Edurell

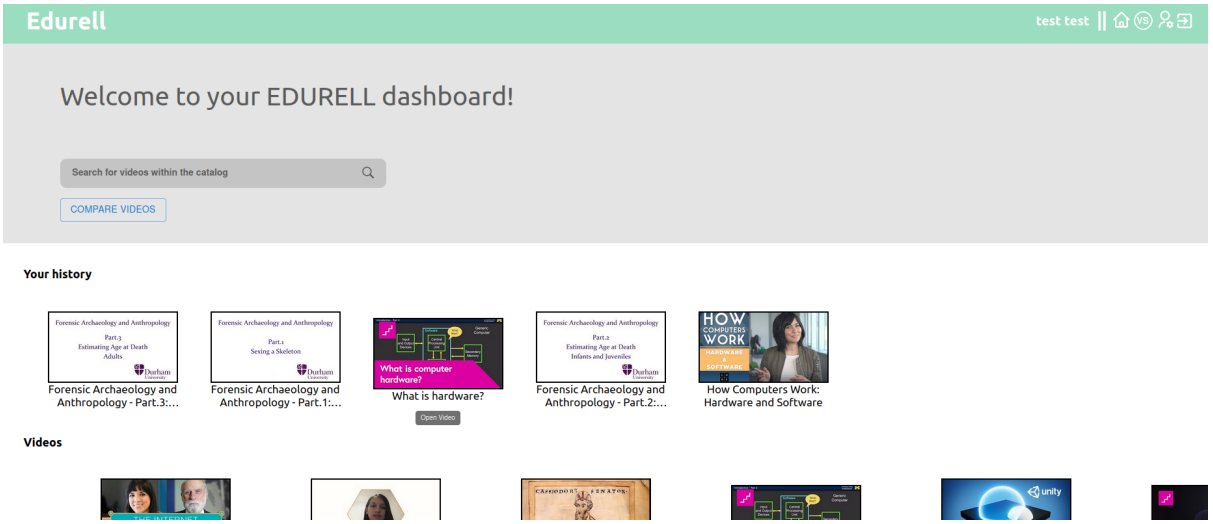
Video Comparison Document

Content Overview

1.1 Interface Functionality

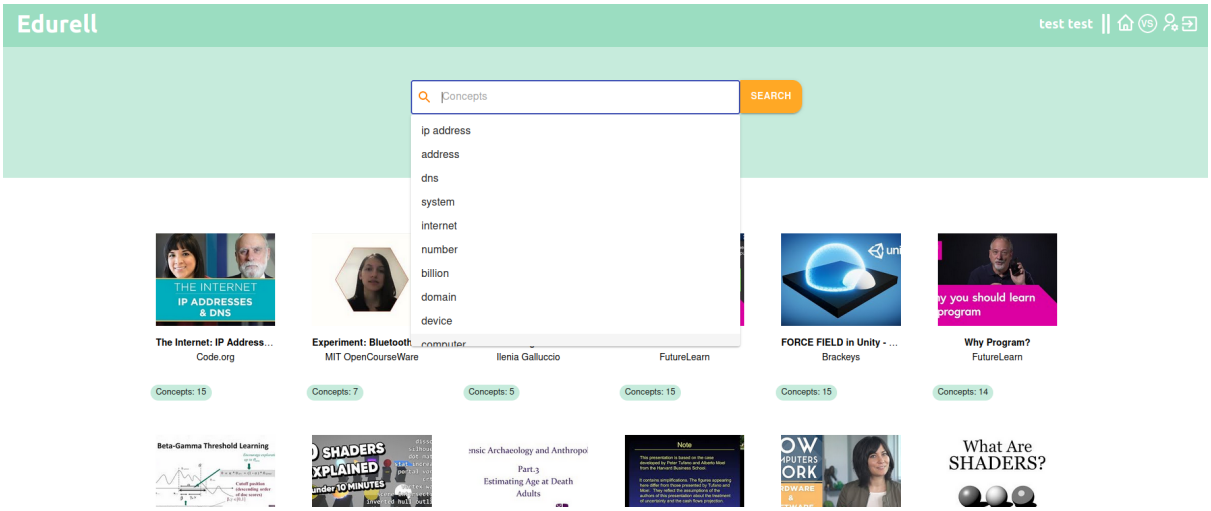
A simple explanation of how the video comparison feature works.

1.2 How to access it



After logged in, to access the part for Compare Videos, a blue button is present in the Dashboard page. Otherwise, in the navbar, an icon with VS, permits to access it too. Useful when the user is in another page and want to access the Comparison page without go again in the Dashboard.

