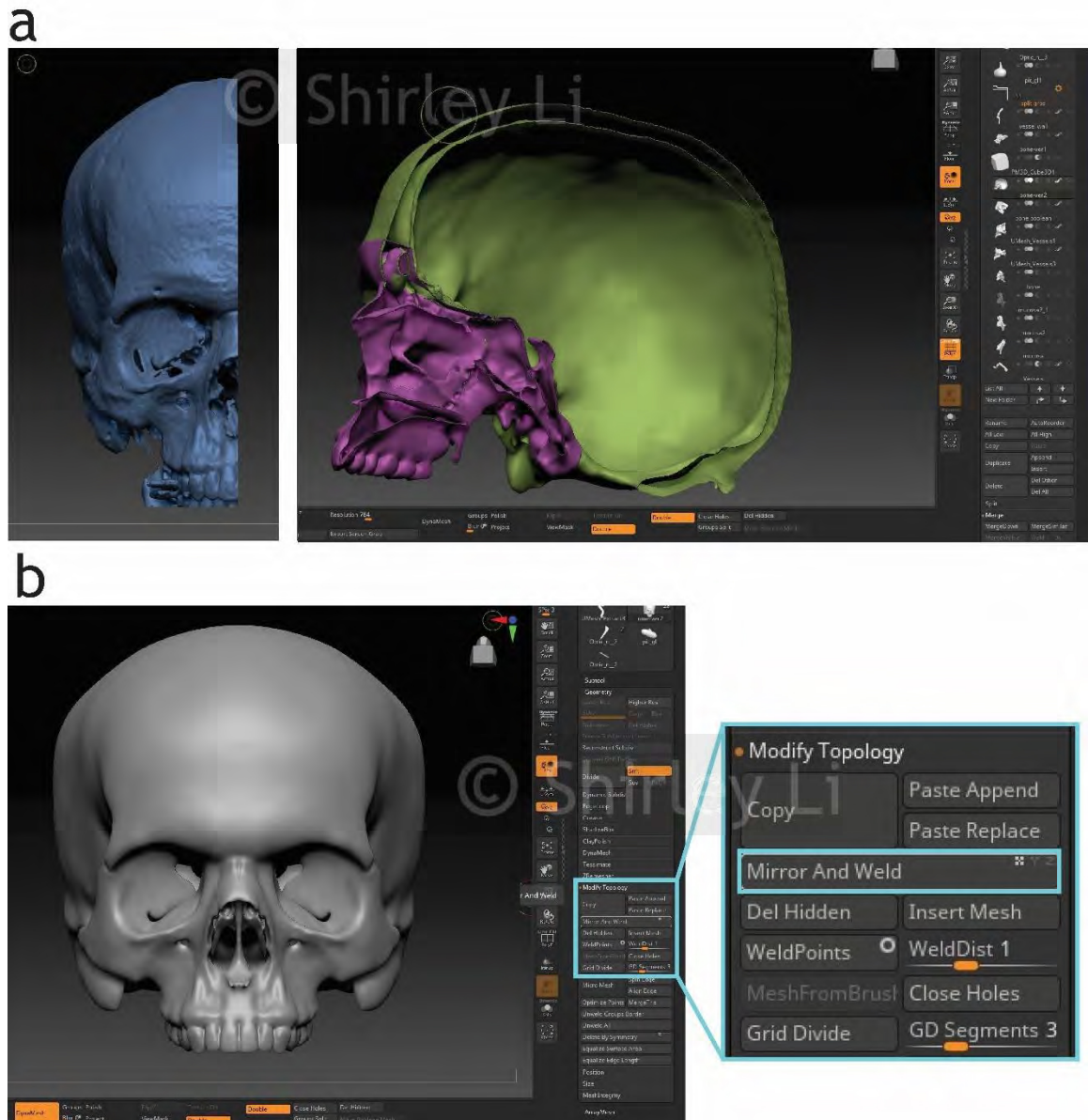


**Figure 16.** Dividing mucosa mesh into different polygroups using ZBrush to differentiate anatomical structures. **(a)** Boundaries between structures defined using the SelectLasso tool and make polygroup command. **(b)** Boundaries further defined using the SliceCurve tool; surrounding structures were hidden. **(c)** Upon completion of structures division, polygroups were split into individual subtools.

In addition to the sinonasal cavity and cranium models, a complete cranium model was also developed in ZBrush. This cranium serves as an orienting reference key figure and would be placed at the bottom left in each 3D model panel in the final product (**Figure 5**). First, the segmented cranium derived from CT data was trimmed along midline using the TrimRect tool. Half of the cranium was polished with unwanted structures removed, and the other half was generated using Modify Topology>Mirror and Weld to form a complete cranium (**Figure 17**). The poly count was reduced using ZRemesher, following the workflow described previously.

Other skull base structures such as the pituitary gland and optic nerve, were imported from Cinema 4D as .OBJ files into ZBrush, where geometry was further adjusted and sculpted to match the fidelity of the CT-reconstructed model.

Finally, all subtools were exported as .fbx files to be imported into Unity.

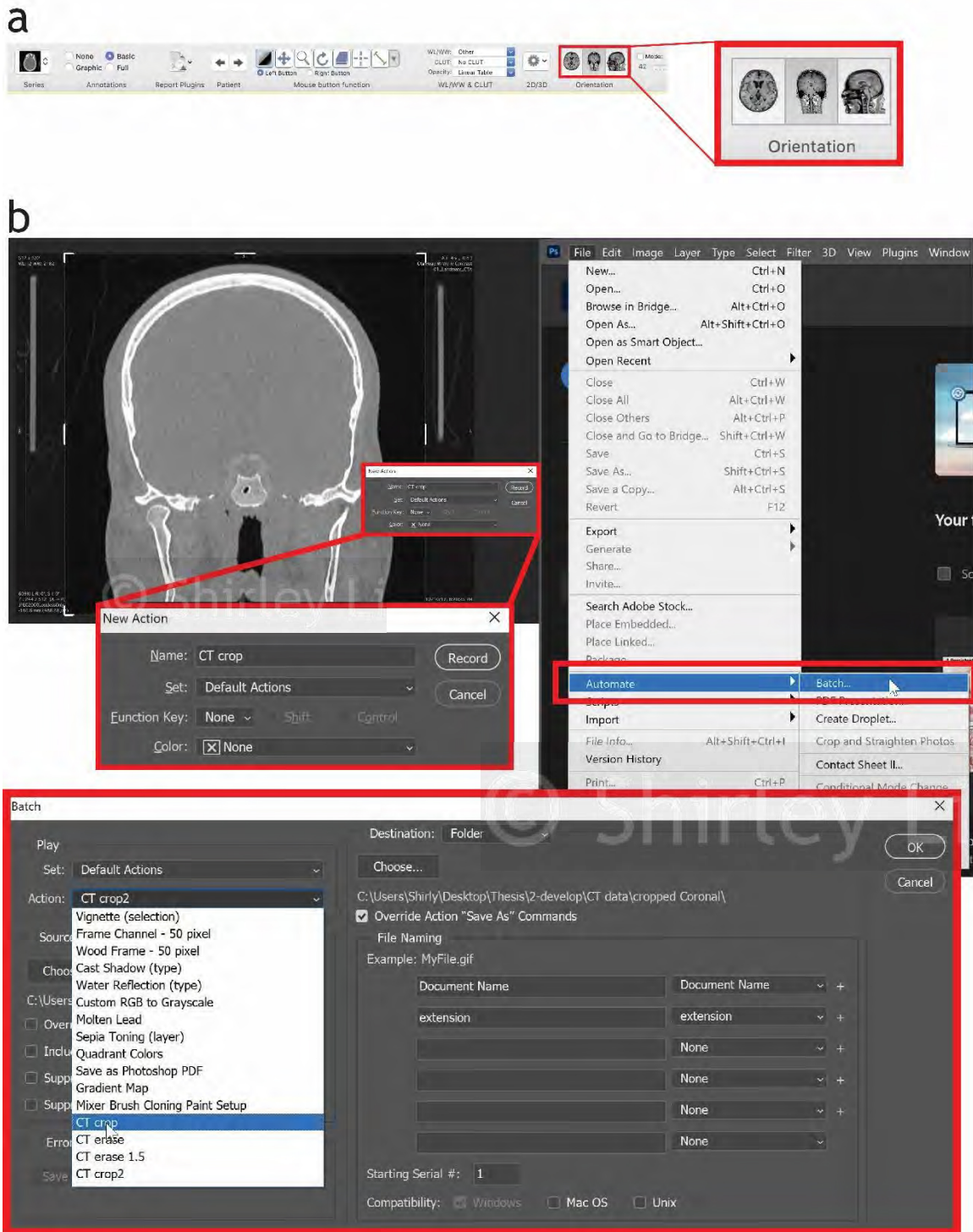


**Figure 17.** Reconstruction of reference cranium using ZBrush. (a) Cranium was trimmed along midline and polished. (b) Mirror and Weld was performed to generate a complete cranium.

### *Preparation of the CT Image Series*

The CT image series were exported as .JPG files from OsiriX MD, using the menu File>Export>JPEG. Both the coronal and axial view image series were separately exported by selecting the different orientation (i.e., axial, coronal) on the tool bar in the OsiriX viewport (**Figure 18a**). Exported CT images were then assessed by project preceptor to determine the range of slices featuring sinonasal structures. The number of images were then reduced by an alternating selection of every fourth image.

The images were subsequently post-processed using Adobe Photoshop, via cropping to exclude labels of the CT dataset. The cropping action was recorded and saved by performing a cropping action to one of the CT images, followed by a batch-processing of the entire image series using File>Automate>Batch, with the newly created cropping action selected (**Figure 18b**). The batch-processed files were then saved as .JPG files to a designated folder for importing into Unity.



**Figure 18.** (a) Selection of coronal and axial orientation of CT images using Osirix MD for export into Photoshop. (b) Batch-processing of CT images as .JPG files using Photoshop. Not all text intended to be read.

## *Stage 2: Development of Interactive Content*

In the second stage of development, focus was placed on developing interactive content using Unity. The process included setting up the user interface (UI), importing assets into Unity, and develop interactivity using C# scripting.

### *User Interface and Interactive Design*

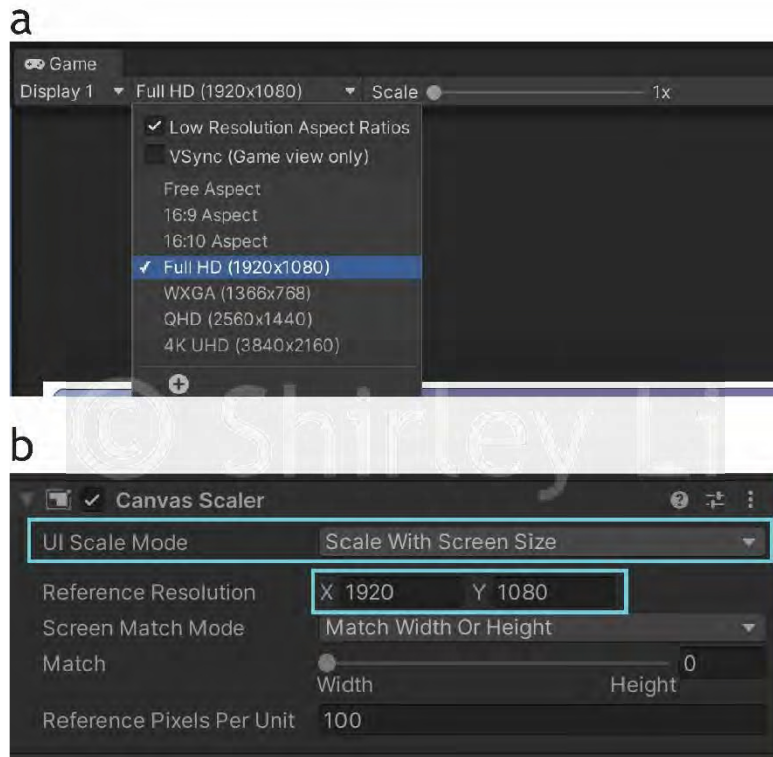
The UI of this web-based interactive resource consisted of a navigation panel on the left and four panels featuring different didactic content (**Figure 5**). The two panels on the middle row feature two different 3D representations of the sinonasal cavity—the schematic and CT-reconstructed models, with the panels residing on the right of 3D representation panels featuring radiological image sequences in axial and coronal views. In each didactic panel, a banner with the title of the content sits at the top of the panel, with a Full Screen button located on the top right corner. In each 3D representation panel, a reference cranium was placed at the bottom left in each panel to help users orienting their view. Within each radiological image panel, a slider was incorporated to scroll through the image series. Furthermore, CT planes in axial and coronal planes were incorporated into the 3D Reconstructed model. The position of the planes relative to the model indicates the location in 3D space of a slice seen in the corresponding radiological (axial and coronal) image panels. Within the 3D Reconstructed panel, buttons were created to toggle off the CT planes and were located beside the Full Screen button in the banner area. Finally, within the navigation panel, users can navigate between modes and access the List of Structures.

Interactive content developed for this project covers most of the interactions in the Explore Mode, which include: 1) synchronized manipulation of 3D models (rotation, panning, and zooming), 2) reflection of selected structure in both navigation panels and

radiological image panels, 3) scrolling through radiological image series using a slider (with CT planes moving as the slider value changes, indicating the level of a specific slice in the 3DReconstructed model), and 4) toggling structures on and off from the navigation panel. When switching to Clinical Mode, the 3D Reconstructed panel would be replaced with a Surgical Video panel. Planned interactive content for the Clinical Mode include: 1) pause and play controls of a video player, 2) an additional section containing a list of surgical videos appearing next to the surgical video on the navigation panel. Further, the rotation of the 3D schematic model would be limited to a confined range within the 3D Schematic panel and selection of a structure would cause the video to jump to the corresponding frame that features that structure. Most of the interactive development for Clinical Mode was not completed in the scope of this research due to time limitations.

