Question 1 (1 point) ✓ *Saved*

You are tasked to build a VR escape room game where users will move around freely to search for clues spatially distributed in a small room. Realness and cybersickness are important concerns and augmentation of movement speeds is known to induce more symptoms.

What locomotion technique is best suited for this use case?

- ◯ teleportation
- ◯ joystick-based
- ◯ walking-in-place (WIP) with KatVR 360 slidemill
- ◉ tracking real movement in physical space
- ◯ walking-in-place (WIP) with HTC Vive HMD and trackers

Question 2 (1 point) ✓ *Saved*

In the Meta Quest 2 menu, you can use the Quest controllers to navigate to different apps and configuration settings. What form of interaction authenticity does this implement?

You can view the YouTube sequence below if you're unsure of the interaction described above.

https://youtu.be/wVkivmMKikU?t=533

- ☐ Natural interaction
- ☑ Artificial augmented natural interaction
- ☐ Artificial magical interaction

Question 3 (1 point) ✓ *Saved*

In your WebXR typescript-based app, you want to include and use the Mocha test framework only during development (and not during deployment).

Write the **full command (no aliases or short forms)** you will type into a command line interface in the project directory (containing package.json) that will fulfil the above requirement using NPM (package is named `mocha` in NPM).

https://www.npmjs.com/package/mocha

npm install --save-dev mocha

Question 4 (1 point) ✓ *Saved*

In the Google Cardboard HMD, you want to make the generated virtual image look **taller vertically.**

What **dimension(s)** can you change to achieve this?

- [ ] Height of the physical display
- [x] Eye relief
- [ ] Width of the physical display
- [x] Focal length of the lenses
- [ ] IPD between the lenses
- [x] Distance between the lenses and the physical display

Question 5 (1 point) ✓ *Saved*

What sort of affordance will most likely occur when you strap on vive trackers **only on the hands** for an experienced VR user in a VR commuting simulator application with walking as the main interaction?

- ○ The user will only consciously use the feet in his/her own way to navigate in the virtual environment
- ○ The user will only consciously use the hands in his/her own way to navigate in the virtual environment
- ○ The user will reach out for a vive controller to use the thumbstick to navigate in the virtual environment
- ○ The user will only consciously lift the legs in a natural fashion to navigate in the virtual environment
- ○ The user will perform a natural walking action with the whole body to navigate in the virtual environment
- ◉ The user will only consciously swing the hands in a natural fashion to navigate in the virtual environment
- ○ The user will consciously use both the hands and feet in his/her own way to navigate in the virtual environment

Question 6 (1 point) ✓ *Saved*

Your UX team aims to enhance the sense of control and reduce self-consciousness, disorientation and nausea. They will run user studies before and after some key feature changes are made to your existing VR application.

What **is/are** the possible famous validated questionnaires to use in the user studies, pertinent to the aims above, to aid in design decisions for your UX team?

- [x] Virtual Reality Sickness Questionnaire (VRSQ)
- [x] Flow State Scale (FSS)
- [ ] Igroup Presence Questionnaire (IPQ)
- [ ] Systems Usability Scale (SUS)
- [x] Simulator Sickness Questionnaire (SSQ)

Question 7 (1 point) ✓ *Saved*

When implementing motion controller interactions in WebXR for a PICO 4 headset, what is the component `id` to use when you want to add an Observer that provides some behavior when the controller thumbstick is operated on?

- (•) xr-standard-thumbstick
- ( ) xr-pico4-thumbstick
- ( ) xr-thumbstick-pico4
- ( ) xr-thumbstick
- ( ) pico4-thumbstick

Question 8 (1 point) ✓ *Saved*

Which file should you amend in your WebXR project to downgrade the babylonjs package version to 5.25.0?