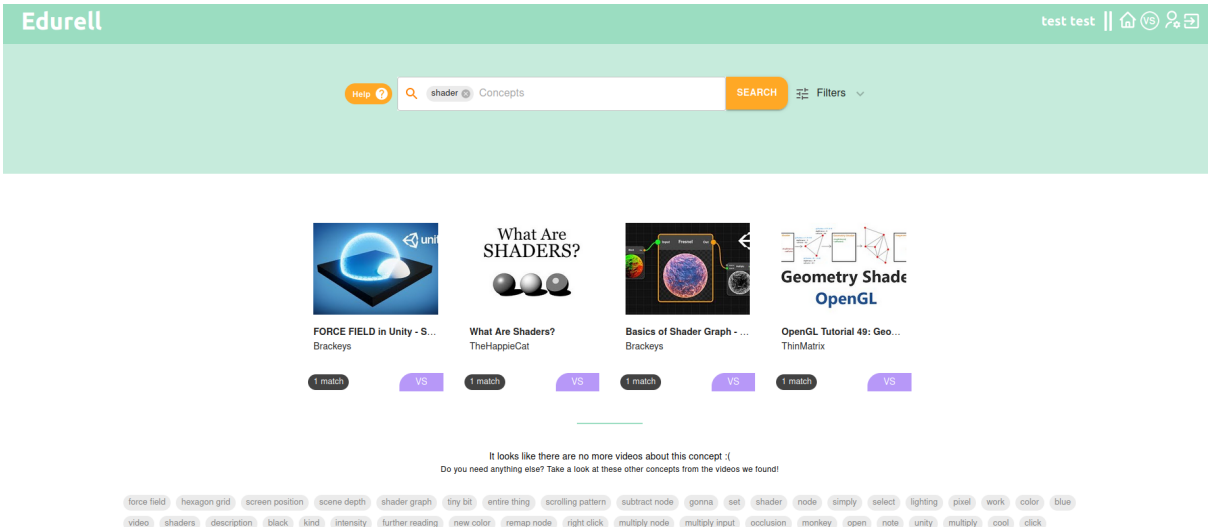
1.3 First page - Dashboard of Comparison



After entered the Comparison page, you will have a list of all videos available on the server wiith a value of learnable concept present in each video. Click any video here to watch it!

It's possible to filter the videos by selecting the concept you are interested to learn using the searchbar

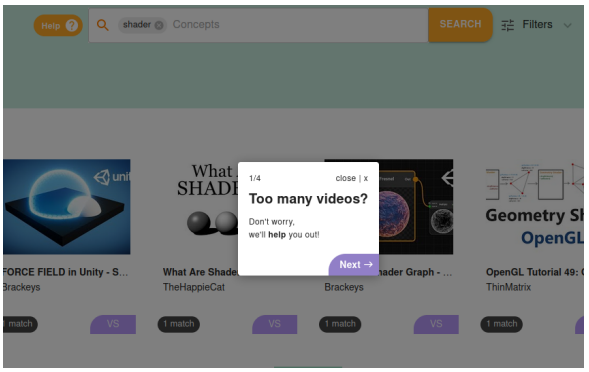
1.4 Second page - Choice of the videos



After selecting a concept the videos will be filtered to match the selected concept. Notice that the page is a little different:

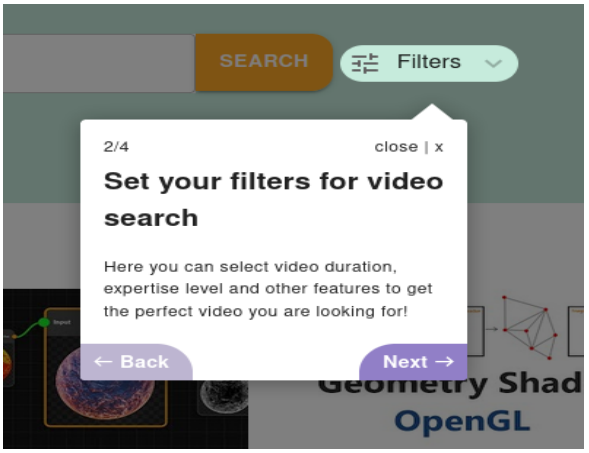
1. A help button
2. Filters button
3. Video panel is different
4. buttons on the end of the page

1.4.1 Help button



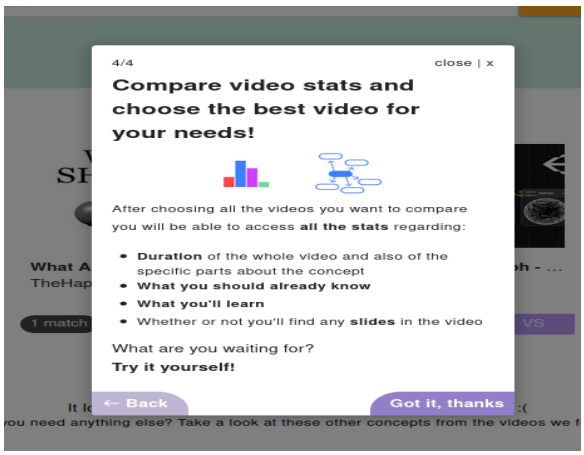
If it's your first time here, the tutorial window will pop up automatically, otherwise, to see again the tutorial, you have to click the button help.