### *Setting up the User Interface*

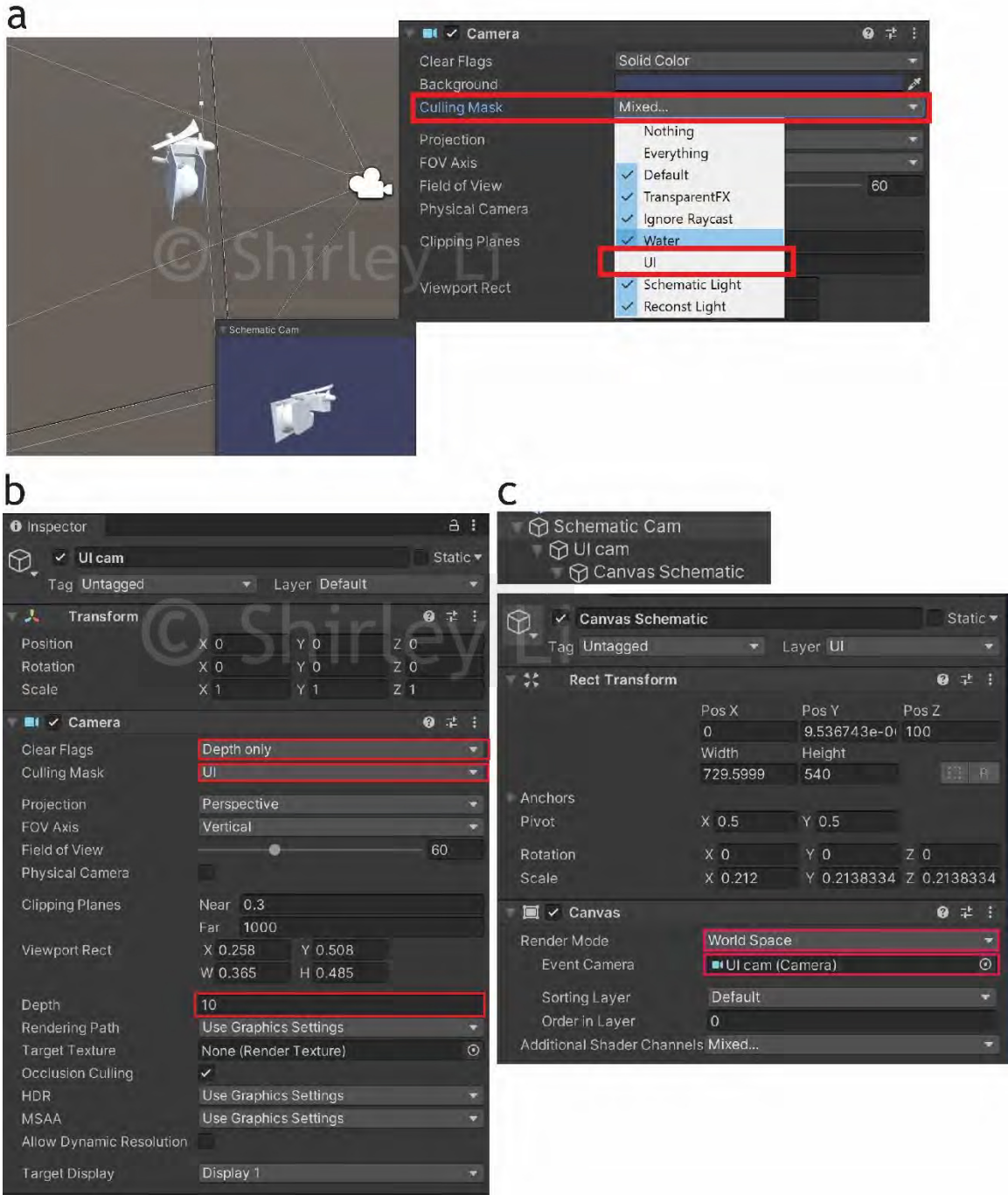
Within a newly created scene in a 3D project, a canvas was created using the menu GameObject >UI>Canvas. This canvas would be the main canvas and include all the UI elements and windows in this game. First, a fixed aspect ratio needs to be designated; this was achieved through the dropdown menu within the Game view (**Figure 19a**). In this case, the resolution was changed to Full HD (1920x1080). Under Free Aspect by default, resolution changes with the dimension of the game window, leading to shifting of UI elements and messing up of the interface layout. Hence, it was crucial to select a fixed aspect ratio to begin with and avoid changing as project develops. Further, the Reference Resolution of the main canvas was also adjusted to 1920x1080 by selecting Scale with Screen Size under Inspector window>Canvas Scaler>UI Scale Mode (**Figure 19b**).



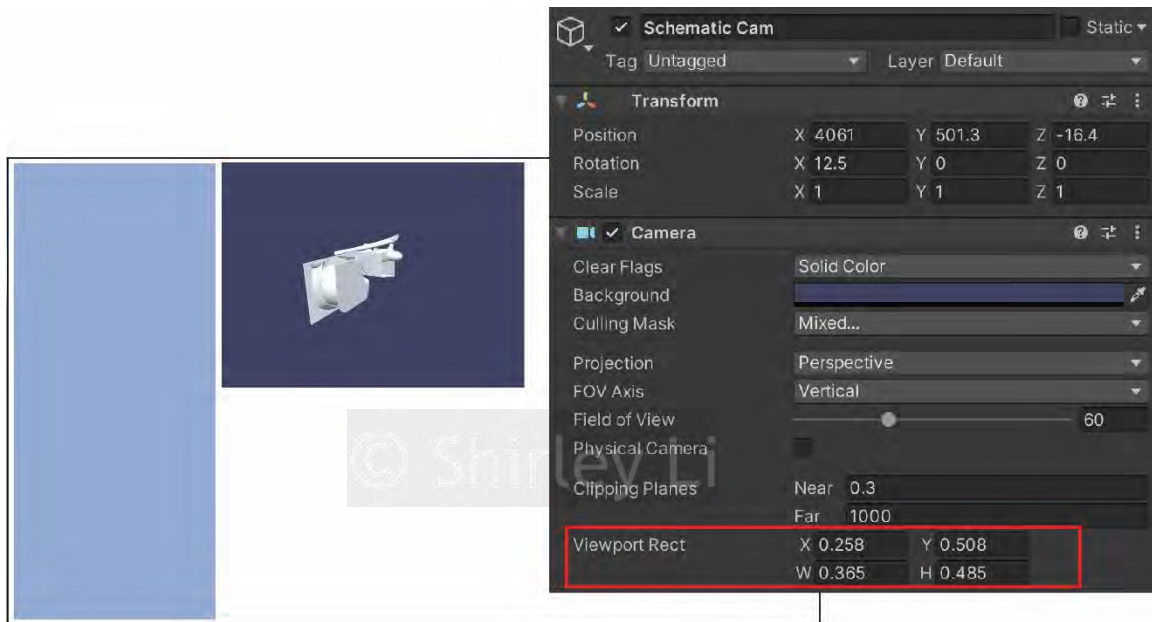
**Figure 19.** Adjusting aspect ratio using Unity. (a) Resolution changed to Full HD from Free Aspect in Game view. (b) Under the Inspector window of the main canvas, the UI Scale Mode switched to Scale with Screen Size and Reference Resolution adjusted to 1920x1080.

To create the navigation panel that hosts the List of Structures and Mode toggle, a panel was created using `GameObject>UI>Panel` and was placed on the left side and made as a child of the main canvas. The panels for didactic content, however, were set up via multiple camera viewports. First, to set up the 3D Schematic panel, a new camera was created, and the schematic model was placed within the camera view (**Figure 20a**). This camera served as the object camera and would only be capturing the object in the world space, in this case the schematic model. Therefore, under the Inspector window, UI was unchecked under the drop-down menu Culling Mask. Next, a UI camera was created as a child of the object camera, with its position and rotation values under Inspector window>Transform set to 0s to overlap with the object camera. This UI camera would only capture the UI elements within the panel; thus, UI was the only item checked under Culling

Mask in this case (**Figure 20b**). Further, Depth Only was selected under the drop-down menu Clear Flags for the UI camera, and a value of 10 was entered under the Depth field. The depth value of a camera dictates the rendering order of the camera view, such that the greater the value, the later it would be rendered. A greater depth number was assigned to the UI camera to make the UI camera view appear on top of the object camera view. Under the UI camera, a canvas was created as a child using the menu GameObject>UI>Canvas (**Figure 21a**). The canvas would be the place to position all the UI elements in the panel. The Render Mode of the canvas was set to Screen Space – Camera under the Inspector window, and the UI camera was assigned to the Event Camera field. Afterwards, the Render Mode was changed to World Space so that the canvas behaves like other objects in the scene view. Finally, the position and dimension of the camera viewport was adjusted through the Inspector window>Viewport Rect of the object camera (**Figure 21b**). After the camera viewport was adjusted to desired size and location, the Viewport Rect values were copied and pasted to the Viewport Rect field of the UI camera.



**Figure 20.** Setting up 3D Schematic panel through camera viewport. **(a)** New camera created and the schematic model placed within camera view, with the Culling Mask excluding the UI item under the Inspector window. **(b)** A UI camera created as a child of the previous camera; the Culling Mask included UI only, and Clear Flags set to Depth only and Depth set to 10. **(c)** A canvas created under UI camera. With Render Mode set to Screen Space – Camera, the UI camera assigned, and Render Mode later switched to World Space. Not all text intended to be read.

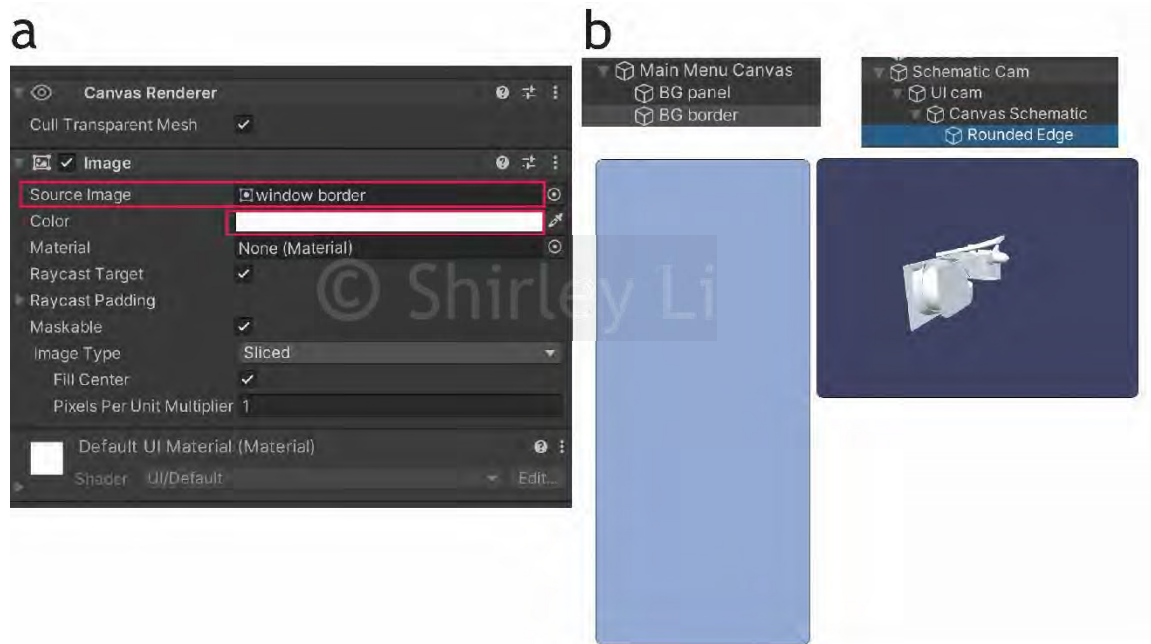


**Figure 21.** Adjusting values in the Viewport Rect field under the Inspector window of the object camera to achieve desired dimension and position of the camera viewport on the main canvas. Values were then copied to the UI camera as well. Shown on the left is the result seen under Game view.

### *Creating Rounded Border for Each Panel*

A circle with black stroke was created using Adobe Photoshop; it was then prepared as a 32x32 .PNG file and imported into Unity as a 2D sprite (**Figure 22a**). To activate Sprite Editor, the 2D Sprite package must be installed using menu Window>Package Manager>2D Sprite>Install. Within the Sprite Editor, a 9-slicing was performed so the sprite could be used at various sizes (**Figure 22b**). In this case, the four slices involving the rounded corners would remain unchanged, while pixels in the other slices would change when the sprite was scaled. To apply the rounded border to each panel, a panel was created as a child under each of the main and schematic canvas, using GameObject>UI>Panel (**Figure 23**). Finally, the sprite was assigned as a Source Image under the Inspector window>Image>Source Image, with color adjusted to white at full opacity to obtain rounded corners.





