PUI Final Write-up

Vicky Sun | Lab E

About

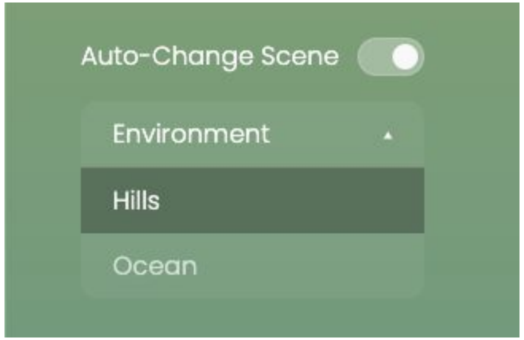
Study Space is a website helping people focus on their work without distractions. It’s a 3D environment that immerses people into a digital world with some visual interest to prevent the user from clicking away to other distracting websites, while having minimal distraction to keep the user from getting distracted. There’s also a figure in the scene studying alongside to user to keep them even more focused.

Users can play ambient music in the background to help them retain attention, and select an environment of their choice for display, or else let the system change the environment according to the type of music they’re playing.

The website is best viewed from a computer screen.

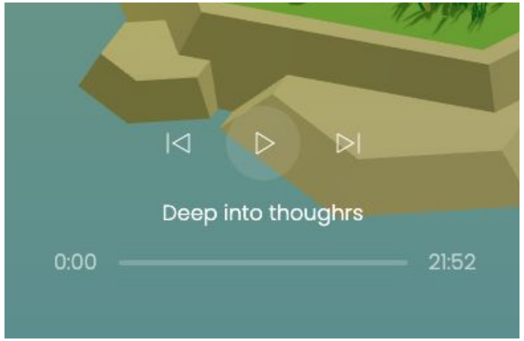
Interactions

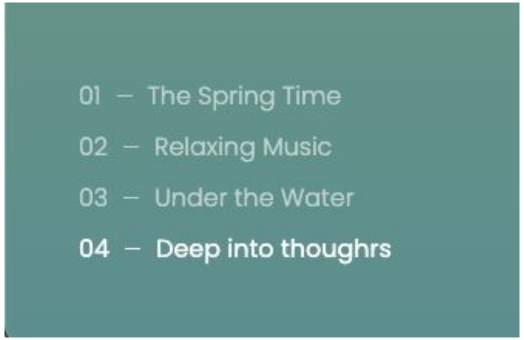
- Selecting an angle of the 3D scene:
    - Drag around on the scene to rotate the angle, or scroll in and out to determine how big you want it to be
- Control the environment:
    - Toggle auto-change scene to determine if you want the website to automatically change the environment based on the current music
    - Select the environment you want on the dropdown below. Know that if you haven't toggled off auto-change, the next song with a different environment will still change it



Interactions

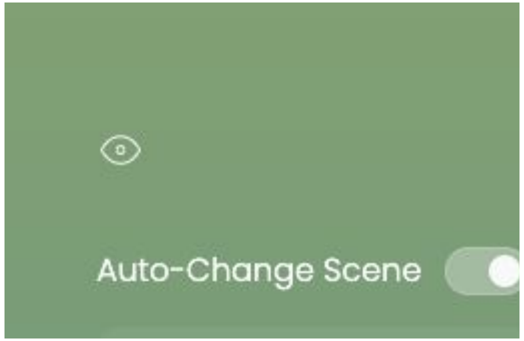
- Music controls:
  - Play/pause/skip the music you’re listening to, or drag the progress bar to skip to your desired spot
  - Once you have played the music before, you can play/pause/skip via the keyboard (eg., F7-F9 keys on macbook). Arrow keys will increment/decrement the progress by 10 seconds
- Select the music:
  - Click on the list of music to select what you want to listen to

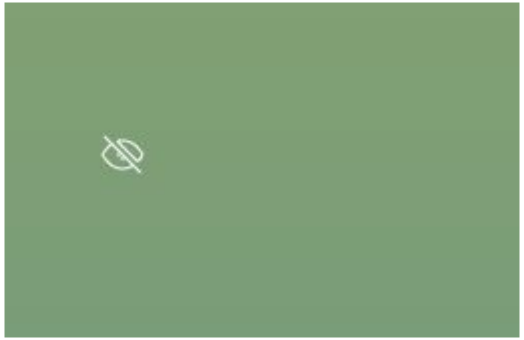



01 – The Spring Time
02 – Relaxing Music
03 – Under the Water
04 – Deep into thoughrs