In total there is 4 pages. The first page is set at the center of the page. Only Next button and the close button is available



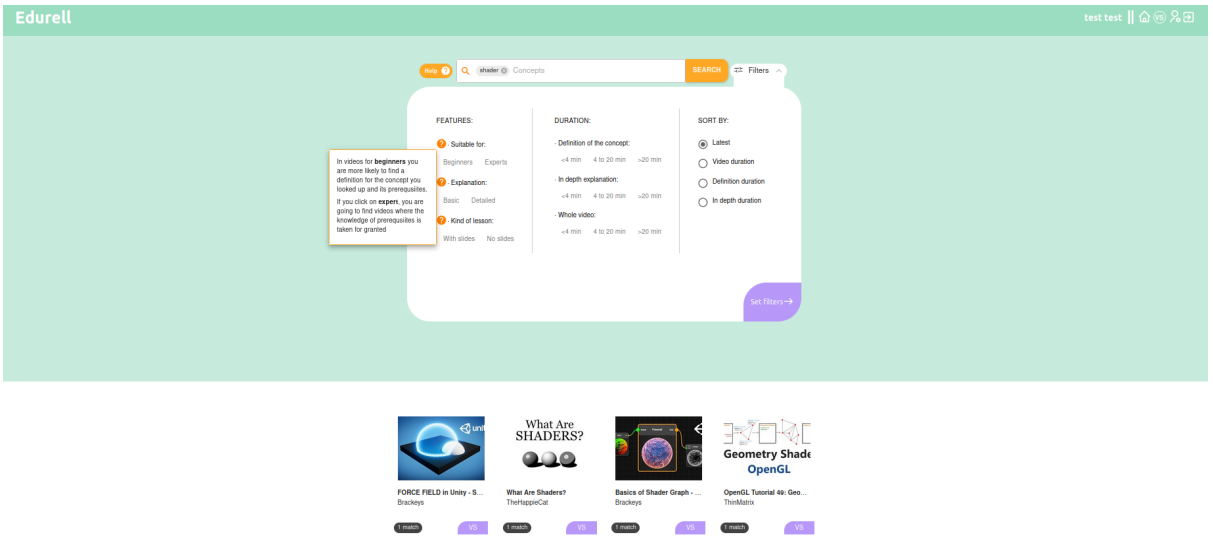
The second page will show the Back button too and the window will jump and highlight the Filters button

The third page will jump dynamically to the first video on the list, highlighting the VS button



The final page will return to the center of the screen with further information

1.4.2 Filter button

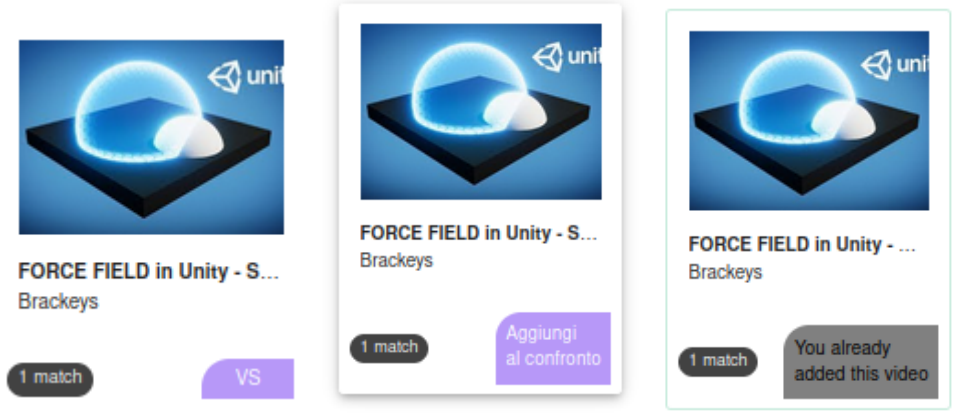


If you click the Filter Button, a new section will appear showing multiple clickable buttons to further filter the videos based on:

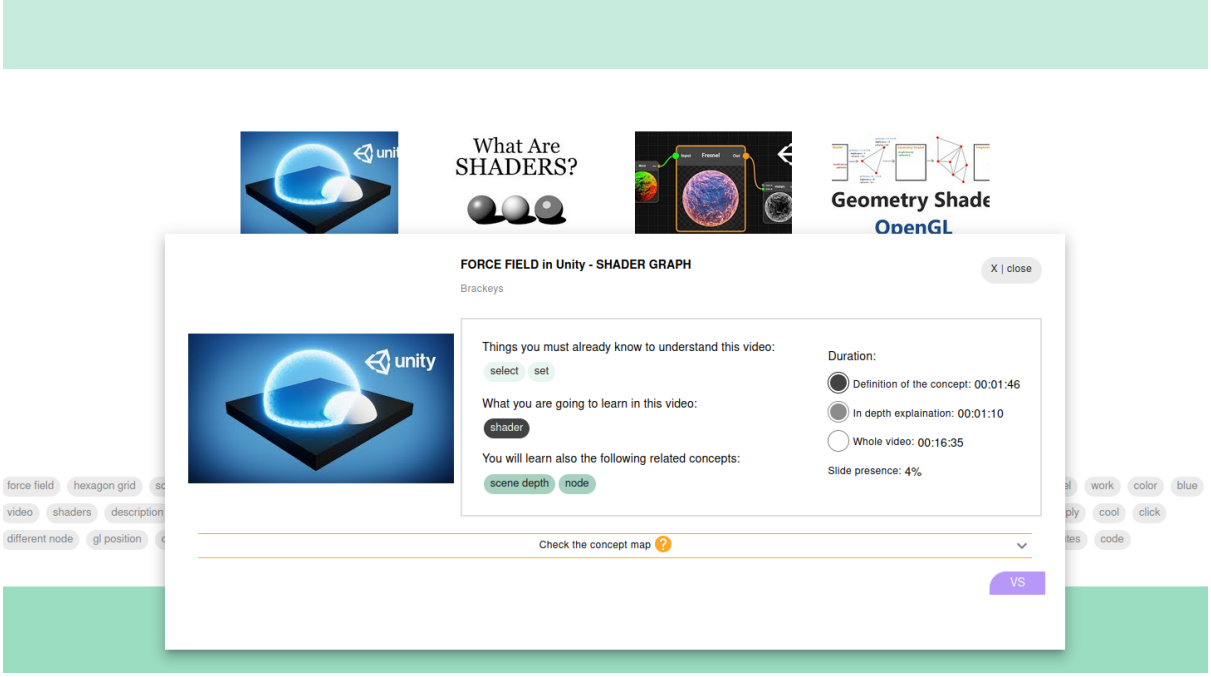
1. Suitable for = if no prerequisite, if prerequisite all explained within the video, if 80% of the prerequisites are explained within the video, it's Beginners. Otherwise Experts
2. Explanation = If the concept selected has only conceptDefinition value in the DB it's basic. If it also has conceptExpansion value it's detailed
3. Kind of lesson = Check the presence of slides in the video
4. Definition of concept = Choose the length of the conceptDefinition value
5. In depth explanation = Choose the length of the conceptExpansion value
6. Whole video = Choose the length of the whole video
7. Sort by = Choose the sorting logic based on the available options