- (●) package.json
- ( ) None of the answers are correct
- ( ) app.ts
- ( ) webpack.config.js
- ( ) tsconfig.json
- ( ) index.ts

Question 9 (1 point) ✓ *Saved*

Which of the following **is/are** characteristic experiential dimensions of flow known in current research literature?

- [x] Loss of self-consciousness
- [ ] Involvement
- [ ] Realness
- [x] Warped sense of time
- [x] Sense of Control
- [x] Effortlessness

Question 10 (1 point) ✓ *Saved*

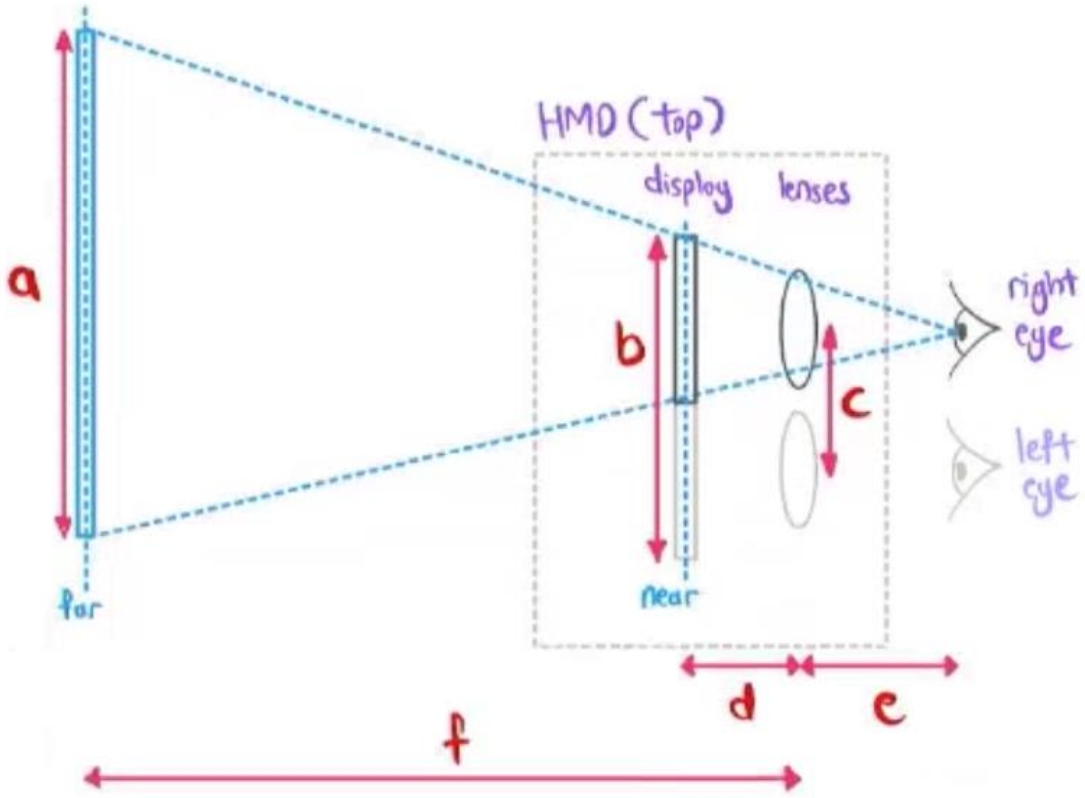
In the Google Cardboard HMD, you want to expand the **horizontal** FOV.

What **dimension(s)** can you change to achieve this?

- [ ] IPD between the lenses
- [x] Focal length of the lenses
- [ ] Height of the physical display
- [x] Distance between the lenses and the physical display
- [ ] Width of the physical display
- [x] Eye relief

Question 11 (1 point) ✓ *Saved*

In the schematic HMD diagram, which value(s) will be changed if **d** was changed?



- [ ] e
- [ ] f
- [ ] c
- [x] a
- [ ] b

Question 12 (1 point) ✓ *Saved*

Which interaction mechanic(s) do gaze-based interactions (like those found in the VR game Land's End) represent?