Interactions

- Show/hide controls:
  - Click on the eye icon on the top left corner to hide most controls to enhance immersion





Tools

- Next.js
  - Framework for the entire project
  - Chose this instead of React.js because I originally planned to incorporate Spotify API to the website and didn't want to create a separate backend
- React Three Fiber / Three.js
  - Constructing the 3D environment in the website
  - It's an existing 3D library with robust features and tutorials for reference
- Three-Custom-Shader-Material
  - Used for implementing custom vertex shaders for the animated grass/flower and kelp
  - Can pass in an existing Three.js material to build on top of it, preventing having to hand-craft everything
- React-Three/Drei
  - Used for importing GLTF 3D models, using animations and the rotating/zooming controls of the 3D scene

Tools

- SCSS & Modules
    - Easier to find elements through SCSS structure
    - Easier naming by having different modules for each component
    - Global variables
- React-Use-Audio-Player
    - Assist with music controls such as loading/playing/pausing/seeking position
    - It’s often hard to create custom interfaces with existing music control packages, so I found this hook that only provides the functions, and I could hand-craft the UI and its implementation logic
- React Context API
    - Handling state control across multiple components
    - Used it because each section of the interface is its own component and on the same hierarchy, and I wanted to keep the code as clean as possible without inserting too much state management on the parent component
- Media Session API
    - Play/pause/skip keyboard controls
    - Media control keys affect the playing state of the music, and it needed to affect the UI as well to prevent confusion for the users

Tools

- Classnames
  - Assists handling classNames with convenience and simpler syntax
  - Helps the code look cleaner
- Framer Motion
  - Animating small interactions
  - Eg., used it to animate between auto height and 0px for the environments dropdown
- Typescript
  - Data type management
  - Used it for practice purposes
- React-Toggle
  - Toggle component for controlling auto-change environment
  - Overrode its style with custom css
- Blender
  - 3D editing software
  - Create custom 3D objects that I couldn’t find on the internet
  - Create animation for the reading character
  - Assemble 3D space and exporting into GLTF file

