

|  |
| --- |
|  |
| CG Final Project |
| 03/01/2022 |
|  |

Table of Contents

[03/01/2022 1](#_Toc92052512)

[Team memberes info 3](#_Toc92052513)

[implementation details 4](#_Toc92052514)

[Implemeted functions from task statement: 4](#_Toc92052515)

[implementation methodology: 4](#_Toc92052516)

[Sample results of the code 5](#_Toc92052517)

[Issues that faced us 6](#_Toc92052518)

[Contribution of other students 7](#_Toc92052519)

# Team memberes info

|  |  |  |
| --- | --- | --- |
| Name | Section | BN |
| Aya Shawky Hassan | 1 | 17 |
| Habiba Ahmed | 1 | 23 |
| Nada Abdelrahman Mohamed | 2 | 38 |
| Hager Ashraf Helal | 2 | 41 |
| Haidy Sayed Hassan | 2 | 42 |

# implementation details

### Implemeted functions from task statement:

* Build volume rendering web app with VTK.js & HTML
* Use datasets provided in vtk examples (head for surface rendering and chest for ray casting)
* Features:
* Support loading DICOM series dynamically using load button
* Surface rendering with adjustable iso value (try sliders)
* Add interactive widget to cut the volume in the three perpendicular planes

### implementation methodology:

At first, we saw the slides of the vtk.js and followed the instructions step by step by implementing the cone at first to see how it works and how the code is written what every line means and do, and then moving forward with the skull and the chest example,

Then we followed the instructions in the react section in vtk website to start implementing each of the skull and the chest with react how it works and how to link our data in it, how to make buttons, how to set flags and how to call functions,

Then after implementing each of them with react we moved forward to know how to link between them we watched lots of videos to implement this way as there were not enough sources in vtk.js website for react,

After that we made a starter page so that when we run the code the localhost opens to it and then you got to choose what to see the chest or the head example,

Each example of these two has its features from iso value to cutting and all the features that we were asked for,

Then when the user choose what to see the page will show it and hide the other and so on.

# Sample results of the code

1. First at running the code

A picture containing text, person, holding

Description automatically generated

1. After pushing the chest button

Graphical user interface, diagram

Description automatically generated

1. After using cutting feature of the chest

A screenshot of a computer

Description automatically generated with medium confidence

1. After disabling the visibility feature

A computer screen capture

Description automatically generated with medium confidence

1. After pushing the skull button

A picture containing text

Description automatically generated

1. After changing the iso value of the skull

A picture containing text, indoor, screen, staring

Description automatically generated

# Issues that faced us

1. The first issue faced us that the way to import data in the vtk.js slides is old not updated to the latest version so we searched and find the solution at <https://kitware.github.io/vtk-js/docs/intro_vtk_as_es6_dependency.html#Migrating-from-vtk-js-to-kitware-vtk-js> to remove each @http.vtk and replace it with @kitwarevtk
2. Also wee faced a problem at trying both the skull and the chest from the examples from the first time without even adding react as the way the website link the url of the data wasn’t working right so we remove the word base url and replace it with the required data file we wanted to download
3. With adding react to both of the examples we faced some problems to modify within but after searching well about how react works we managed to do it well
4. We faced a problem between linking the buttons in both of the chest and skull examples to return to the other but also managed to solve it though it took lots of time
5. The chest with its features took us lots of time too to implement it right

# Contribution of other students

* Some of the teams helped us when we faced the problem of applying both examples using react as our TA asked us to ask for their help, they sent us some links to help us solve the errors we faced
* Also we discussed how to implement the whole task like brainstorming with our friends to show the best results
* Each time we face an unsolvable error and search for it for along time with no vain we send it to the group and immediately one of our colleagues tell us where the problem is and how to solve it.