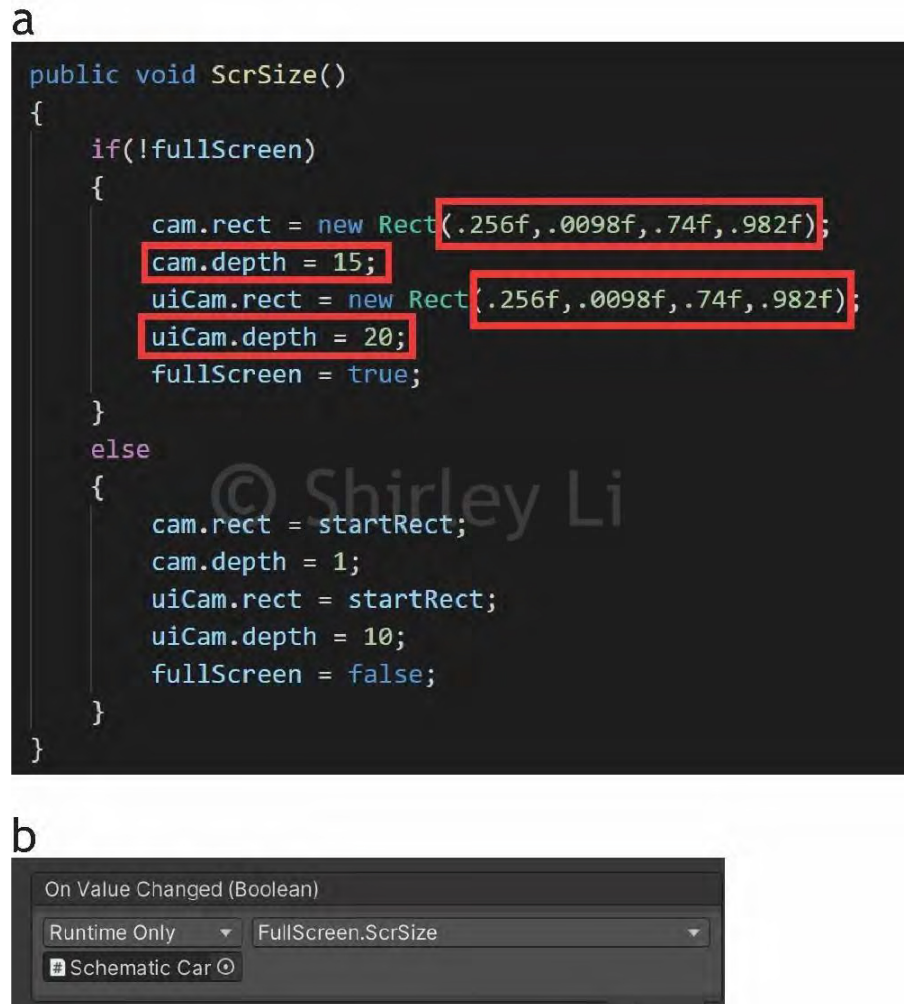
**Figure 23.** Applying rounded border to panels. (a) Within the panel, the sprite was dragged into the Source Image field with Color changed to white at full opacity. (b) Resulting rounded corners for each panel under Game view. Arrangement of UI elements in the Hierarchy window also shown for navigation and 3D schematic panels.

### *Creating the Full Screen Button*

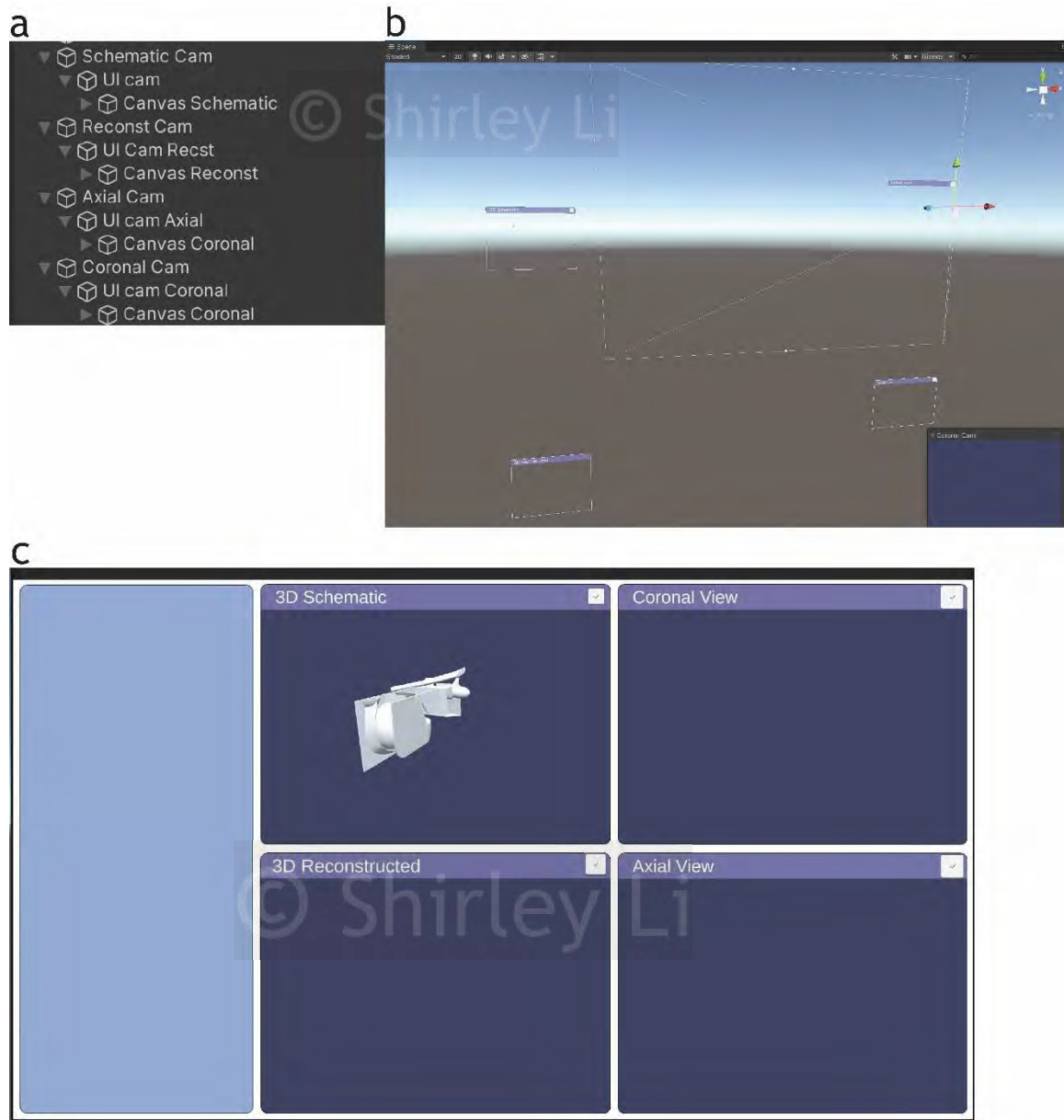
To create the full screen button for each didactic panel, a Toggle was created using `GameObject>UI>Toggle` as a child of the canvas under the UI camera. The full screen function was established by attaching a C# script, the “FullScreen” script, to the object camera that changes the Viewport Rect and depth values of the cameras to make the viewport enlarge and sit on top of other viewports (**Figure 24**). Finally, under the On Value Changed field of the Toggle, the object camera with the script attached was dragged into the Object input and the full screen function was selected from the dropdown menu.

Upon incorporating the full screen button, this 3D Schematic camera viewport containing all the basic UI elements could serve as a template for other didactic panels. Thus, the object camera of the 3D Schematic panel was duplicated three times and placed

into position by adjusting the Viewport Rect values (**Figure 25**), thus completing the basic set up of the UI.



**Figure 24.** Establishing full screen function. (a) Screenshot of part of the Full Screen script. When toggle is activated, the Viewport Rect and depth values of the camera change. (b) Under Inspector window of full screen toggle, a full screen size function was assigned.

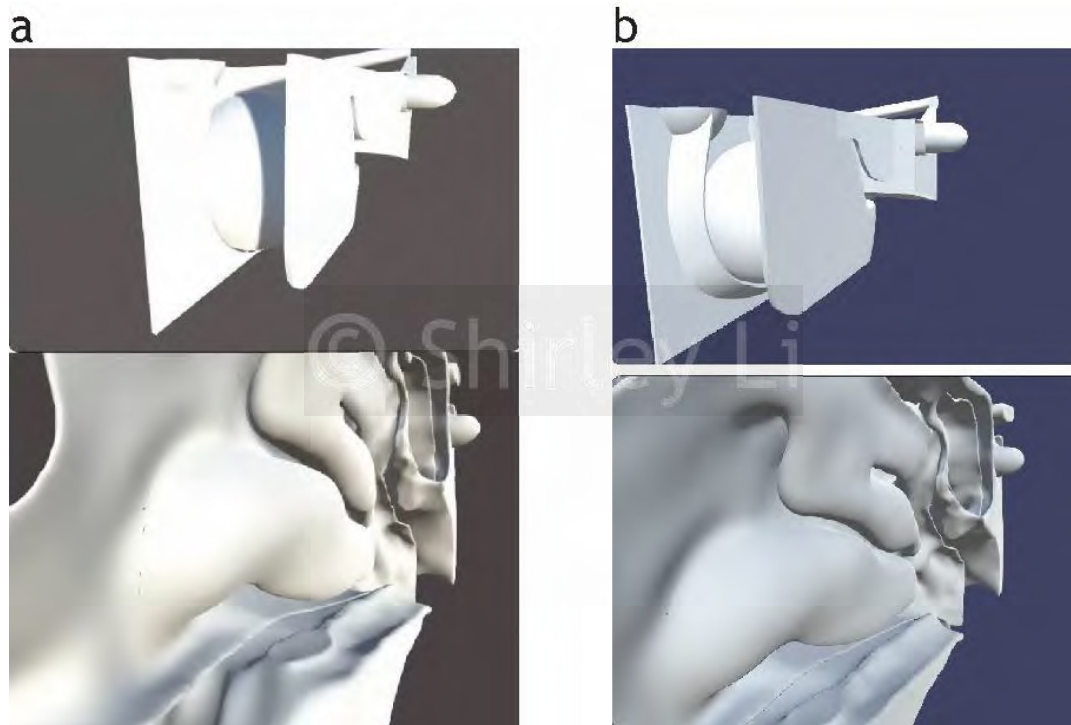


**Figure 25.** Duplicated first camera viewport for completed basic UI set up. Camera viewports arrangement under (a) Hierarchy window, (b) Scene view, and (c) Game view. Not all text intended to be read.

### *Importing 3D Models into the Scene*

The 3D schematic, CT-reconstructed, and orienting cranium models were imported into Unity and were brought into their corresponding camera viewports. Next, the scale and orientation of the models were adjusted to match each other. The orienting cranium model was duplicated and made a child of the canvas under the UI camera of each of the 3D Schematic and 3D Reconstructed object cameras. To avoid overlapping of the orienting cranium with both 3D models of the sinonasal cavity, in the drop-down menu of the Inspector window the Layer was set to “UI” for the orienting cranium.

To achieve a similar lighting effect on the schematic and reconstructed models, instead of a single universal light source, different light sources were used for each model (**Figure 26**). First, all the child objects under the schematic model were switched to a newly created layer called “Schematic Light” and child objects of the reconstructed models were switched to another layer called “Reconst Light”. Next, the default directional light was duplicated. Under the Inspector window of one light, the “Schematic Light” was unchecked under the Culling Mask drop-down menu, whereas the “Reconst Light” was unchecked within the other light, such that each light only affects its corresponding model individually. Finally, within the Inspector window under the corresponding object camera for each 3D model, both newly created light layers were included under the Culling Mask drop-down menu.



**Figure 26.** 3D models before (a) and after (b) incorporating multiple individual light sources.

### *Synchronized Rotation and Panning of 3D Models*

To allow coordinated rotation and panning manipulation of both 3D models, colliders were added to each individual structure under the 3D schematic and CT-reconstructed models. For structures with more complex shape, either a mesh collider that takes the form of the mesh or multiple colliders of various shapes were applied. In addition, a “Structure” tag was created and assigned to each structure via the Inspector window>Tag>Add Tag...>Structure. Next, the “StructureControl2” script was created, and a series of public variables were declared. This included a ControlCamera variable and three public GameObject variables to allow for synchronized rotation and panning of the three objects (two orienting craniums and the 3D schematic and CT-reconstructed models) (**Figure 27a**). Further, a Raycast was done to allow controlling objects when the ray hits a tagged

“structure” with colliders. This allows for controlled manipulation of models when actively clicking on them. This script was then attached to the 3D schematic model and CT-reconstructed model. Under the inspector window, each of the two corresponding object cameras was assigned as the control camera, through which the three models would be manipulated in concert (**Figure 27b**). This script allows for rotation on left-click and panning on right-click of the mouse.

a

```

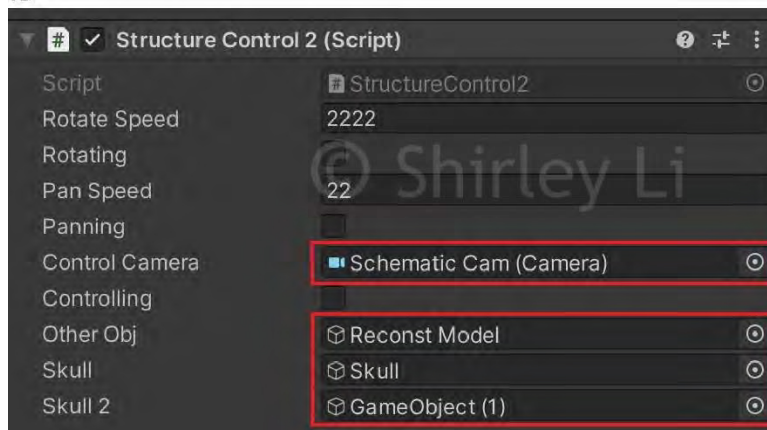
public Camera controlCamera;
public bool controlling = false;

public GameObject otherObj;
public GameObject skull;
public GameObject skull2;

// Update is called once per frame
void Update()
{
    if (Input.GetMouseButtonDown(0) || Input.GetMouseButtonDown(1)) //right click OR left
    {
        Ray ray = controlCamera.ScreenPointToRay(Input.mousePosition);
        RaycastHit hit;
        if (Physics.Raycast(ray, out hit) && hit.transform.tag == "Structure") //if raycas
        {
            controlling = true;
        }
    }
    if (Input.GetMouseButtonUp(0) || Input.GetMouseButtonUp(1))
    {
        controlling = false;
        rotating = false;
        panning = false;
    }
}

```

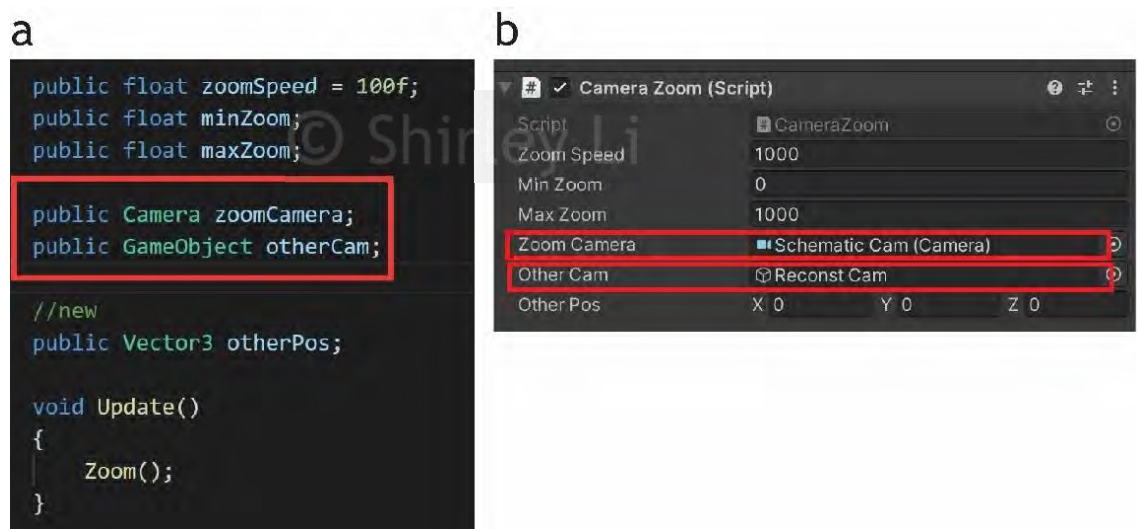
b



**Figure 27.** Script for synchronized rotation and panning of models. (a) A screenshot of the “StructureControl2” script highlighting the public *GameObject* variables for synchronized manipulation of models and a Raycast. (b) Attaching the script to the schematic model as an example, with the corresponding camera and models also assigned under the Inspector window.