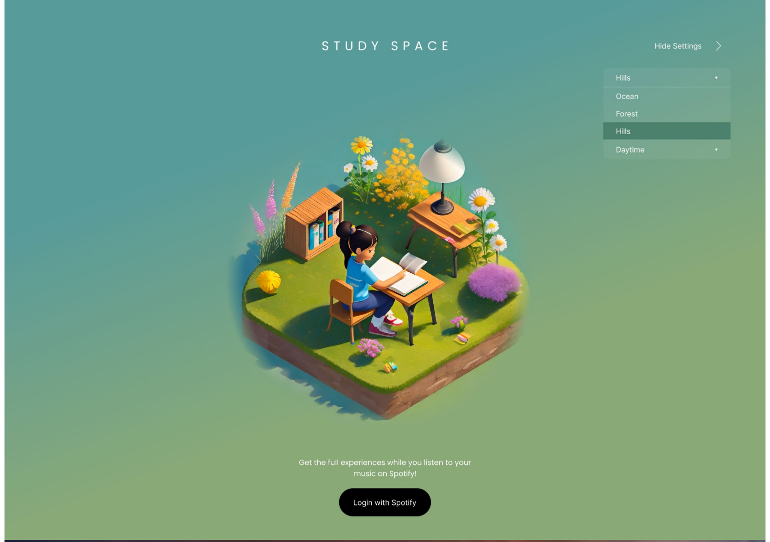
Tools

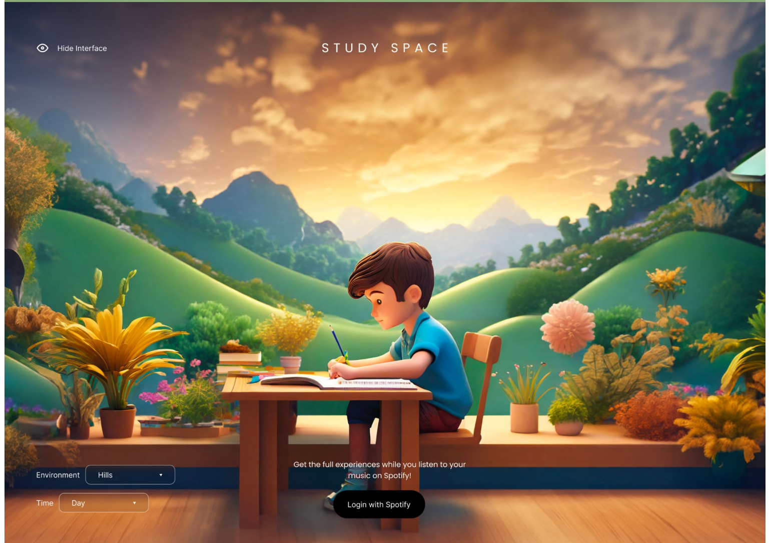
- Pixabay
  - Website used for finding music resources
- Poly.pizza
  - Website used for finding 3D models
- Sketchfab
  - Website used for finding character 3D model

Iteration #1

For my first iteration, I wanted to incorporate Spotify API to have a diverse user option for the music, and that the environment would change based on the genre of the music.

However, while I was testing and trying to implement the Spotify Web API, I discovered that it required the user to open Spotify on a device to be able to play music, creating a disconnection between the music and the website. This further proved to be an issue as I found there were no measures to notify the website if the user has paused/played the music on the playing device to update the UI accordingly. The other Spotify service Web Playback SDK required the logged in user to have a premium account, which even though it worked without an issue, it was not applicable for the grading of this project.

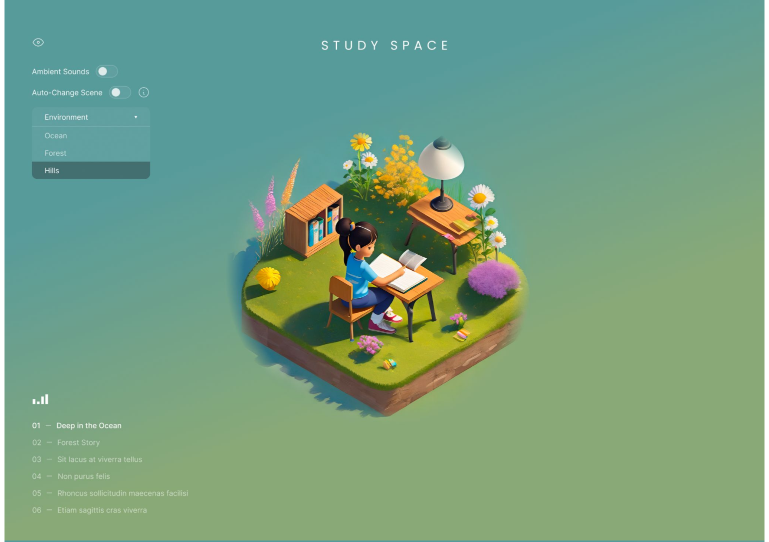


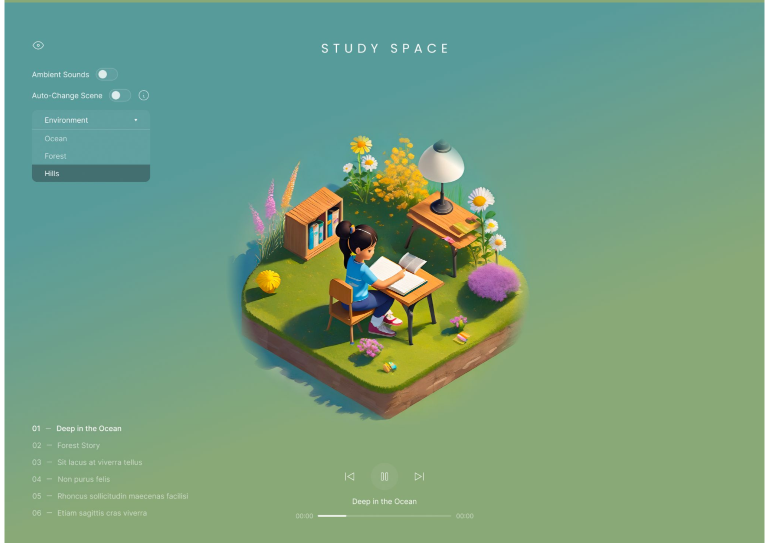

Hide Interface
STUDY SPACE
Environment
Hills
Time
Day
Get the full experiences while you listen to your music on Spotify!
Login with Spotify

Iteration #2 & #3

Due to the limitations of Spotify API, I decided to remove it entirely and have a set list of music in the website for the music to listen to.