There are also 3 orange buttons that on hover mouse, will appear a help window to describe the meaning of the specific filter

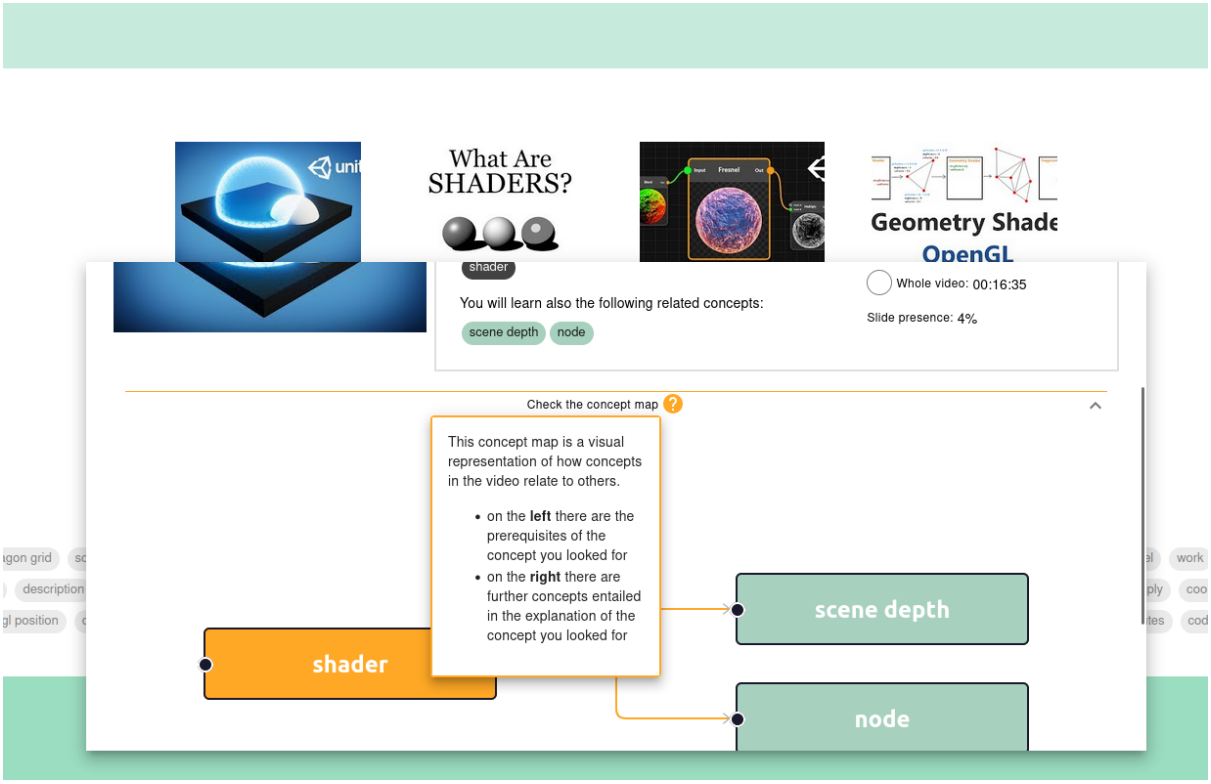
1.4.3 Video panel



This video panel is different from the previous one. The previous one has the number of concepts. Here you have a matching value and a button. Hovering the mouse on the panel will cast a shadow and if you hover on the button VS, it will show further information. If you click the button, it will be added for the comparison.



If instead you click the panel, a new window will pop up showing further information concerning about the list of prerequisite, the concept you selected, the derived concept, and the time value of the definition, in depth and the video duration. Also the presence of the slide. There is also the same expandible button to add it for the comparison.