### *Zooming of 3D Models*

The zooming function was established with the script “CameraZoom”. This script allows synchronized zooming of models at the position of the mouse on scrolling the mouse wheel. In this script, only one main public Camera was declared and this would be the camera that the script was attached to, with the camera for the other model declared as a public GameObject, to avoid both rays casted by two cameras hitting the same game object at the same time (**Figure 28**). The script was then attached to both cameras for the schematic and reconstructed models.



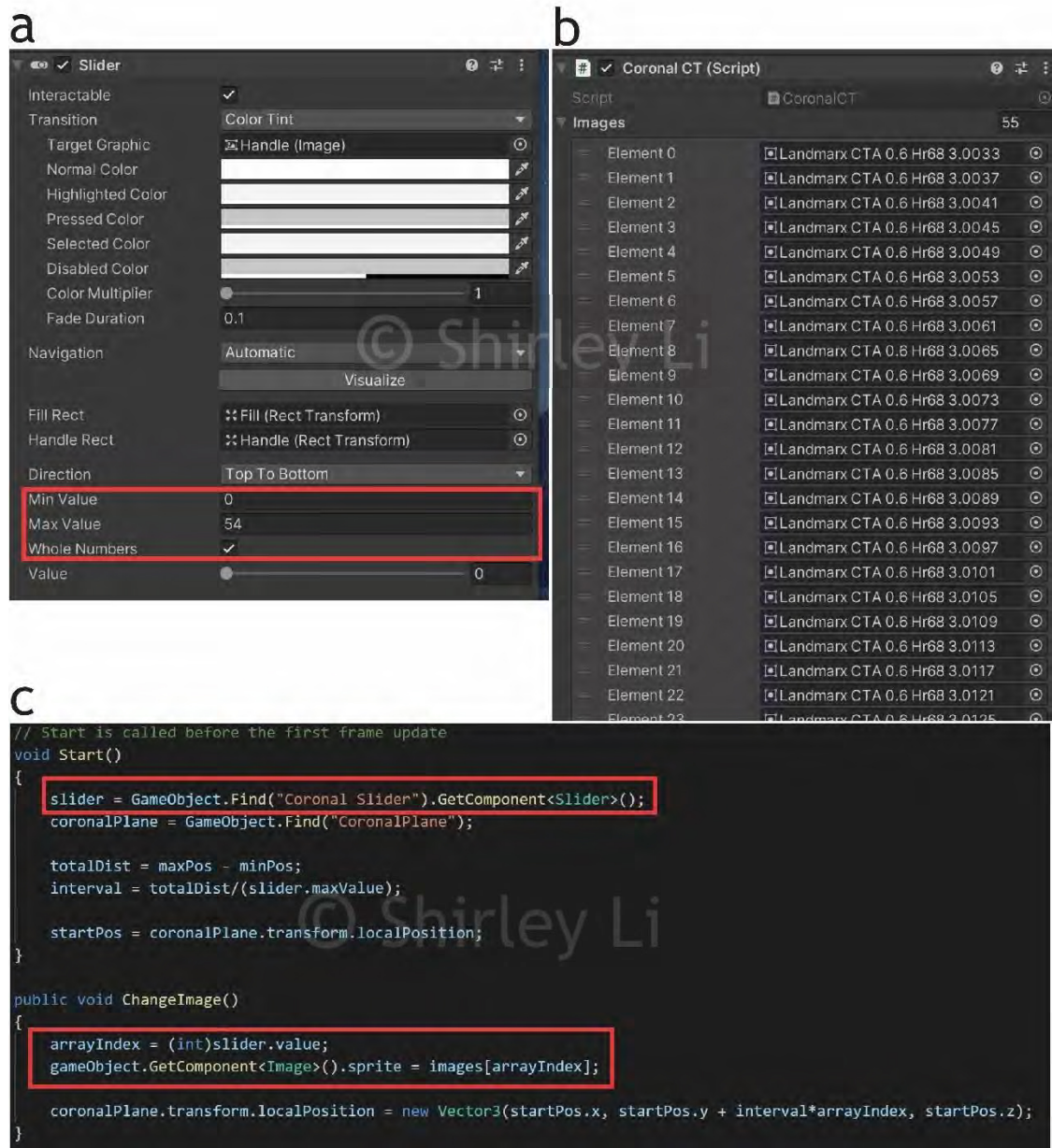
**Figure 28.** Script for synchronized zooming of models. (a) A screenshot of the “CameraZoom” script showing only one public Camera variables declared from which zooming control information was gathered. (b) Attaching the script to the schematic model as an example, with the schematic model camera assigned as the Zoom Camera and the reconstructed model camera assigned as Other Cam.

### *Creating Scrollable Radiological Image Series*

The Coronal and Axial CT image series were imported into Unity, with Texture Type changed to Sprite (2D and UI) for all images. Under each UI camera of the radiological image panels, an Image and a Slider UI object were created via GameObject>UI>Image/Slider as child objects of the canvas under the UI camera. For each slider, the Min Value was set to 0 and Max Value set to the total number of images in the array minus one, (i.e., 54 images in the coronal plane, with Whole Numbers checked under the Inspector window (**Figure 29a**). The “CoronalCT” and “AxialCT” scripts were attached to the corresponding Image UI object, and each respective image array was loaded into the Images field under Inspector window>Coronal CT/Axial CT (Script)>Images (**Figure 29b**). The Change Image function was then assigned to the slider by associating the corresponding Image UI object that contains the script under the Inspector window of the slider. This function allowed for changing of CT images upon slider value changes (**Figure 29c**).

To enable zooming and panning within the CT image panels, an Empty Object Parent was created under the canvas to house the Image UI object containing the CT image array. A Scroll Rect component was added to this parent object within the Inspector window with the Image UI object dragged into the Content field. This allowed for panning of 2D radiological images. Next, the Width and Height value of the Image UI object was adjusted to the original dimension of the image, and a Rect Mask 2D component was added to the parent object and scaled to cover the entirety of the image viewport. The “ImageZoom” script was then attached to the Image UI object and the Image UI object was scaled down to fit the window, defining the minimal dimension the image could be zoomed out within the game. Finally, values were adjusted to change the initial position and scale of

the CT images, under the Image Zoom script component within the Inspector window of the Image UI object, such that upon initiating the game CT images were zoomed into the sinonasal area.



**Figure 29.** Setting up scrollable CT image series, with the coronal CT image panel shown as an example in this figure. (a) Input of Min and Max Value with Whole Numbers checked under the Slider component in the Inspector window. (b) Importing image array into the Images field of the Script component under the Inspector window of the Image UI object. (c) A screenshot of the “CoronalCT” script highlighting the Change Image function.

### *Incorporating CT Slice Orientation Planes in the 3D Reconstructed Model*

Two perpendicular CT slice orientation planes were created using the menu GameObject>3D Object>Cube as child objects of the 3D reconstructed model. Upon transforming and scaling the planes into desired initial positions and size, customized materials were applied with “Transparent” selected under the Rendering Mode drop-down menu within the Inspector window. For each plane, positions of the initial and the final slice of the CT image (for both axial and coronal arrays) were determined in space, and the change in Position value of the axis as it moves along was recorded (**Figure 30a**). Within the “AxialCT/CoronalCT” script, the Position value of the first and last image in the image array was input as the initial default values for the public float variables, minPos and maxPos, respectively. When the slider value within either the Axial or Coronal CT image panel was changed, the script allowed for movement of the corresponding CT slice orientation plane in the 3D Reconstructed Model panel in a designated direction by changing the positional value of the axis.

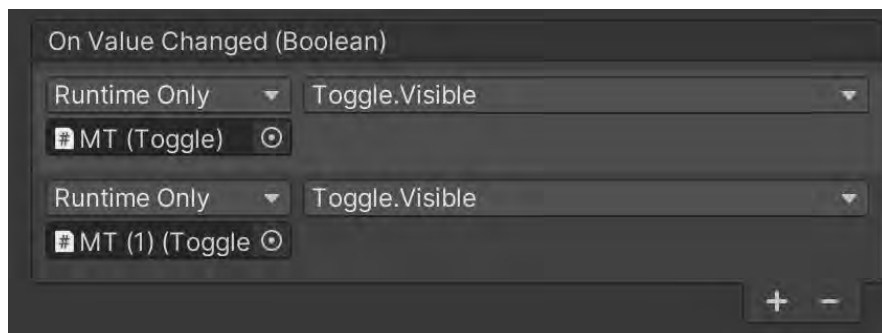
To toggle the CT planes on and off in the 3D Reconstructed panel, a Toggle UI object for each plane was added under the UI Camera canvas of the 3D Reconstructed camera and placed beside the Full Screen view button. Additional scripting was added to the “AxialCT/CoronalCT” script to enable the toggling function to make the planes invisible until the CT sliders were moved within the CT image panels (**Figure 30b**). This function was then added to the Toggle UI object under the Inspector window>On Value Changed (Boolean) by associating the corresponding Image UI object.



**Figure 30.** Script for moving CT planes in 3D Reconstructed view. The axial CT plane was shown as an example in this figure. **(a)** A screenshot of the “AxialCT” script showing the input of minPos and maxPos values, determined by the position value of the initial and last image in the array. **(b)** The part of the script allowing for on/off toggling of the CT plane.

### *Creating Toggles to Hide Structures*

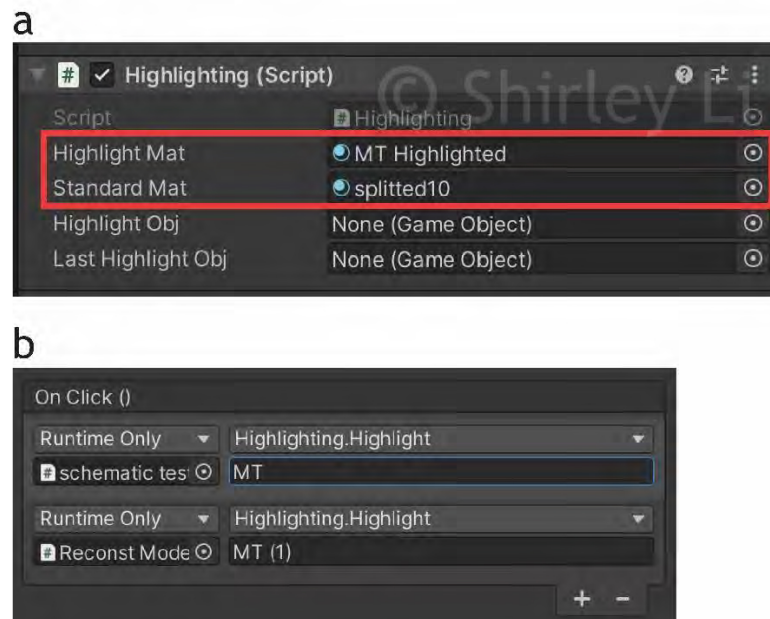
To develop a list of structures for the navigation panel, a series of Toggle UI objects were created and labelled with the names of anatomical structures in the 3D models. An on/off “Toggle” script was created and attached to each individual structure under the 3D model parent objects. Under the Inspector window for each of the toggle objects listed in the navigation panel, two events were added to the On Value Changed field (**Figure 31**). These affect on/off visibility changes of the same structure in both the 3D schematic and reconstructed model panels. With the Toggle Visible function assigned from the drop-down menu, the script allows for turning-off visibility of the corresponding structure in both 3D models, and when the toggle object was toggled off from the List of Structures in the navigation panel.



**Figure 31.** Upon attaching “Toggle” script to each individual anatomical structure, two events were added to the toggle object with corresponding structures from both models assigned.

### *Highlighting a Structure by Selecting an Item from the Navigation Panel*

A script was created to change the material of a structure upon selecting the structure name from the List of Structures under the navigation panel. First, under the main canvas, a parent object was created for each toggle object in the navigation panel. A button UI object was created and was placed under the parent object of the toggle object. Next, the “Highlighting” script was attached to the parent objects of both 3D schematic and reconstructed models, and the original and highlighted materials were assigned to the corresponding field under the script component within the Inspector window (**Figure 32a**). Under the On Click field of each button object, two events were added, with 3D schematic and reconstructed models associated with an event. To enable highlighting of that structure in both models upon clicking on the button, the Highlight function was assigned, and names of the structure object in both models were entered into text fields. (**Figure 32b**).



**Figure 32.** Attaching “Highlighting” script to parent object of the 3D model. **(a)** Assigning materials to corresponding field. **(b)** Assigning highlighting function to each button to enable highlighting of structures on both 3D models upon activating the button.

### *Switching between Explore and Clinical Mode and Incorporating Video Component*

To establish the toggle button that switches between modes, a Slider UI object was created as a child object of the main canvas. Under the canvas of the panel that would become the Surgical Video panel upon switching to Clinical Mode, a Raw Image UI object was created. In addition, a Video Player was created using `GameObject>Video>Video Player`. Next, a surgical video saved in .MP4 format was provided by the preceptor and imported into a Video folder in Unity. A Render Texture was created under the same folder by right-clicking on the Project window>Create>Render Texture. The video was then dragged into the Video Clip field within the Inspector window of the Video Player, with the newly created Render Texture assigned as Target Texture. Further, the Render Texture was assigned as Texture within the Inspector window of the Raw Image object to project video output onto itself.