After some more feedback for the designs, I added the basic music controls that people told me they wanted - play/pause/skip and having a progress bar to skip to whichever location they wanted to.

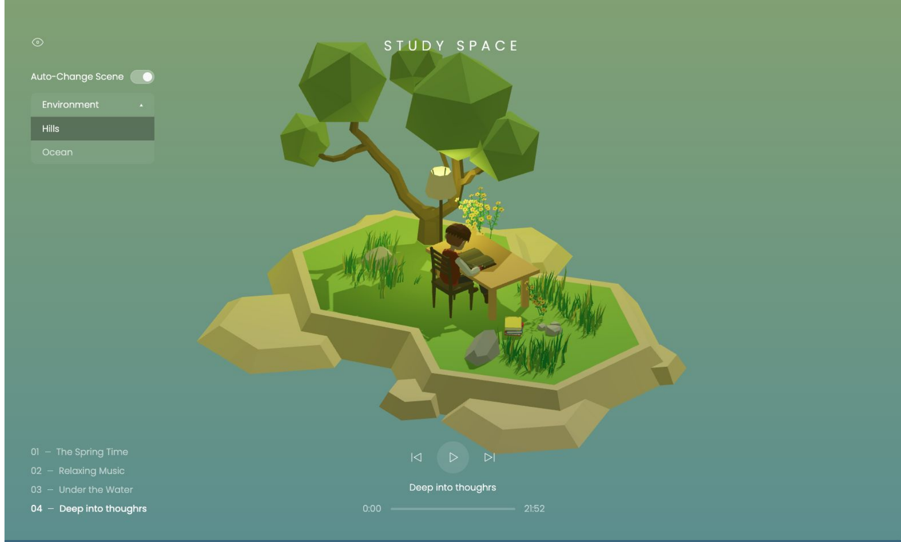


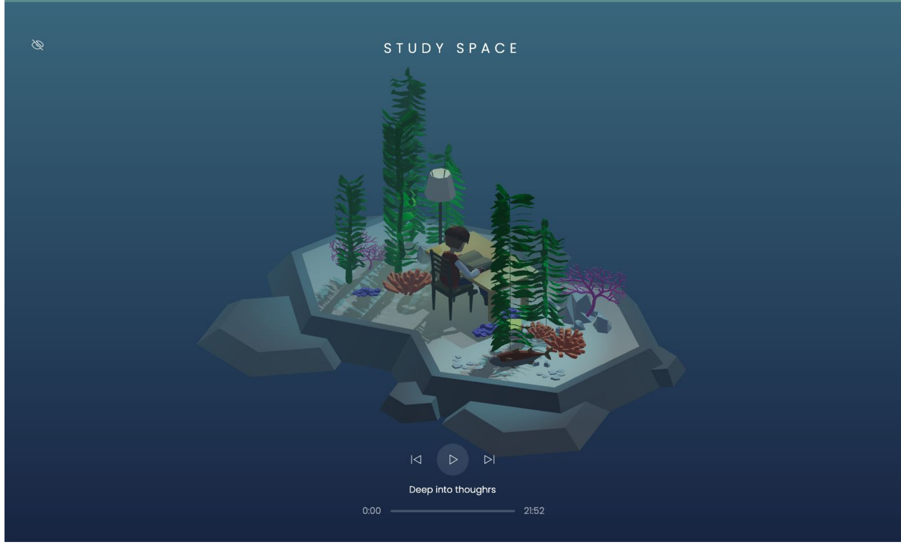


Implementation

In the final product, I have implemented everything aside from ambient sounds, hence the website is missing a toggle for that.

This is due to the fact that the music I chose already have an ambient quality to them, and I found that people enjoyed it much more without the ambient sounds. In addition, I couldn't find a suitable track for the Hills environment.