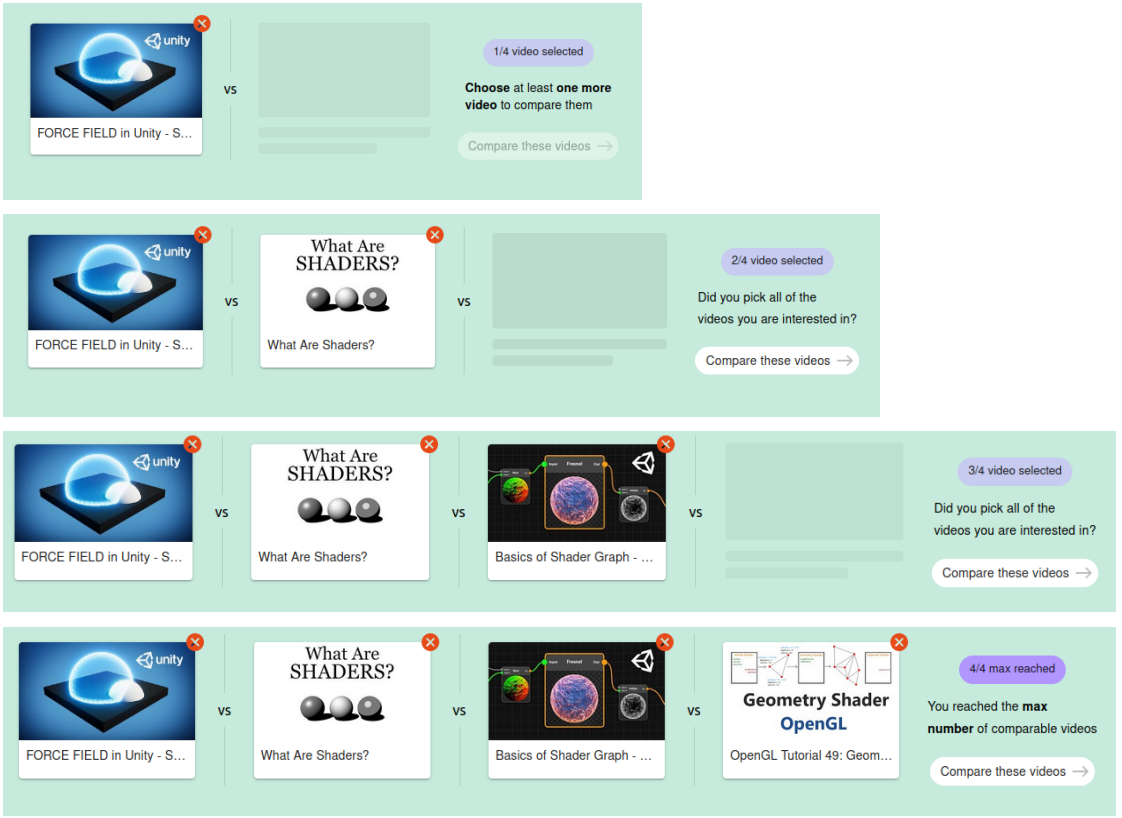
If you click the golden button, it will show the concept map. Also still present a help icon to popup some further information to explain the concept map

1.4.4 Button at end of page

Notice at the end of the page, many buttons. Those are the list of all concepts present in the videos filtered. So if you interested in comparison those videos with a different concept, click on it and the searchbar will be updated with the new concept.

Beware that some concept isn't present in all the filtered videos, so if you start with 4 maybe you end up with just one!

1.4.5 Video selected for comparison

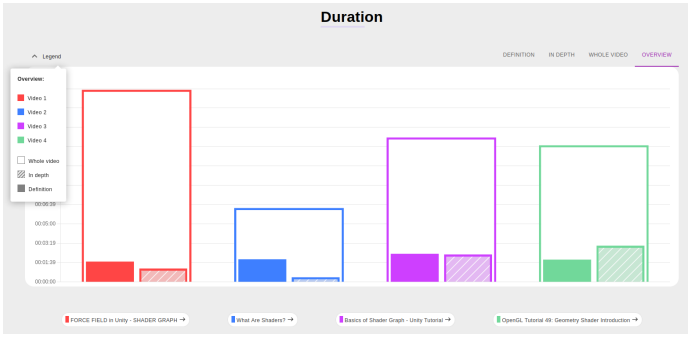


When pressing the VS button, a new section will pop up between the searchbar and the list of videos. The selected videos will appear in a panel with just the image of the clip and the title with a red button to remove it from the selected list. At the right some further information to guide the user.
The compare button will be enabled when the selected videos are between 2 and 4 and after that it will bring the user to the next page, where the user can see more detailed information about the selected videos and decide what to watch in the end

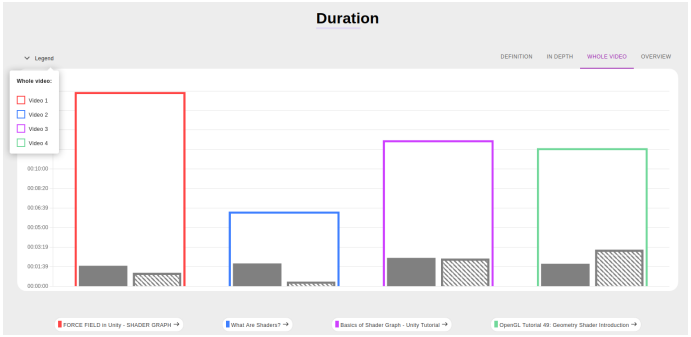
1.5 Third page - Compare the videos

At the top of the page you will see a searchbar with the previously selected concept. You can press the X to remove it and go back to the previous page to select another concept.