Upon switching from Explore Mode to Clinical Mode, a List of Videos (representing different case studies) would appear under the navigation panel, while the 3D Reconstruction panel would be replaced with a Surgical Video panel. To achieve this, upon switching to Clinician Mode, the “ModeControl” script allows changing the label of the 3D Reconstructed panel to “Surgical Video”, activates the List of Videos and the video viewport, and hides the 3D reconstructed model and orienting skull (**Figure 33**). This script was then attached to the default main camera, with the corresponding objects assigned under the script component within the Inspector window. Lastly, the List of Videos included a series of Buttons allowing users to switch between different videos. This is achieved through an additional “VideoSwitch” script that was attached to the Video Player. Within the Inspector window of each button, the Video Player was dragged into the On Click field where the “VidSwitch” function was assigned.

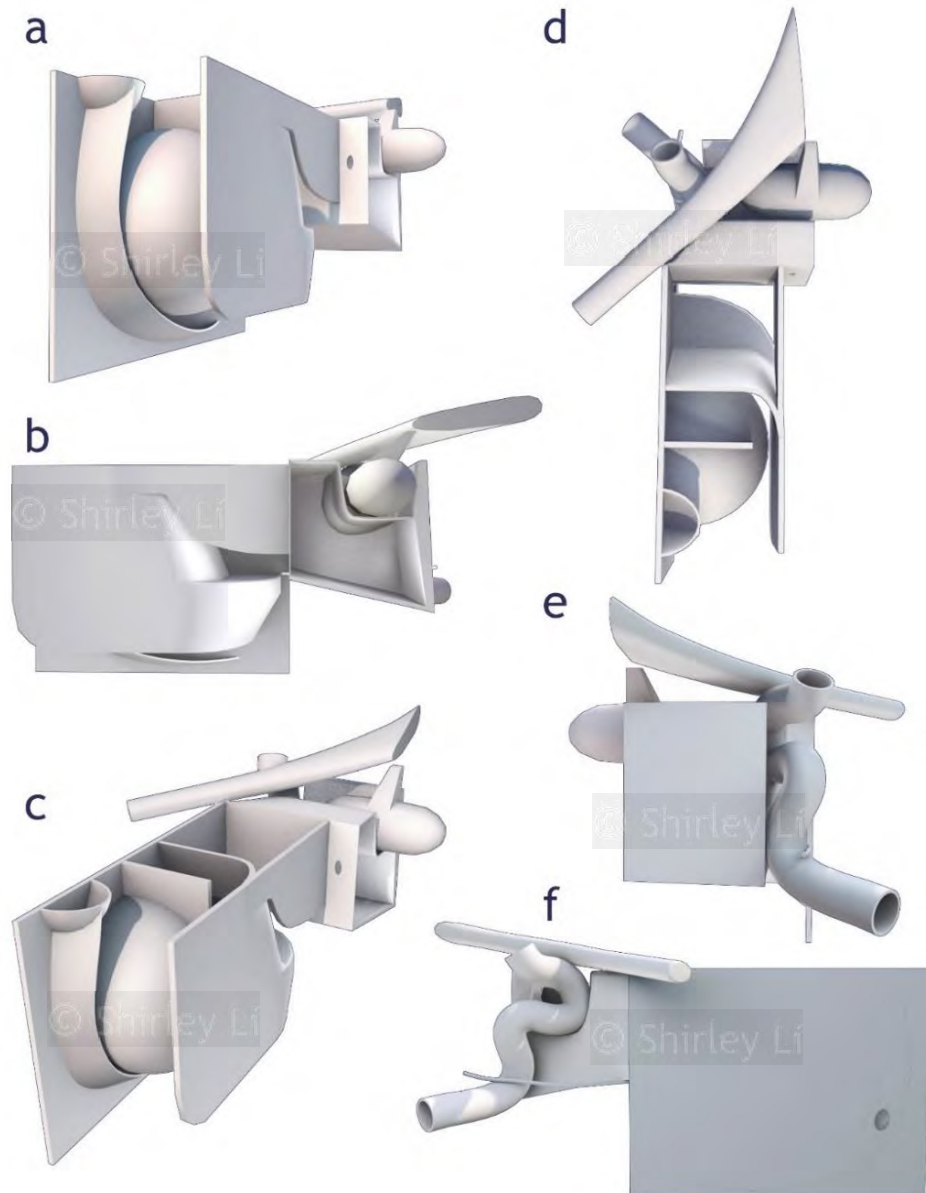


**Figure 33.** Screenshot of the “ModeControl” script and the Inspector window. Highlighted are changes occurring in GameObject variables upon switching to Clinical Mode.

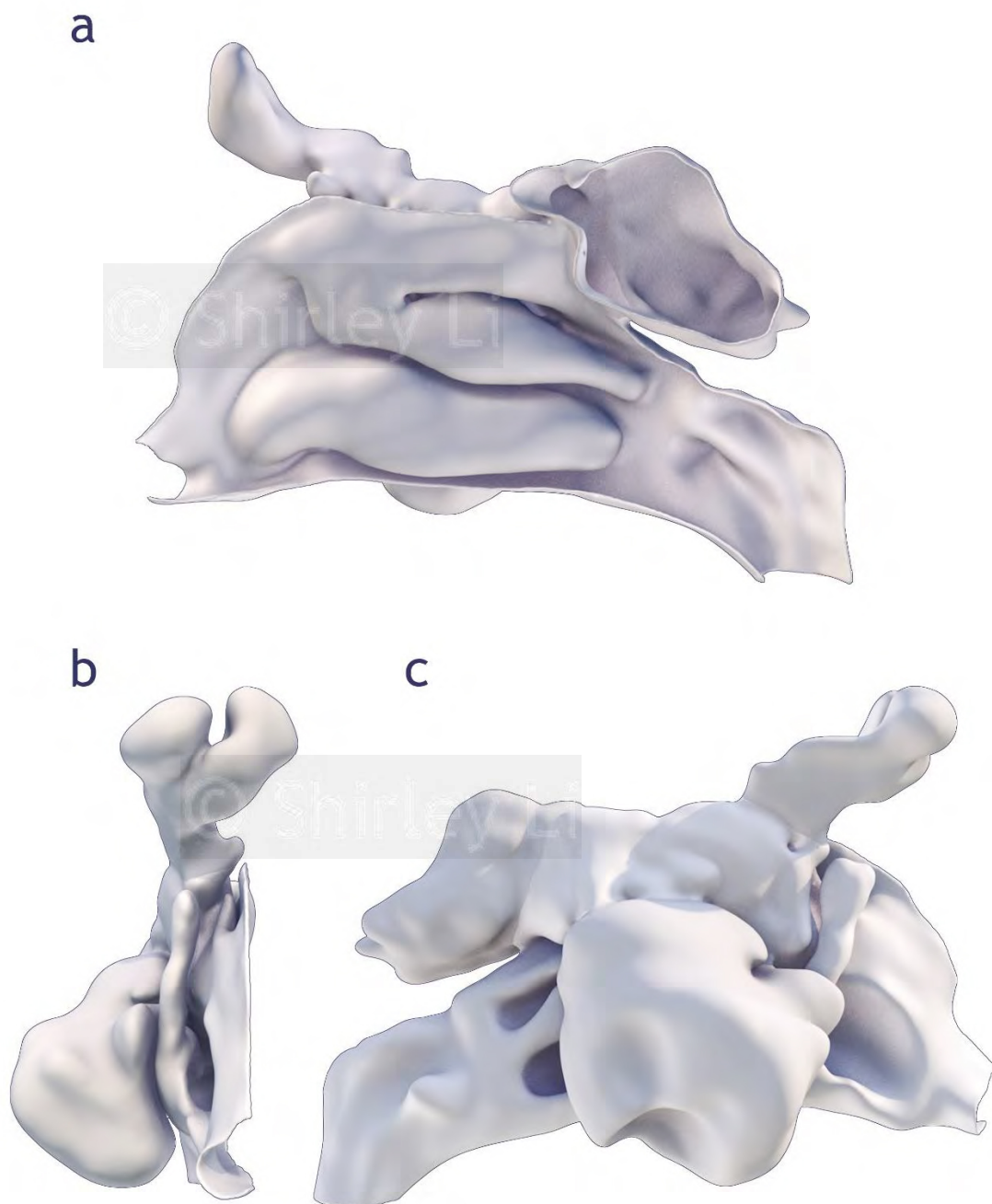
## Results

### *Stage 1: 3D Models*

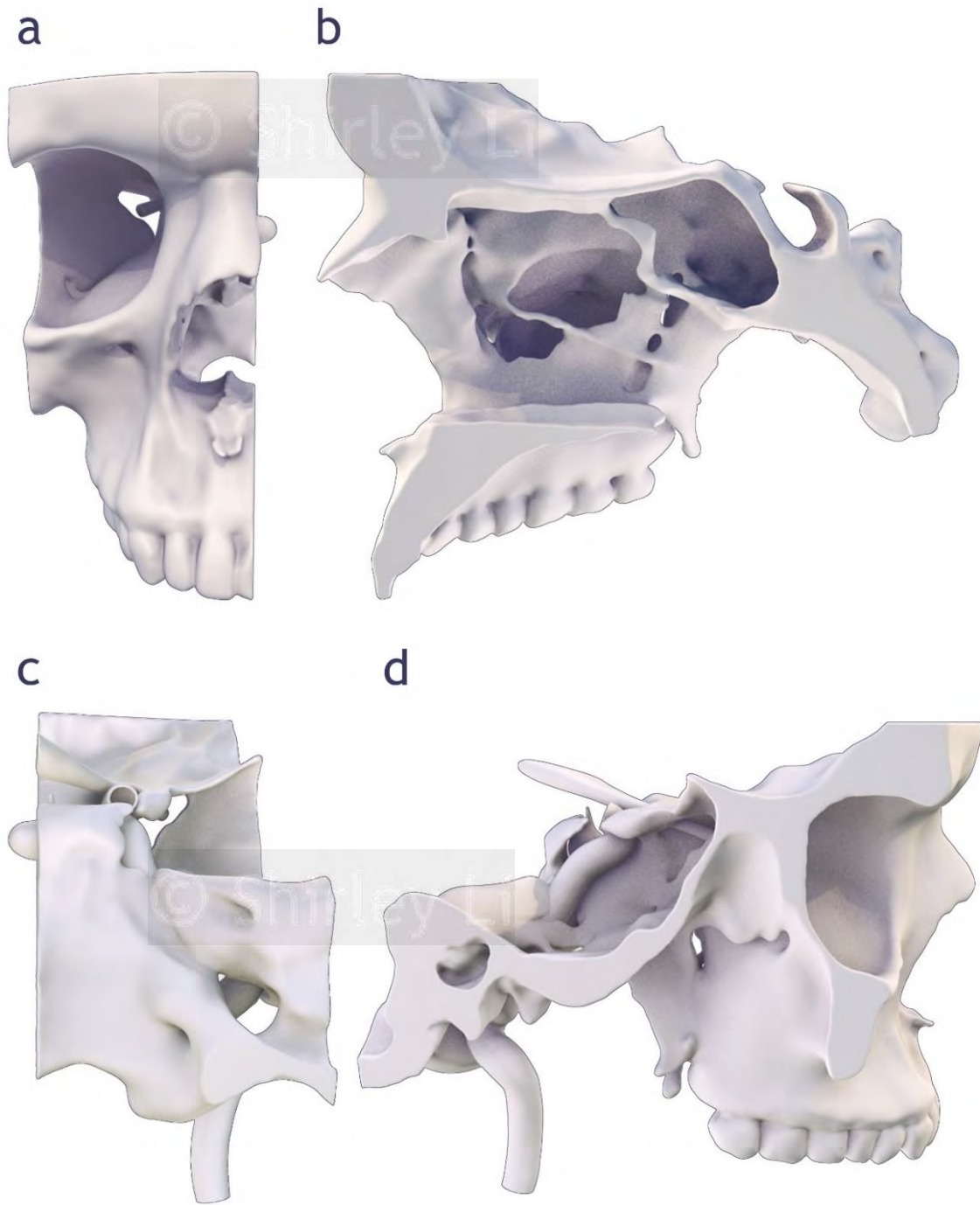
The final 3D schematic and CT-reconstructed models of the sinonasal cavity and related structures, as well as the 3D orienting cranium model are displayed in **Figure 34-38**.



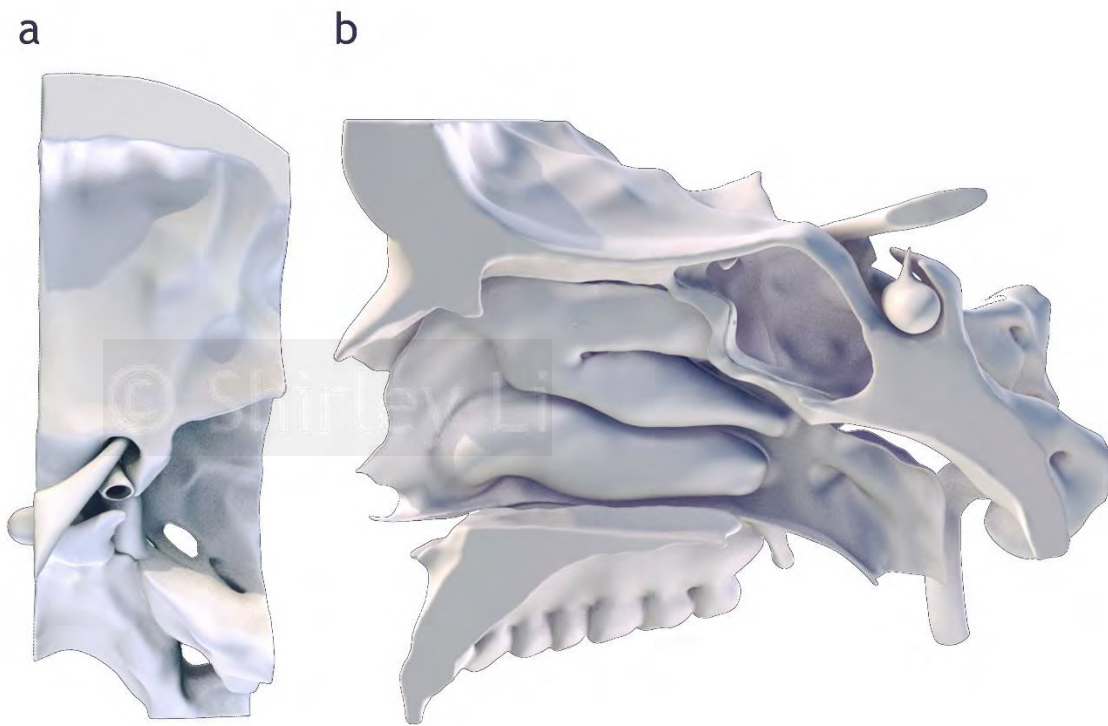
**Figure 34.** Rendered schematic model of the sinonasal cavity and skull base structures in (a) anteroinferior, (b) medial, (c) anterosuperior, (d) superior, (e) posterior, and (f) lateral views.