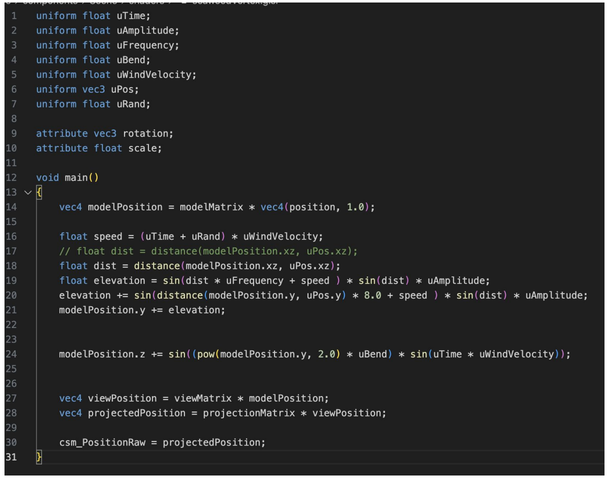


Challenges

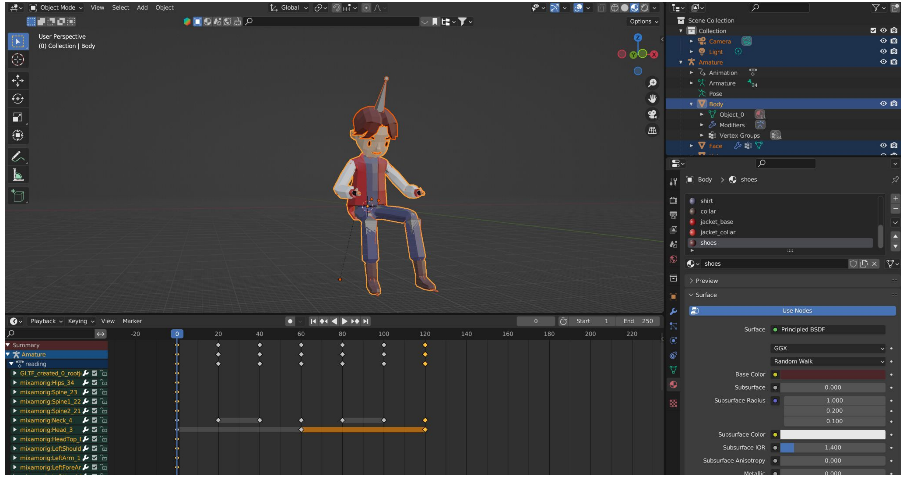
Aside from struggling to find a way to workaround Spotify's API in the initial stages, the most challenge I faced was making the custom vertex shader and creating the animation for the character in Blender.

I watched a tutorial to make the vertex shader, and though the code for it doesn't seem complicated, my terrible math cost me hours to implement the simple swaying motion I wanted.

The character animation was somehow buggy when I imported it into the scene, so I tried re-configuring the armature and the model in Blender, which is a software I'm not familiar with and had to watch many tutorials in the process.

