You can navigate to google.com, youtube.com, wikipedia.org, stackoverflow.com, immersive-web.github.io, doc.babylonjs.com, npmjs.com

You can also view the source files for the DIAversity project: app.ts, hmd.ts, ui.ts, frustumVisualizer.ts, main.ts

(note that any links directing to other domains will not work)

#### Question 1 (1 point)

Rank the tools from the most accessible (for developers without coding experience to use) to the least.

OpenXR SDK in C++
Unity
CoSpaces

#### Question 2 (1 point)

You need to implement a cross-platform native VR experience using a novel hand tracking algorithm and you want to focus on implementing VR components with the same underlying code-base across Meta's Quest HMDs, HTC's Vive HMDs, Google Cardboard. You hope to align it to an open standard so that there is a vibrant developer ecosystem and that the underlying tooling will be constantly improved. What standard should you be targeting?

OpenCL
Vulkan
OpenXR
WebXR
OpenGL

## Question 3 (1 point)

WebGL

You introduced a new "/components" directory containing new source files that are

part of your Babylon.js project. Which file should you modify to make sure your web app still runs?
app.ts main application logic file
package-lock.json Ensures consistent dependency versions across different environments
Configures the TypeScript compiler. It defines how TypeScript code should be compiled to JavaScript, which files to include/exclude, module resolution settings, and more.
main.ts entry point of your TypeScript application
index.html HTML entry point of your web app
package.json project metadata and dependencies. It's essential for Node.js-based projects and helps with dependency management, scripts automation, and project configuration.
Question 4 (1 point)
In your WebXR typescript-based app, you have edited a file in your project directory to specify that the Babylon.js version should be 7.3.0.
After running `npm install`, a file will be automatically updated in your project directory. Name the file in your project directory that will be updated.
package-lock.json &
Question 5 (1 point)
<pre>import { Engine, Scene } from "@babylonjs/core";</pre>
<pre>class App {    private engine: Engine;</pre>
<pre>constructor(engine: Engine) {     this.engine = engine; }</pre>
<pre>createScene() {    const scene = new Scene(this.engine);</pre>

scene.createDefaultCamera(true, true, true);

scene.createDefaultLight(true);

return scene;

}

```
}
 const canvas =
 <HTMLCanvasElement>document.getElementById('renderCanvas');
 const engine = new Engine(canvas, true);
 const app = new App(engine);
 app.createScene().then(scene => {
      engine.runRenderLoop(() => {
           scene.render():
      })
 });
 When you run the code above, the following error is shown in the console:
 app.createScene().then' is undefined
 The error is a missing keyword in the code. What is the missing keyword?
                                          Adding async makes createScene() return
   async
                                          a Promise.
                                          The then() method works on Promises, so
                                          this resolves the error
Question 6 (1 point)
 In IPA1, the package.json file looks like this:
 {
    "name": "hello-immersive",
    "private": true,
    "version": "0.0.0",
    "type": "module",
    "scripts": {
      "dev": "vite",
      "build": "tsc && vite build",
      "preview": "vite preview",
      "test": "vitest --reporter=verbose",
      "test:ui": "vitest --ui",
```

"test:run": "vitest --run",

```
"clean": "rm -rf dist node modules package-lock.json"
    },
    "devDependencies": {
      "@vitest/ui": "^1.1.1",
      "typescript": "^5.2.2",
      "vite": "^5.0.8",
      "vitest": "^1.1.1"
    },
    "dependencies": {
      "@babylonjs/core": "^6.36.1"
    }
 }
 You want to deploy your Babylon.js application to a web server. What should you
 run in the terminal to get the files to deploy?
     npm run preview
    ) npm run vite
    ) npm run test
  npm run build
    npm run dev
Question 7 (1 point)
 Where is createScene(...) normally defined?
  package.json
  tsconfig.json
   ) index.html
  app.ts
     package-lock.json
```

# Question 8 (1 point)

Which file should you add to your team's version control if you want the project

index.html
o app.ts
package.json
_ tsconfig.json
Question 9 (1 point)
You want to add a custom script as part of your BabylonJS project test workflow. Which file should you modify?
index.html
√ package.json
o app.ts
opackage-lock.json
tsconfig.json
Question 10 (1 point)
When connecting the Meta Quest 2, there is no prompt to enable connection and can't see developer options in the settings. What is the likely issue?
Headset is in low power mode
Olid not tap the Build Number 7 times in settings
Olid not connect a USB data cable from PC to headset
✓ Did not enable Developer Mode on the Meta Horizon app on the phone
Question 11 (1 point)
What does the following Babylon.js snippet accomplish?
<pre>const xr = await scene.createDefaultXRExperienceAsync({     uiOptions: {</pre>

```
sessionMode: "immersive-vr",
},
});

Creates a default BabylonJS scene with a sphere and plane, and then initializes typical components for VR experiences.

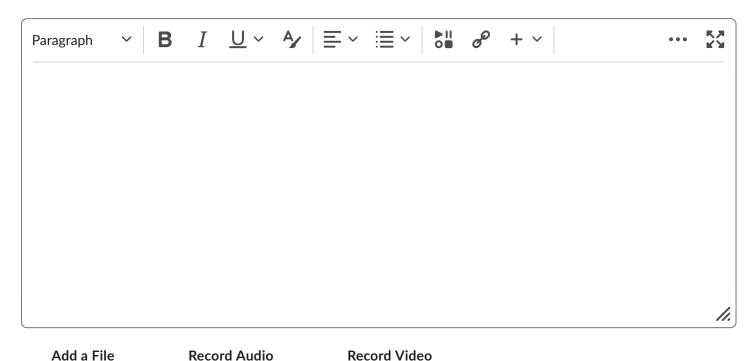
Initializes typical components for hybrid XR (AR & VR) experience.

Makes a non-blocking method call to initialize typical components for VR experiences.
```

### Question 12 (1 point)

Please write any feedback you have so far, for this CSD312x module

Creates a default BabylonJS scene with a sphere and plane.



Add a File Record Audio Record Video

Submit Quiz

0 of 12 questions saved