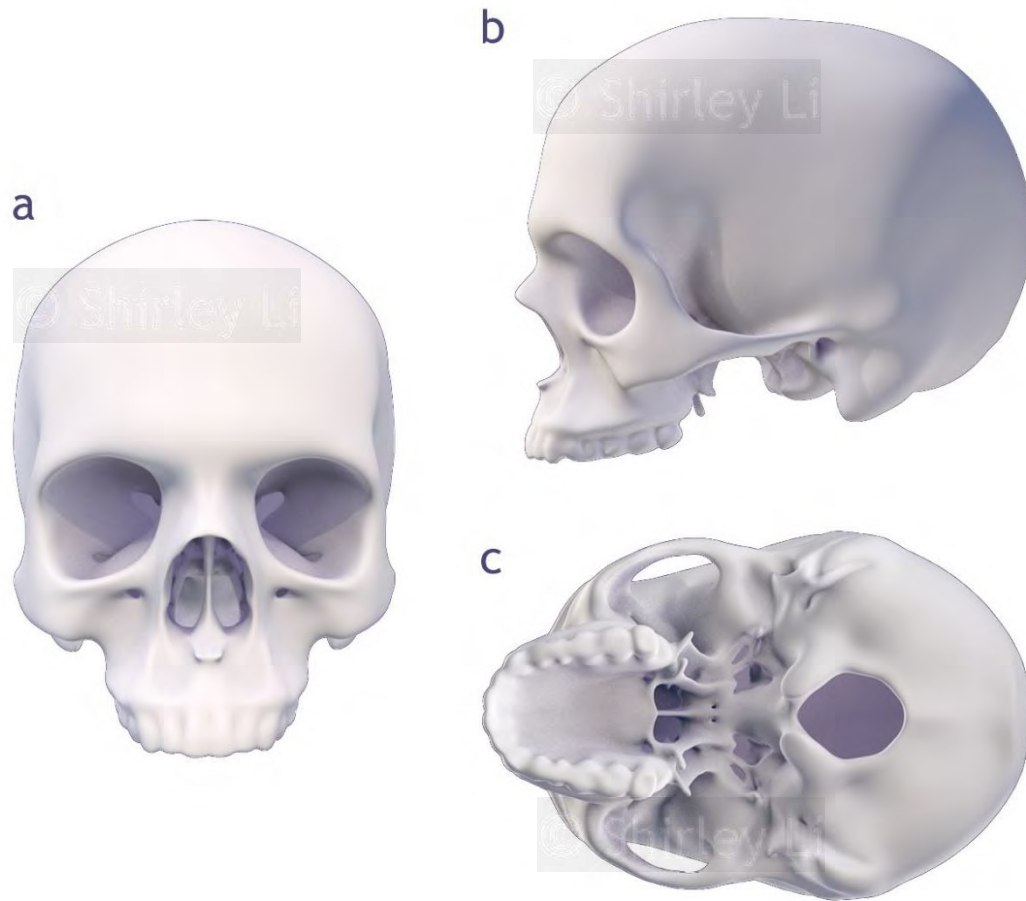
**Figure 35.** Rendered CT-reconstructed model of the sinonasal cavity in (a) medial, (b) anterior, and (c) lateral views.



**Figure 36.** Rendered CT-reconstructed model of the partial cranium and relevant skull base structures in (a) anterior, (b) medial, (c) posterior, and (d) lateral views.



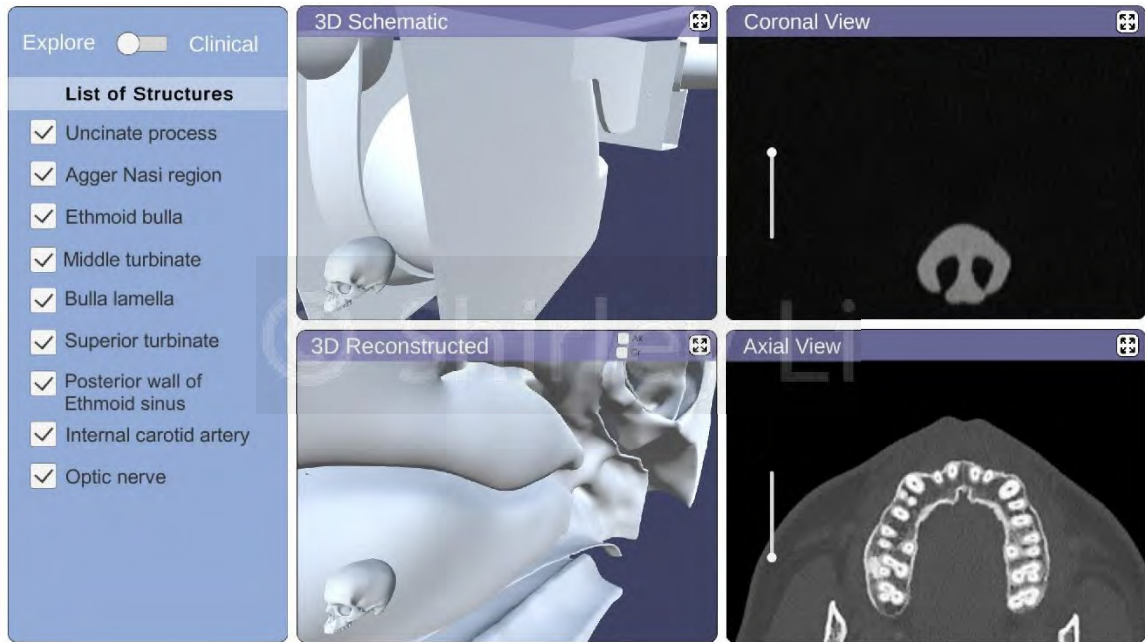
**Figure 37.** Combined cranium and sinonasal cavity models with relevant skull base structures in (a) superior, and (b) medial views.



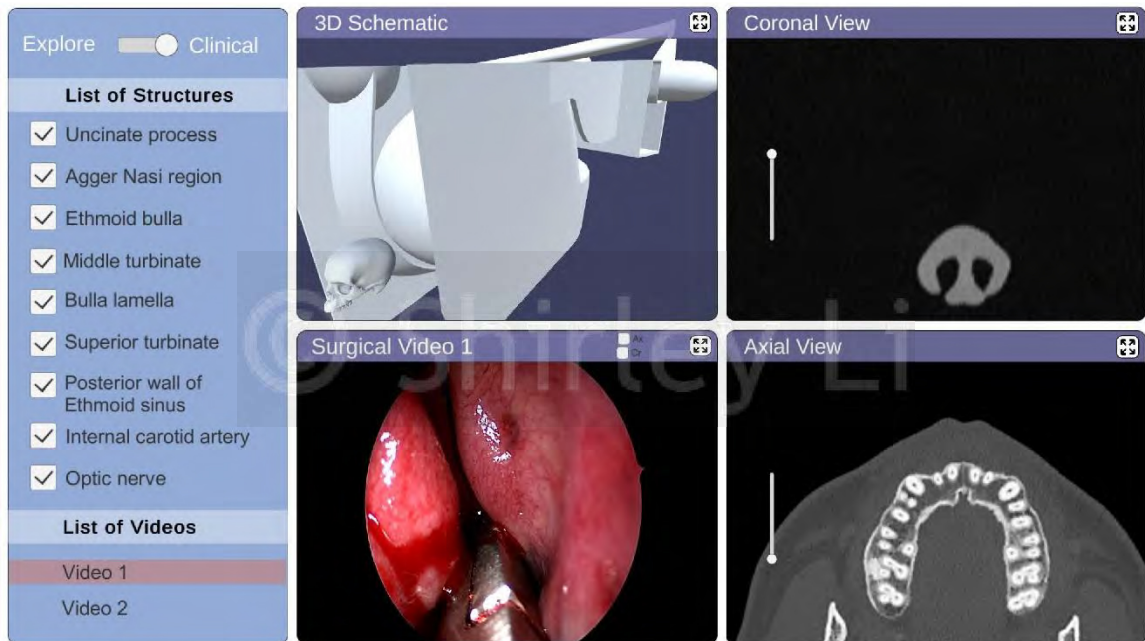
**Figure 38.** Rendered orienting cranium model in (a) anterior, (b) lateral, and (c) inferior views.

### *Stage 2: Interactive Elements*

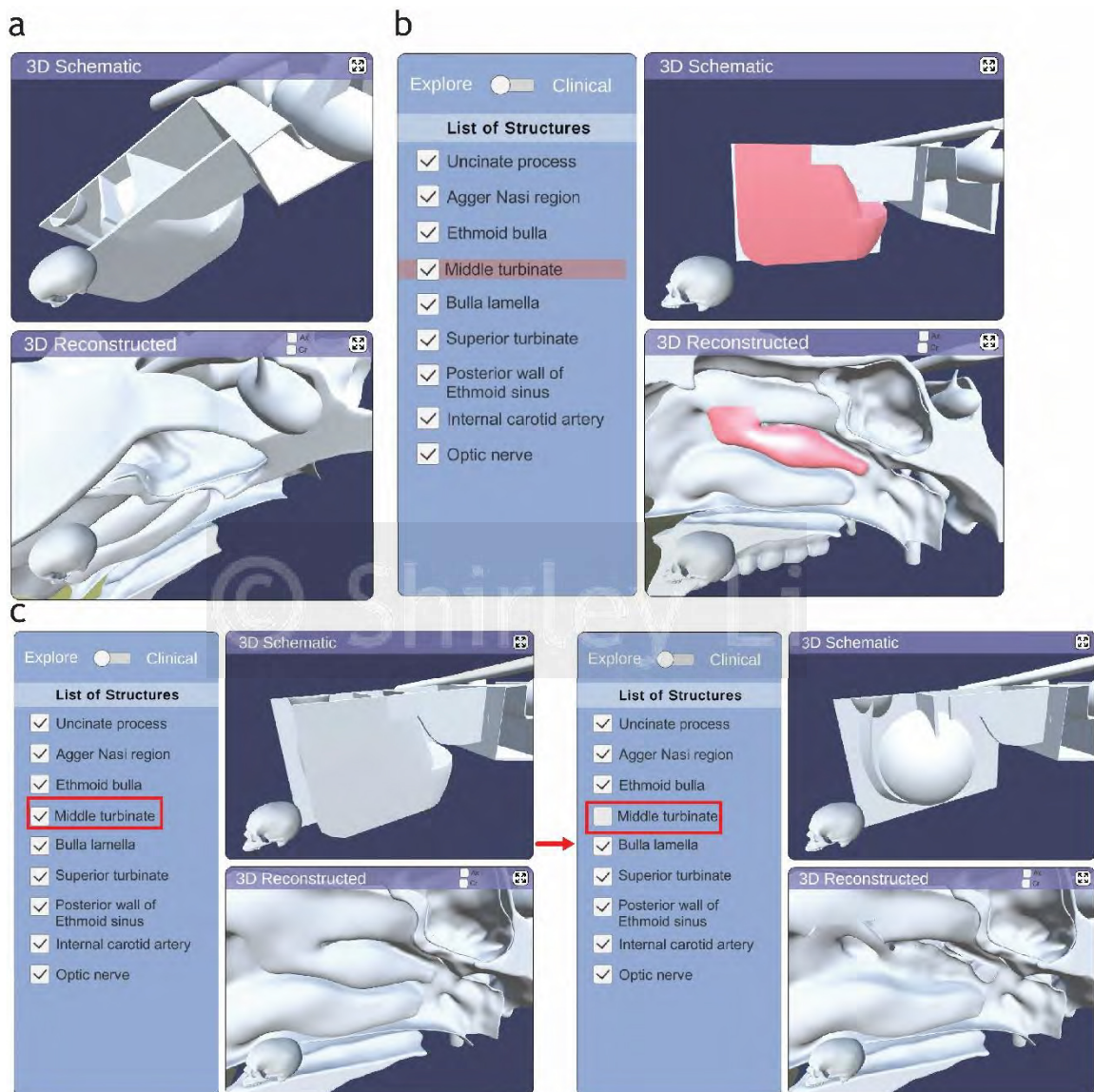
The user interface layout of this interactive resource, as well as developed interactions within Unity are displayed in **Figure 39-43**.