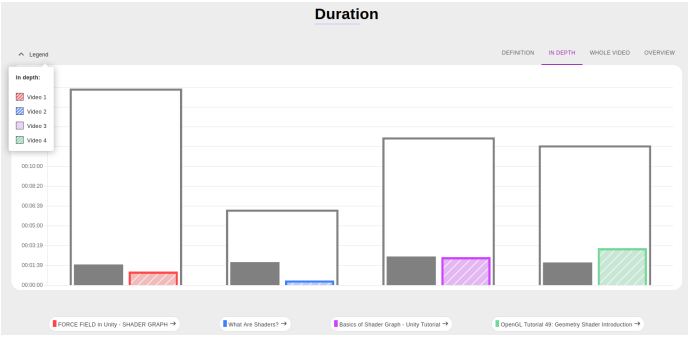
At the right it's the list of all video selected for the comparison.



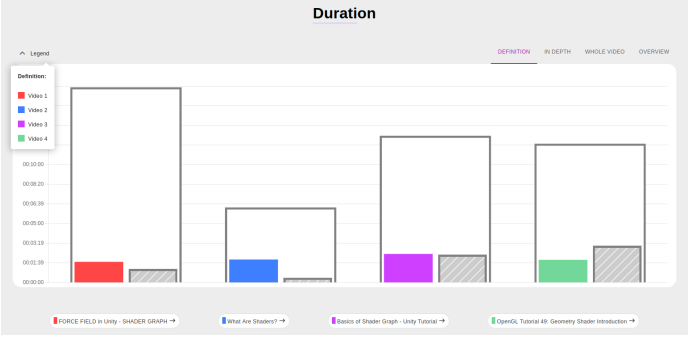
The first graph is a barchart showing the length in terms of hh:mm:ss of the whole video compared to the length of the conceptDefinition and conceptExpansion



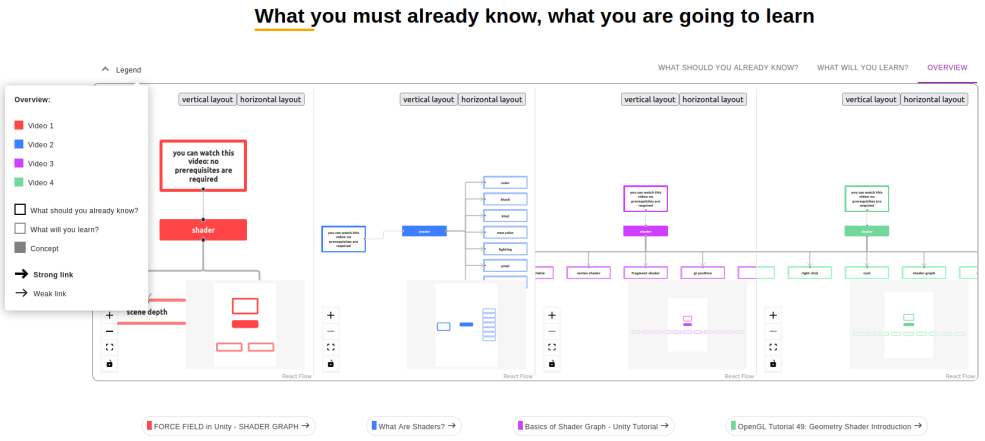
By pressing the button on the top right, you can focus on a specific bar. in this case the whole video length, making the other two grey. Notice how the Legend changes accordingly



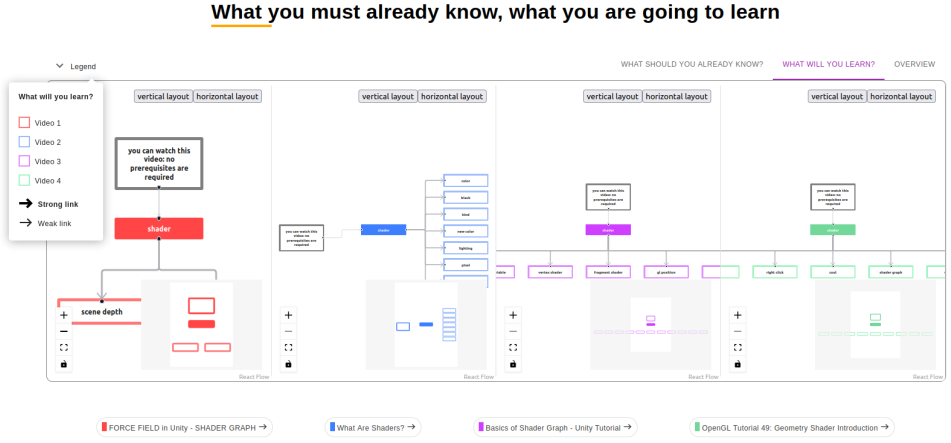
This one is for the In Depth. conceptExpansion