You can view the YouTube video below to get an idea of the gameplay.

Land's End YouTube Gameplay by VR Adventure

- [ ] body (excluding hands) gestures
- [x] viewpoint control
- [x] hand gestures

Question 13 (1 point) ✓ *Saved*

When designing for immersion in the next version of our VR commuting simulator, I want to improve the experience of **presence**.

What **data collection method(s)** can be appropriate when I want to find out whether this goal has been met after I have translated the above goal into implementation?

- [ ] Let users fill in the VRSQ
- [ ] Let users fill in the FSS
- [x] Let users fill in the IPQ
- [x] Perform semi-structured interviews with users
- [ ] Create telemetry tracking mechanisms to observe users

Question 14 (1 point) ✓ *Saved*

What is a result of reducing the eye relief?

- ○ decreased distance between lens to virtual image
- ◉ increased FOV
- ○ decreased FOV
- ○ increased distance between lens to virtual image

Question 15 (1 point) ✓ *Saved*

When designing for immersion in the next version of our VR commuting simulator, I want to improve the experience of **presence**.

Which of the following **is/are suitable approaches** that translate this goal into implementation?

- ☑ increase the visual fidelity of the graphics with custom physically based rendering shaders
- ☐ implement GUI elements to present clear goals for the user to attain at every point of the commuting experience
- ☐ implement mechanics to structure the commuting experience akin to completing progressively challenging levels in a game setting
- ☑ implement real-walking locomotion (tracking actual walking in a room-scale setting) instead of the current walking-in-place locomotion
- ☑ add AI-driven human characters with realistic behaviors in the simulation
- ☐ implement teleportation locomotion instead of the current walking-in-place locomotion

Question 16 (1 point) ✓ *Saved*

When designing for immersion in the next version of our VR commuting simulator, I want to improve the experience of **flow**.

Which of the following **is/are suitable approaches** that translate this goal into implementation?

- [x] implement mechanics to structure the commuting experience akin to completing progressively challenging levels in a game setting
- [ ] implement real-walking locomotion (tracking actual walking in a room-scale setting) instead of the current walking-in-place locomotion
- [ ] implement teleportation locomotion instead of the current walking-in-place locomotion
- [x] implement GUI elements to present clear goals for the user to attain at every point of the commuting experience
- [x] add AI-driven human characters with realistic behaviors in the simulation
- [ ] increase the visual fidelity of the graphics with custom physically based rendering shaders

Question 17 (1 point) ✓ *Saved*

The following code snippet is from a **text-plane.ts** file.

```
1  /**
2   * A display component that has a plane Mesh and TextBlock.
3   */
4  class TextPlane {
5      mesh: Mesh;
6
7      /**
8       * Ctor to create the plane Mesh and TextBlock.
9       */
10     constructor(
11         name: string,
12         width: number,
13         height: number,
14         x: number,
15         y: number,
16         z: number,
17         text: string,
18         color: string,
19         fontSize: number,
20         scene: Scene
21     ) {
22         const plane = MeshBuilder.CreatePlane(
23             name + " text plane",
24             { width: width, height: height },
25             scene
26         );
27         plane.position.set(x, y, z);
28         const planeTexture = AdvancedDynamicTexture.CreateForMesh(
29             plane,
30             width * 100,
31             height * 100,
32             false
33         );
34         planeTexture.name = name + " plane texture";
35         const planeText = new TextBlock(name + " text");
36         planeText.text = text;
37         planeText.color = color;
38         planeText.fontSize = fontSize;
39         planeTexture.addControl(planeText);
40
41         this.mesh = plane;
42     }
43 }
```

This TextPlane component was imported and used in a class in the **main app.ts** file.

However a build error occured complaining that 'TextPlane' could not be found.