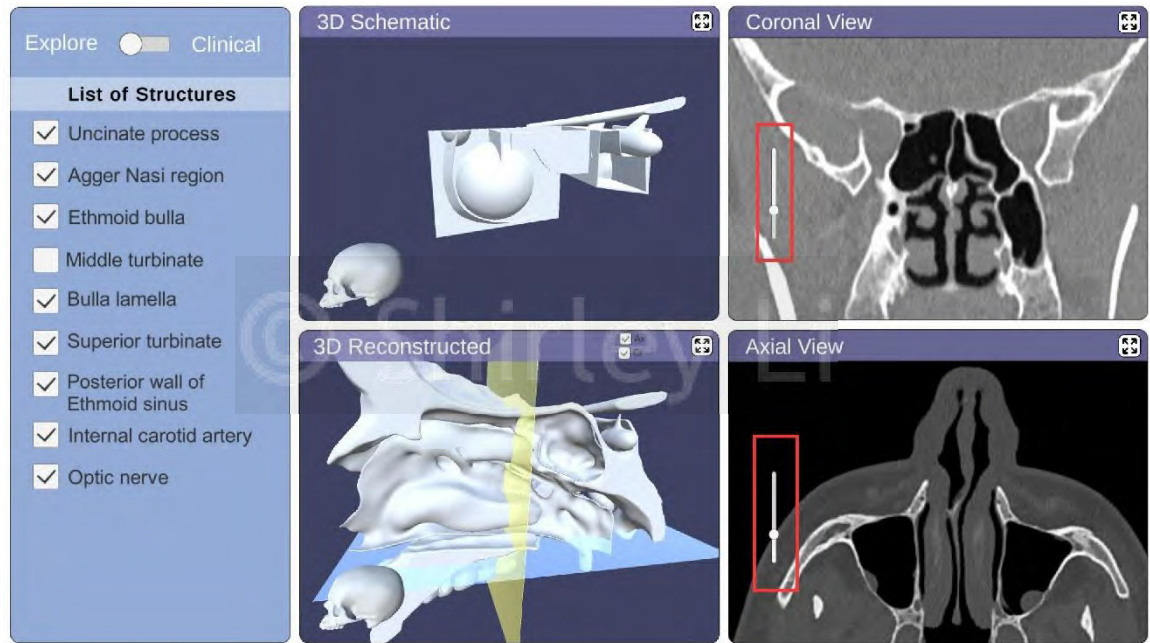
**Figure 39.** Screenshot of Game view within Unity showing user interface within Explore Mode.



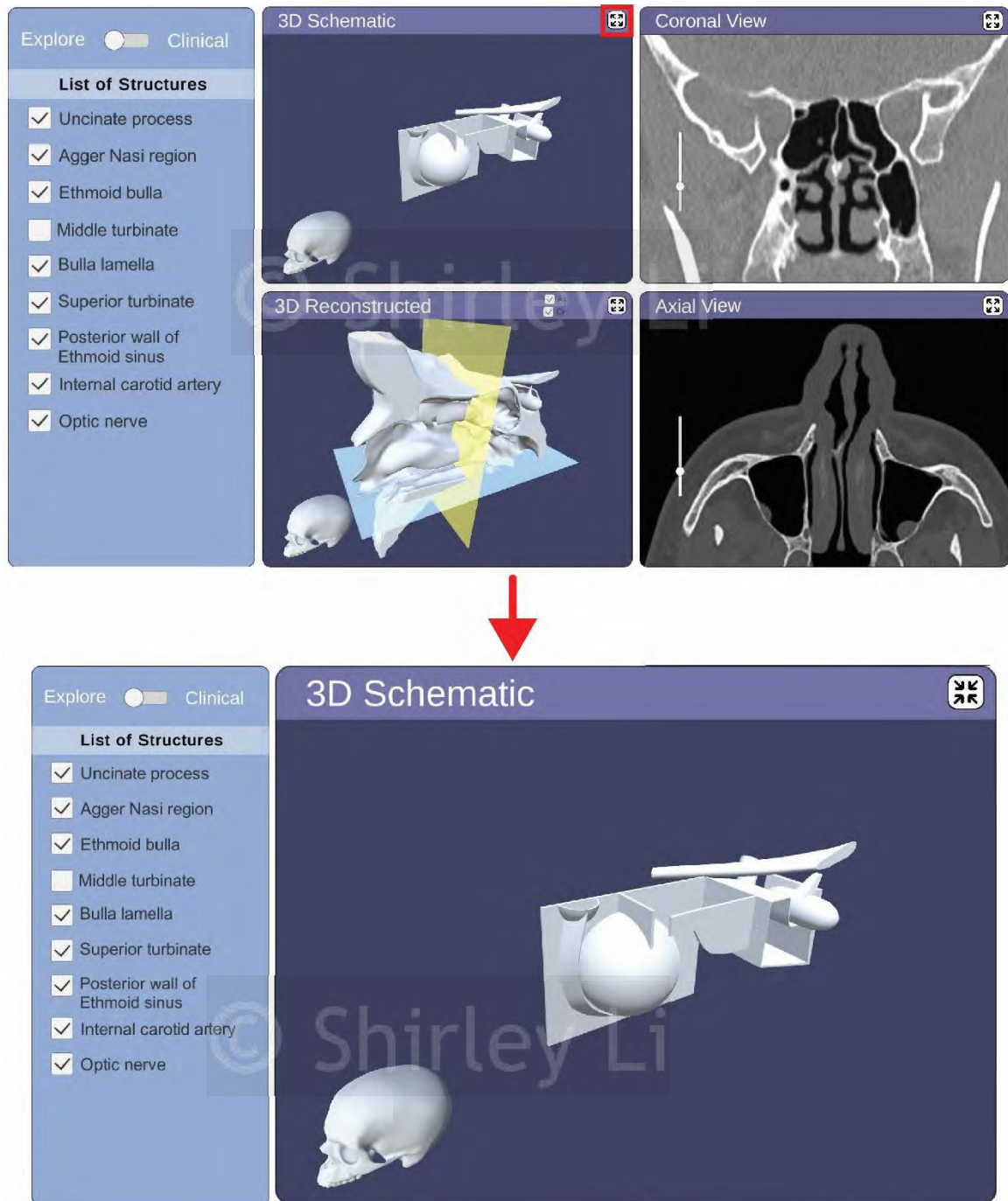
**Figure 40.** Screenshot of Game view within Unity showing user interface within Clinical Mode.



**Figure 41.** Screenshot of Game view within Unity showing interactions with 3D models, (a) synchronized manipulation of schematic and reconstructed models, (b) highlighting an anatomical structure in both models by selecting corresponding label Structures in the navigation panel, and (c) toggling off anatomical structure by unchecking corresponding checkbox in the navigation panel (as highlighted in red). Not all text intended to be read.



**Figure 42.** Screenshot of Game view within Unity showing activation of CT slice orientation planes upon slider value change in coronal CT image panel (as highlighted in red). Text not intended to be read.



**Figure 43.** Screenshot of Game view within Unity showing schematic panel put into full-screen mode upon clicking on the full screen button (as highlighted in red).  
Text not intended to be read.

*Access to Assets Resulting from this Thesis*

Access to the website for this interactive resource, 3D models, and script resulting from this thesis can be accessed in part at [www.shirleylistudio.com](http://www.shirleylistudio.com), or by contacting the author at [sli198@jhmi.edu](mailto:sli198@jhmi.edu). The author may also be reached through the Department of Art as Applied to Medicine via the website [www.hopkinsmedicine.org/medart](http://www.hopkinsmedicine.org/medart).

## Discussion

### *Project Goal*

The primary goal of this project was to create a web-based interactive learning resource featuring both 2D and 3D visualizations of the sinonasal cavity to help conceptualize this anatomical region and related structures and facilitate clinically relevant correlations across different media. In Explore Mode, users are able to explore schematic and CT-segmented 3D models of the sinonasal area while making side-by-side correlations to the corresponding CT radiological image series to help bridging the gap between 2D and 3D visualizations. In Clinical Mode, visualizations presented from an endoscopic perspective, using surgical video clips, along with a manipulable schematic 3D model with limited rotational range corresponding to the surgical view help users understand and familiarize themselves with the anatomy from a clinical perspective. Results of this project support the intended aim to improve surgical training outcomes among otolaryngology surgical trainees by reinforcing foundational anatomical knowledge and developing understanding of complex 3D spatial relationships in sinonasal region by using this didactic educational resource in adjunct with clinical exposure.

### *Considerations of Project Design*

#### *Order of Listed Anatomical Structures within Navigation Panel*

A thorough knowledge of embryology is crucial for understanding pathological processes possible in the sinonasal cavity (Sargi Z. B. & Casiano R. R., 2007). Anatomical structure labels provided in the List of Structures within the navigation panel were listed in order of embryonal development. Familiarity with developmental basis of these structures aids in

the analysis of radiologic and endoscopic pathological evaluation, as well as offers an opportunity for an improved understanding of morphological relationships, a vital component of safe, and technically excellent, surgical dissection. As such, we elected to organize developmentally related structures in proximity to one another in the List of Structures.

#### *Basis for and Presentation of Schematic Model with CT-Based Model*

Studies show that 3D schematic models allow acceleration of learning complex anatomical knowledge and help with comprehending difficult surgical concepts (Schmidt R. et al. 2021). The schematic model created for this project was adapted from a similar model published by Wise S. K. et al. (2012). Our simplified model improves conceptualization of the complex spatial relationships between clinically relevant anatomical structures compared to a more realistic model, as important anatomical landmarks are located deep in the cranium and are difficult to visualize in a realistic representation. However, pairing the schematic model with a realistic model derived from CT segmentation provides further context and serves as a clinically relevant reference for the schematic model.

#### *Choice of Materials and Colors for 3D Models*

A monochromatic white matte material was chosen as the default color seen for all structures of the two 3D models. Upon selecting a structure label among the List of Structures, the corresponding structure is highlighted using didactic colors to facilitate immediate identification of structures against non-selected structures shown in the default white color. The default white material serves to increase visual contrast between highlighted and unhighlighted structures. Furthermore, use of a monochromatic vs. chromatic schema paired with the limited color palette of the user interface allows users to

shift attention to brightly highlighted structures instantly across all panels, upon selecting a structure to explore.

### *Pairing of 3D Model with Video in Clinical Mode*

Within Clinical Mode, the 3D schematic model was chosen to be paired with surgical videos to provide a conceptual 3D roadmap to understanding special relationships between clinically relevant structures not seen from an endoscopic perspective. Without being able to retract structures in CT-reconstructed model, important clinical landmarks such as the uncinate process and bulla could not be seen, given the conscripted limited range of rotation dictated by camera angles seen in the video. The schematic 3D model provided a clearer representation of clinically relevant anatomy without imposing additional complexity, tissue, or structures seen in the CT reconstructed model.

### *Limitations*

Due to time limitations, the following pre-planned interactions were not able to completed in the scope of this research:

- Double-clicking on a structure in either of the 3D models to highlight the corresponding structure across all panels.
- Highlighting each structure across various slices in coronal and axial CT image arrays using the didactic color schema in Photoshop and developing coding within Unity, such that upon selecting a 3D structure or label, the sets of CT images with corresponding structure highlighted would be called up and displayed in the respective CT image panels.

- Adjusting further lighting used for 3D models within Unity.
- Limiting the range of spatial manipulation of the 3D schematic model in Clinical Mode.
- Incorporating pause and play controls within the surgical video panel.
- Labelling structures in surgical videos and enabling the video to jump to the frame that features the selected structure using coding in Unity.
- Improving the user interface through further designing of customized graphical assets for UI elements such as buttons used for full screen and panels.
- Building the website to host the resulting interactive resource of this project
- Reducing and optimizing the mesh of CT-reconstructed and orienting cranium 3D models further to improve game performance.

### *Project Innovations*

Despite limitations encountered due to time constraints, this project represents the first such use of didactic 3D models paired with clinical CT imaging and surgical video visualizations to help elucidate anatomical complexities of the sinonasal cavity via a self-directed interactive resource. With recent advances in technology and emergence of new visualization modalities, use of anatomical 3D models has become an increasingly popular in medical education. However, in the field of otolaryngology, existing educational resources using surgical videos and static CT images are unable to bridge the challenges of understanding 3D spatial knowledge of the sinonasal structures across the various media. For example, structures such as the infundibulum and hiatus semilunaris are challenging to conceptualize using sliced 2D visualization seen in CT. In the literature, the author found no effective teaching resources for visualizing these structures. However, by using our newly

created manipulable 3D model, the 3D channel of the infundibulum and the 2D air passage opening of the hiatus semilunaris can be easily conceptualized.

The pairing of patient-specific CT-derived 3D models with the corresponding 2D CT image series frequently used in clinical evaluation helps to promote correlation of 3D structures in space and 2D CT visualization. A study by Yao et al. (2014) indicates that combining 3D images and 2D CT scans of the sinus leads to positive outcomes in learning sinonasal anatomy. Therefore, the resulting product of this study featuring scrollable 2D CT slice data paired with interactive 3D visualizations of the sinonasal cavity serves as an innovative educational approach for teaching clinical trainees.

### *Future Directions*

Improvements in the educational aims of this research may occur through development of additional didactic content such as animations to visualize dynamic physiological processes of the sinonasal cavity (i.e., drainage pathways, embryological development, and pathological process). Inclusion of additionally relevant didactic content could further promote clinical correlation goals (i.e., upon selecting a structure, a callout panel explaining clinical importance and pitfalls can be activated). Such further development of this educational resource can raise awareness of additional considerations of clinical relevance amongst surgical trainees.

A logical next step for this project is to gather formal feedback from users and leverage the accessibility of a world-class otolaryngology training program to inform further adjustments to didactic content, functionality, and user interface design.

## **Conclusion**

The sinonasal cavity is a highly complex and morphologically variable anatomic region that is difficult to comprehend using traditional 2D visualization. This research addresses a lack of existing teaching resources of this anatomical region by creating a resource with the following elements that can be manipulated and explored in concert: 1) a manipulable 3D schematic model to conceptualize the spatial knowledge of the sinonasal cavity, 2) a CT-based 3D reconstructed model, 3) corresponding CT axial and coronal slice data, and 4) surgical videos. This unique interactive resource serves to reinforce foundation of anatomical knowledge, provide further 3D spatial understanding, and help bridge the gap between 3D, 2D, and clinical visualizations of sinonasal cavity as a means to improve clinical training outcomes for otolaryngology surgical trainees.

In this research, the workflow for developing such an interactive resource for clinical education incorporating various types of media is documented and provides the framework for a novel education tool with potential broad applicability. The resulting product offers the opportunity to reach a wide number of surgical trainees in need of improved educational methods for routine otolaryngology surgeries that carry the risks of potentially devastating anatomical-related complications (i.e., visual loss or stroke). This foundational work offers the opportunity to development of further educational resources in otolaryngology and neurosurgery, such as endoscopic skull base surgery, and other surgical subspecialties where a need exists to correlate information across a variety of media to improve learning.