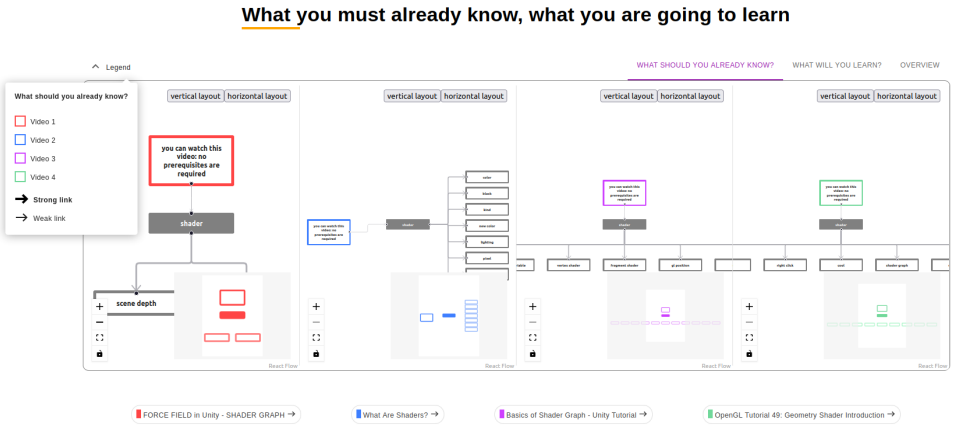


This one for Definition. conceptDefinition.
Notice at the end 4 buttons where if pressed you will go to watch that specific video

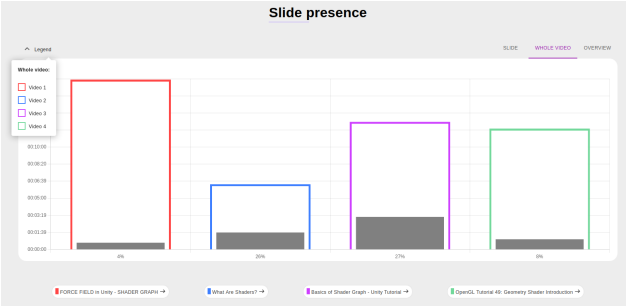


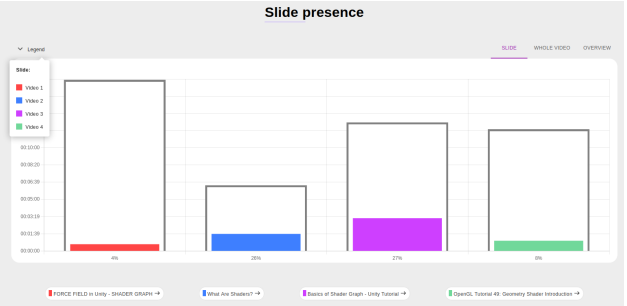
This one is the flowchart showing the relationship between prerequisite, concept selected, derived concept. You can rotate from vertical to horizontal view. You can see the whole structure in the minimap and you can a set of command to move and control the chart. If no prerequisite is found, a message will pop up warning the user.

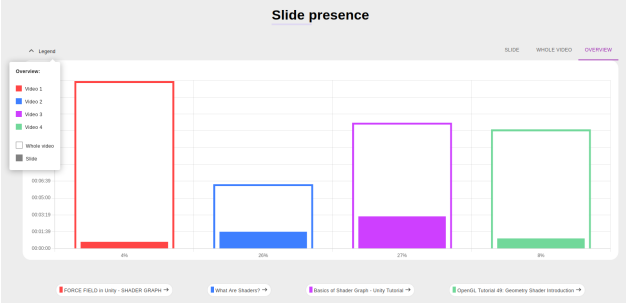




Notice how as the barchart, the Legend and the color changes accordingly







This graph is about the percentage of slide presence in the video. As before, you can focus on a specific value with the buttons and the Legend will changes too

2.1 React Structure

2.2 Comparison.js

When entering the first time into the comparison page and before clicking the button to compare videos (before going to the other page with the three graph), everything is handled by the root component Comparison.js

Here the structure:

- Comparison.js
    - Tutorial.js
    - Querybar.js
        - Queryinput.js
            - Buttonsecondary.js
        - Filters.js
        - Videoselected.js
    - Listvideo.js
        - Videoavailable.js
        - VideoFiltered.js
            - FlowChartSmall.js

Comparison.js
Holds all the structure and perform all the call to retrieve the data from the database and holds most of the data using useState

Tutorial.js
The tutorial popup will appear only after the first query or by pressing the button inside in **Querybar.js** but only after the first query

Querybar.js
It's the whole section above the **Listvideo.js** holding the searchbar, the filters, the videoselected and the button to go to the comparison result. The button will use history to send the data to the next page, **Result.js**

Queryinput.js
It's the searchbar used to search for the concept and filter the video

Buttonsecondary.js
A button that will send back all the data from this child to the root **Comparison.js** to update the state

Filters.js
To open and close the filters, the button is situated inside **Querybar.js** and press it will show the filters, to filter more the video after the first search of the concept

Videoselected.js
The video selected for comparison. It has a red button to remove it

Listvideo.js
Depending on the concept selected inside **Queryinput.js** it decides if show **Videoavailable.js** or **VideoFiltered.js**

Videoavailable.js
All video from DB uses this component and by pressing it it will go to the page to view the video

VideoFiltered.js
Appears after selected the concept. It has a lot of information and also a small flowchart used by **FlowChartSmall.js**

FlowChartSmall.js
It's a small flowchart using React-Flow

2.3 Result.js

This is the page after the user selects the videos and press the button from the previous page. It holds all the three graph showing in depth information about each videos.