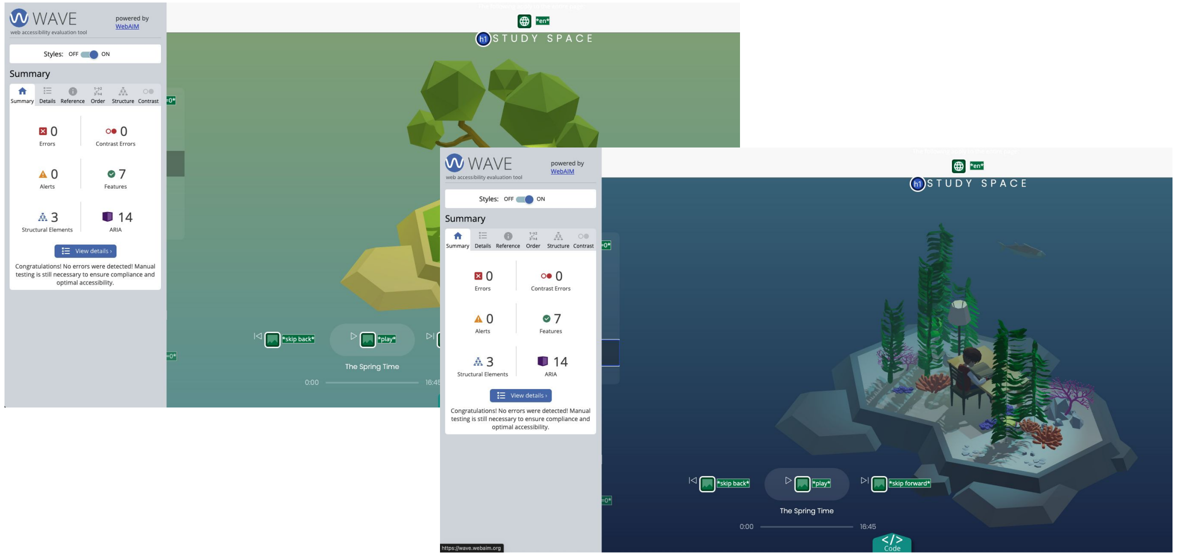


```
uniform float uTime;
uniform float uAmplitude;
uniform float uFrequency;
uniform float uBend;
uniform float uWindVelocity;
uniform vec3 uPos;
uniform float uRand;

attribute vec3 rotation;
attribute float scale;

void main()
{
    vec4 modelPosition = modelMatrix * vec4(position, 1.0);

    float speed = (uTime + uRand) * uWindVelocity;
    // float dist = distance(modelPosition.xz, uPos.xz);
    float dist = distance(modelPosition.xz, uPos.xz);
    float elevation = sin(dist * uFrequency + speed ) * sin(dist) * uAmplitude;
    elevation += sin(distance(modelPosition.y, uPos.y) * 8.0 + speed ) * sin(dist) * uAmplitude;
    modelPosition.y += elevation;


    modelPosition.z += sin((pow(modelPosition.y, 2.0) * uBend) * sin(uTime * uWindVelocity));


    vec4 viewPosition = viewMatrix * modelPosition;
    vec4 projectedPosition = projectionMatrix * viewPosition;

    csm_PositionRaw = projectedPosition;
}
```



Accessibility Summary

Note: for some reason the website background doesn't show up via the website, but works for the google extension tool


WAVE web accessibility evaluation tool, powered by WebAIM

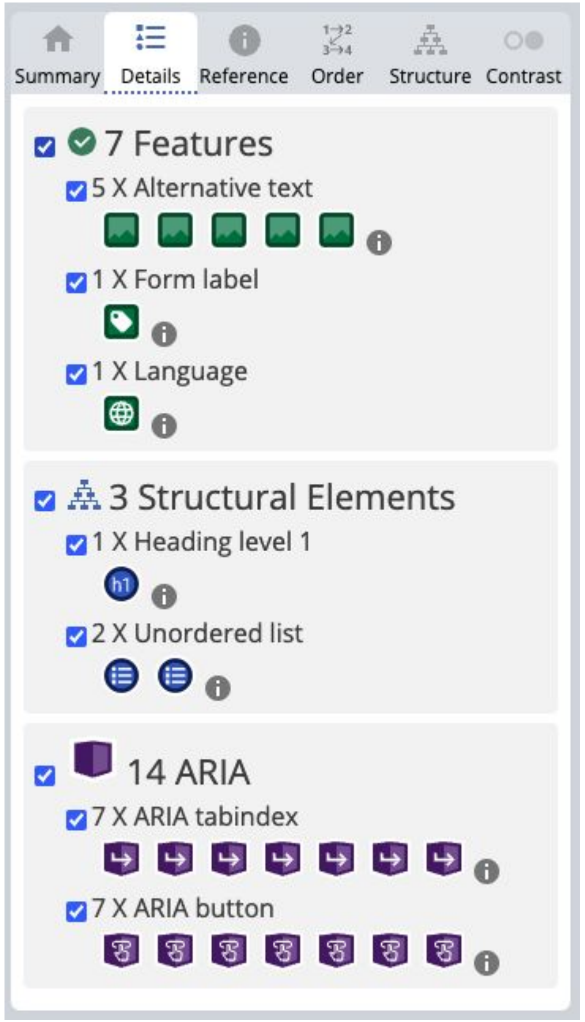
Styles: OFF ON

Summary

0 Errors	0 Contrast Errors
0 Alerts	7 Features
3 Structural Elements	14 ARIA

View details ›

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.



7 Features
- 5 X Alternative text
- 1 X Form label
- 1 X Language

3 Structural Elements
- 1 X Heading level 1
- 2 X Unordered list

14 ARIA
- 7 X ARIA tabindex
- 7 X ARIA button