## Appendices

### *Appendix A: “FullScreen” Script*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FullScreen : MonoBehaviour
{
    public Camera cam;
    public Camera uiCam;

    public bool fullScreen = false;
    public Rect startRect;

    // Start is called before the first frame update
    void Awake()
    {
        cam = GetComponent<Camera>();
        uiCam = transform.GetChild(0).GetComponent<Camera>();
        startRect = cam.rect;
    }

    public void ScrSize()
    {
        if(!fullScreen)
        {
            cam.rect = new Rect(.256f,.0098f,.74f,.982f);
            cam.depth = 15;
            uiCam.rect = new Rect(.256f,.0098f,.74f,.982f);
            uiCam.depth = 20;
            fullScreen = true;
        }
        else
        {
            cam.rect = startRect;
            cam.depth = 1;
            uiCam.rect = startRect;
            uiCam.depth = 10;
            fullScreen = false;
        }
    }
}
```

## *Appendix B: "StructureControl2" Script for Concerted Manipulation of Models*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StructureControl2 : MonoBehaviour
{
    public float rotateSpeed = 10;
    public bool rotating = false;

    public float panSpeed = 100;
    public bool panning = false;

    public Camera controlCamera;
    public bool controlling = false;

    public GameObject otherObj;

    public GameObject skull;
    public GameObject skull2;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(0) || Input.GetMouseButtonDown(1)) //right click OR
left click
        {
            Ray ray = controlCamera.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;
            if (Physics.Raycast(ray, out hit) && hit.transform.tag == "Structure") //if raycast
happens (always true) and hit the transfm (boundary) of obj w/ tag "structure", then allow
ctrl
            {
                controlling = true;
            }
        }
        if (Input.GetMouseButtonUp(0) || Input.GetMouseButtonUp(1))
        {
            controlling = false;
            rotating = false;
            panning = false;
        }

        if (controlling)
        {
            //rotation
            if (Input.GetMouseButtonDown(0)) //Left Click
```

```

    {
        Vector3 view = controlCamera.ScreenToViewportPoint(Input.mousePosition);
        if (view.x >= 0 && view.x <= 1 && view.y >= 0 && view.y <= 1)
        {
            rotating = true;
        }
        //rotating = true;
    }
    if (rotating)
    {
        transform.Rotate((Input.GetAxis("Mouse Y") * rotateSpeed * Time.deltaTime),
        (Input.GetAxis("Mouse X") * -rotateSpeed * Time.deltaTime), 0, Space.World);
        otherObj.transform.Rotate((Input.GetAxis("Mouse Y") * rotateSpeed *
        Time.deltaTime), (Input.GetAxis("Mouse X") * -rotateSpeed * Time.deltaTime), 0,
        Space.World);

        skull.transform.Rotate((Input.GetAxis("Mouse Y") * rotateSpeed *
        Time.deltaTime), (Input.GetAxis("Mouse X") * -rotateSpeed * Time.deltaTime), 0,
        Space.World);
        skull2.transform.Rotate((Input.GetAxis("Mouse Y") * rotateSpeed *
        Time.deltaTime), (Input.GetAxis("Mouse X") * -rotateSpeed * Time.deltaTime), 0,
        Space.World);
    }

    //panning
    if (Input.GetMouseButtonDown(1)) //Right Click
    {
        Vector3 view = controlCamera.ScreenToViewportPoint(Input.mousePosition);
        if (view.x >= 0 && view.x <= 1 && view.y >= 0 && view.y <= 1)
        {
            panning = true;
        }
        //panning = true;
    }
    if (panning)
    {
        transform.Translate((Input.GetAxis("Mouse X") * panSpeed * Time.deltaTime),
        (Input.GetAxis("Mouse Y") * panSpeed * Time.deltaTime), 0, Space.World);
        otherObj.transform.Translate((Input.GetAxis("Mouse X") * panSpeed *
        Time.deltaTime), (Input.GetAxis("Mouse Y") * panSpeed * Time.deltaTime), 0,
        Space.World);
    }
}

//zooming
}
}

```

### *Appendix C: “CameraZoom” Script for Zooming Action*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraZoom : MonoBehaviour
{
    public float zoomSpeed = 100f;
    public float minZoom;
    public float maxZoom;

    public Camera zoomCamera;
    public GameObject otherCam;

    //new
    public Vector3 otherPos;

    void Update()
    {
        Zoom();
    }

    void Zoom()
    {
        Vector3 view = zoomCamera.ScreenToViewportPoint(Input.mousePosition);
        if (view.x >= 0 && view.x <= 1 && view.y >= 0 && view.y <= 1)
        {
            Ray ray = zoomCamera.ScreenPointToRay(Input.mousePosition);
            Vector3 scrollDirection = ray.GetPoint(5);

            if(transform.position.y > otherCam.transform.position.y)
            {
                otherPos = new Vector3(scrollDirection.x, scrollDirection.y - 500,
scrollDirection.z);
            }
            else
            {
                otherPos = new Vector3(scrollDirection.x, scrollDirection.y + 500,
scrollDirection.z);
            }

            float step = zoomSpeed * Time.deltaTime;

            Ray ray2 = zoomCamera.ScreenPointToRay(Input.mousePosition);
            if (Input.GetAxis("Mouse ScrollWheel") > 0)
            {
```

```

        transform.position = Vector3.MoveTowards(transform.position, scrollDirection,
Input.GetAxis("Mouse ScrollWheel") * step);
        otherCam.transform.position =
Vector3.MoveTowards(otherCam.transform.position, otherPos, Input.GetAxis("Mouse
ScrollWheel") * step);
    }
    if (Input.GetAxis("Mouse ScrollWheel") < 0)
    {
        transform.position = Vector3.MoveTowards(transform.position, scrollDirection,
Input.GetAxis("Mouse ScrollWheel") * step);
        otherCam.transform.position =
Vector3.MoveTowards(otherCam.transform.position, otherPos, Input.GetAxis("Mouse
ScrollWheel") * step);
    }
}
}
}

```

#### *Appendix D: “CoronalCT” Scrollable Image Series Script*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class CoronalCT : MonoBehaviour
{
    public Sprite[] images;
    public Slider slider;

    public int arrayIndex = 0;
    public int oldValue = 0;

    public GameObject coronalPlane;
    public Vector3 startPos;
    public float minPos = -0.59f;
    public float maxPos = 0.22f;

    public float totalDist;
    public float interval;

    // Start is called before the first frame update
    void Start()
    {
        slider = GameObject.Find("Coronal Slider").GetComponent<Slider>();
        coronalPlane = GameObject.Find("CoronalPlane"); //Moving plane in 3D model panel

        totalDist = maxPos - minPos;
        interval = totalDist/(slider.maxValue);

        startPos = coronalPlane.transform.localPosition;
    }

    public void ChangeImage()
    {
        arrayIndex = (int)slider.value;
        gameObject.GetComponent<Image>().sprite = images[arrayIndex];

        coronalPlane.transform.localPosition = new Vector3(startPos.x, startPos.y +
interval*arrayIndex, startPos.z);
    }
}
```

### *Appendix E: “Toggle” Script for Visibility Control of Structures*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Toggle : MonoBehaviour
{
    public bool visible = true;

    public Renderer rend;

    void Awake ()
    {
        rend = GetComponent<Renderer>(); //grab renderer--don't need to assign manually
    }

    public void Visible()
    {
        if (visible)
        {
            rend.enabled = false;
            visible = false;
        }
        else
        {
            rend.enabled = true;
            visible = true;
        }
    }
}
```

## *Appendix F: “Highlighting” Script*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Highlighting : MonoBehaviour
{
    public Material highlightMat;
    public Material standardMat;

    public GameObject highlightObj;
    public GameObject lastHighlightObj;

    // Start is called before the first frame update
    void Awake()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    public void Highlight(string structureName)
    {
        highlightObj = GameObject.Find(structureName);
        highlightObj.GetComponent<Renderer>().material = highlightMat;

        if (lastHighlightObj != null)
        {
            lastHighlightObj.GetComponent<Renderer>().material = standardMat;
        }

        lastHighlightObj = highlightObj;
    }
}
```

### *Appendix G: “ModeControl” Mode Switching Script*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class ModeControl : MonoBehaviour
{
    public bool clinical = false;

    public TextMeshProUGUI bottomLeftText;

    public GameObject video;

    public GameObject videosList;

    public GameObject skull2;

    public GameObject reconstModel;

    void Awake()
    {
        video = GameObject.Find("Video");
        videosList = GameObject.Find("vid list");

        video.SetActive(false);
        videosList.SetActive(false);
        skull2.SetActive(true);
        reconstModel.SetActive(true);
    }

    public void Clinical ()
    {
        if (!clinical)
        {
            bottomLeftText.text = "Surgical Video";

            videosList.SetActive(true);
            video.SetActive(true);
            skull2.SetActive(false);
            reconstModel.SetActive(false);

            clinical = true;
        }
        else
    }
```

```
{
    bottomLeftText.text = "3D Reconstructed";

    videosList.SetActive(false);
    video.SetActive(false);
    skull2.SetActive(true);
    reconstModel.SetActive(true);

    clinical = false;
}
}
```

## References

- Baily C. E., Grauer J. S., Chen P. G., et al. (2021) Development of a Self-directed Sinonasal Surgical Anatomy Video Curriculum: Phase 1 Validation. *Int Forum Allergy Rhinol.* DOI 10.1002/alr.22814
- Cardesa A., Also L., Nadal A., & Franchi A. (2017) Nasal Cavity and Paranasal Sinuses. *Pathology of the Head and Neck*, DOI: 10.1007/978-3-662-49672-5\_2
- Chung S., Yoon S., & Dhong H. (1999) A Comparison between Coronal and Axial Scans in PNS CT. *J Rhinol* 6 (2): 131-135
- Dalgorf D, M., & Harvey R. J. (2013). Sinonasal Anatomy and Function. *Am J Rhinol Allergy*, 27, S3-S6.
- Georgakopoulos B., Hohman M. H., & Le P. H. (2022) Anatomy, Head and Neck, Nasal Cavity. *StatPearls*, <https://www.ncbi.nlm.nih.gov/books/NBK544232/>
- Koch G. K., Sethi R. K. V., Kozin E. D., et al. (2017) Online Teaching Tool for Sinus Surgery: Trends toward Mobile and Global Education. *American Academy of Otolaryngology—Head and Neck Surgery*. DOI 10.1177/2473974X17729812
- Lee A. S., Solyar A. Y., Lanza D. C., & Georgalas C. (2013) Nasal and Paranasal Sinus Anatomy and Embryology. *Rhinology and Skull Base Surgery*. DOI: 10.1055/b-0034-77977
- Loevner L. A. & Mikityansky I. (2012) Radiologic Imaging of the Paranasal Sinuses and Skull Base. *Rhinology: Diseases of the Nose, Sinuses, and Skull Base*. Thieme Medical Publishers. ISBN 978-1-60406-060-7
- Papadopoulou A., Chrysikos D., Samolis A., et al. (2021) Anatomical Variations of the Nasal Cavities and Paranasal Sinuses: A Systemic Review. *Cureus* 13(1): e12727. DOI: 10.7759/cureus.12727
- Rapp A. K., Healy M. G., Charlton M. E., et al. (2016) YouTube is the Most Frequently Used Educational Video Source for Surgical Preparation. *J. Surg. Educ.* 73: 1072-1076
- Sargi Z. B. & Casiano R. R. (2007) *Surgical Anatomy of the Paranasal Sinuses. Thinologic and Sleep Apnea Surgical Techniques*. Springer. ISBN 978-3-540-34019-5
- Schmidt R., Gartrell R., Yeung J. M. et al. (2021) A Pipeline for Generating Interactive, Schematic 3D Surgical Anatomy Models. *Journal of Surgical Education* 78 (5): 1419-1424
- Sobiesk J. L. & Munakomi S. (2022). Anatomy, Head and Neck, Nasal Cavity. *StatPearls*, <https://www.ncbi.nlm.nih.gov/books/NBK544232/>

- Stull A. T., Hegarty M., & Mayer R. E. (2009) Getting a Handle on Learning Anatomy with Interactive Three-dimensional Graphics. *J Educ Psychol* 101: 803-816  
<https://doi.org/10.1037/a0016849>
- Wise S. K., Orlandi R. R., & DelGarudio J. M. (2012) *Sinonasal Development and Anatomy. Rhinology: Diseases of the Nose, Sinuses, and Skull Base*. Thieme Medical Publishers. ISBN 978-1-60406-060-7
- Yao W. C., Regone R. M., Huyhn N. et al. (2014) Three-Dimensional Sinus Imaging as an Adjunct to Two-Dimensional Imagine to Accelerate Education and Improve Spatial Orientation. *The Laryngoscope* 124: 596-601

## **Vita**

Shirley Li was born in China and raised in Vancouver, Canada. She was obsessed with biology in high school and decided to pursue a degree in medicine. In 2017, she received a BSc in Comparative Physiology at University of Toronto, Mississauga (UTM). Without any exposure to visual art as she grew up, Shirley's creative side was awoken in the third year of her undergraduate studies when she used doodling skills to help understand and memorize complex scientific concepts. Upon realizing that art could be such a powerful tool in communicating science and learning about the field of medical illustration from the Biomedical Communication department at UTM, she decided to pursue studying medical illustration.

Following graduation, Shirley spent two years refining her art skills via formal art training before her matriculation into the Johns Hopkins University graduate program in Medical and Biological Illustration at the Department of Art as Applied to Medicine. During Shirley's first year, she received an Award of Merit for her anatomical plate "Bony Landmarks of the Eustachian Tube" in the Student category of the 2021 Annual Association of Medical Illustrators (AMI) Conference. She was also honored to be selected as a 2022 Vesalian Scholar recipient granted by the Vesalius Trust for this thesis proposal. Shirley is currently a candidate to receive a Master of Arts in Medical and Biological Illustration in May 2022.

Following graduation, Shirley hopes to keep growing as an artist, and take her skills and experiences to facilitate audience interest in science and promote more effective communication among STEM professions using 2D and 3D animation and interactive media.