Here is the structure:

- Result.js
    - BarroGraph.js
    - FlowChart.js
    - BarroGraph2.js

Result.js
The root parent holding all the information of the page

BarroGraph.js
It uses react-chartjs-2 to use the barchart to show the length of the concept length compared to the whole video

FlowChart.js
It uses React-Flow to show the structure of the concept. Like **FlowChartSmall.js**, this one has further information and controls over the graph

BarroGraph2.js
Similarly to the other one, but showing the slide presence

3.1 Backend

```
@app.route('/api/ConceptVideoData/<video_id>/<concept_searched>')
@auth.login_required
def ConceptVideoData(video_id_list, concept_searched):
```

This function is called from react using that API and this function will perform a query using SPARQL to the mongodb. We have already the information using an old API but that old uses the collection Videos. We need further information about the video and all those information resides inside Graphs collection and we use SPARQL query to retrieve it.

```
result = {
'video_id':video_id,
'concept_starttime':[],
'concept_endtime':[],
'explain':[],
'list_preconcept': [],
'list_prenotes':[],
'list_postnotes':[],
'list_derivatedconcept':[],
'video_slidishness':
}
```

This is the structure that initialize before the query and query after query we collect the information and send it back to the user.

The query is called from the user inside **Comparison.js** by the function

```
async function QueryConceptExtra(videoid, concept) {
```

From **Comparison.js** we collect the list of videoid we want to compare + the concept and we send the array + concept to the **main.py** for the flask server to handle the data and perform the query so a for loop is performed throughout the element inside the array **video_id_list** inside the function **ConceptVideoData()** and for each loop we perform the following queries:

To retrieve the created data of the annotation

```
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX edu: <https://teldh.github.io/edurell#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?created
WHERE{
        ?who oa:motivatedBy oa:describing.
        ?who dcterms:created ?created.
        ?who a oa:Annotation.
        ?who oa:hasBody ?c_id.
        ?c_id skos:prefLabel ?c_selected.
}
```

To retrieve the timeline of the selected concept and if it's about conceptDefinition or conceptExpansion

```
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX edu: <https://teldh.github.io/edurell#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?concept_starttime ?concept_endtime ?explain
WHERE {

        ?who oa:hasBody ?c_id.
        ?c_id skos:prefLabel ?c_selected.
        ?who oa:motivatedBy oa:describing.
        ?who oa:hasTarget ?target.
        ?target oa:hasSelector ?selector.
        ?selector oa:hasStartSelector ?startselector.
        ?startselector rdf:value ?concept_starttime.
        ?selector oa:hasEndSelector ?endselector.
        ?endselector rdf:value ?concept_endtime.
        ?who skos:note ?explain
}
```

To retrieve the prerequisite concept of the selected concept

```
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX edu: <https://teldh.github.io/edurell#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?preconcept ?prenote
WHERE{
        ?who oa:hasBody ?preconceptIRI.
        ?c_id skos:prefLabel ?c_selected.
        ?who oa:motivatedBy edu:linkingPrerequisite.
        ?who oa:hasTarget ?target.
        ?target dcterms:subject ?c_id.
        ?preconceptIRI skos:prefLabel ?preconcept.
        ?who skos:note ?prenote.

}
```

To retrieve the list of concept where our selected concept is their prerequisite

```
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX edu: <https://teldh.github.io/edurell#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?c_derivated ?postnote
WHERE{
        ?who oa:hasBody ?c_id.
        ?c_id skos:prefLabel ?c_selected.
        ?who oa:motivatedBy edu:linkingPrerequisite.
        ?who oa:hasTarget ?target.
        ?target dcterms:subject ?c_derivatedIRI.
        ?c_derivatedIRI skos:prefLabel ?c_derivated.
        ?who skos:note ?postnote
}
```

To retrieve the timeline as the concept but this time for the our ?c_derivated

```
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX edu: <https://teldh.github.io/edurell#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?dc_starttime ?dc_endtime
WHERE {
        ?who oa:hasBody ?c_id.
        ?c_id skos:prefLabel ?c_selected.
        ?who oa:motivatedBy oa:describing.
        ?who oa:hasTarget ?target.
        ?target oa:hasSelector ?selector.
        ?selector oa:hasStartSelector ?startselector.
        ?startselector rdf:value ?dc_starttime.
        ?selector oa:hasEndSelector ?endselector.
        ?endselector rdf:value ?dc_endtime.
}
```

For video_slidishness is used flask mongoengine instead of rdflib, because the information isn't inside the graph collection but in an another collection that doesn't use json-ld notation

```
 VTS=VideoTextSegmentation.objects(video_id = video_id)
       for VTSdoc in VTS:
          result['video_slidishness'] = str(VTSdoc.video_slidishness)
       result_list.append(result)
```