First, write the offending line number that caused this bug:

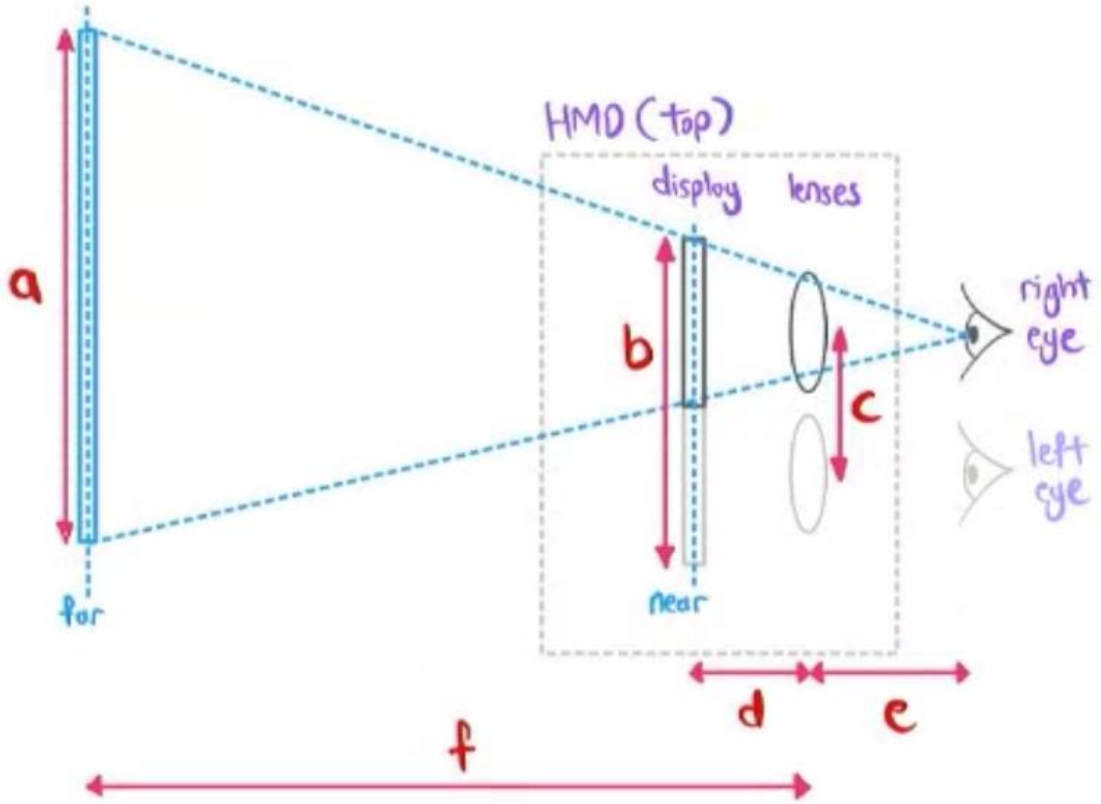
4

Next, write the single keyword that needs to added to this line in order for it to work:

export

Question 18 (1 point) ✓ *Saved*

In the schematic HMD diagram, where is the eye relief?



Question 19 (1 point) ✓ *Saved*

When designing for immersion in the next version of our VR commuting simulator, I want to improve the experience of **flow**.

What **data collection method(s)** can be appropriate when I want to find out whether this goal has been met after I have translated the above goal into implementation?

- [ ] Let users fill in the IPQ
- [ ] Create telemetry tracking mechanisms to observe users
- [ ] Let users fill in the VRSQ
- [x] Perform semi-structured interviews with users
- [x] Let users fill in the FSS

Question 20 (1 point) ✓ *Saved*

When using the `WebXRFeaturesManager` to `enableFeature`, you need to specify the name of the targetted desired feature, i.e., a name that represent cool interaction features like teleportation, walking-in-place locomotion and hand tracking.

Write the name of the babylonjs API **class** that contains, as properties, this list of available feature names?

WebXRFeatureName