META-META Club

WORKBOOK · MODULE 5 OF 5

Into the headset

Browser Worlds. WebXR, performance on phones and Quest, and the final ship.

Pro studios | 3 lessons | ~3.5 h | 4-question practice quiz | Notes pages

IN THIS MODULE

Into the headset

WebXR, performance on phones and Quest, and the final ship.

#	LESSON	TYPE	TIME	DONE
5.1	Add WebXR	XR	60 min	☐
5.2	Performance on phones and Quest	READ	30 min	☐
5.3	Ship: your browser world	PROJECT	120 min	☐

How to use this workbook

- Work through each lesson on metametaclub.com/learn, or listen to its narration, then use the page here to take notes.
- Tick the steps as you go, and write your answers under **Check yourself** before you look anything up.
- Do the practice quiz at the end, then mark it with the answer key on the last pages.
- Keep every **Ship it** piece. Together they become what you show.

Before you start

What do you already know about this, and what do you want to be able to make by the end?

LESSON 5.1 · XR · 60 MIN

Add WebXR

Make your browser world open in a Meta Quest 3 or 3S headset. Same page, same scene: one button enters VR. Then add thumbstick movement so people can travel through your streets.

Key points

How WebXR works

WebXR is the browser API for headsets. The page asks for an `immersive-vr` session.

A **reference space** sets where zero is. `local-floor` puts y = 0 on the floor, so your 1.6 m eye height comes from the real person, not the code.

To move someone, don't move the headset camera. Put the camera inside a **rig** (a Group) and move the rig.

Comfort

- Offer snap turning instead of smooth turning.
- Keep a steady horizon. Never move the camera without the person asking.
- Teleport is the most comfortable option. Add it once stick movement works.

Steps checklist

- [ ] Serve over HTTPS or `localhost`. WebXR needs a secure context.
- [ ] Add the code below to your world.
- [ ] On the Quest, open the Meta Quest Browser, load the page and press Enter VR. Use your laptop's address on the same Wi-Fi, over HTTPS.
- [ ] Set the camera's y to 0 in XR, since the floor reference adds real height.
- [ ] Move with the left thumbstick. Keep speed low; fast smooth motion makes some people sick.

Pushing the stick forward gives a negative y value, hence the minus sign.

```
This lesson has code samples. They are in the lesson on metametaclub.com/learn.
```

SHIP IT

Your world running in a Quest 3 or 3S, with a short clip recorded from the headset.

- [ ] Shipped. Link or file: _____

Check yourself

1. Why move a rig instead of the camera?

2. What does `local-floor` do to eye height?

3. Why must the page be served over HTTPS?

LESSON 5.2 · READ · 30 MIN

Performance on phones and Quest

Learn the budget your world must fit in on a mid-range phone and a Quest 3 or 3S, and the five tools that keep it there. Then measure your own scene.

Key points

The frame budget

A laptop screen wants 60 frames a second: about 16.7 ms per frame. A Quest headset runs at 72 Hz or more, often 90 Hz: about 13.9 ms or 11.1 ms, and it draws two eyes.

What costs the most

- **Draw calls.** Each separate mesh and material pair is a call to the GPU. Mobile chips hate many of them. Aim for low hundreds at most on Quest.
- **Pixels.** Overdraw, transparent layers and high pixel ratios multiply GPU work.
- **Textures.** A 4096 × 4096 texture uses far more memory than a 1024 × 1024 one. Mobile memory is small.
- **Lights and shadows.** Every shadow-casting light re-renders the scene from its own view.

Five tools

1. **Merge** static geometry that shares a material: `mergeGeometries`.
2. **Instance** repeated objects like streetlights, trees and kiosks with `THREE.InstancedMesh`: one draw call for hundreds of copies.
3. **Compress textures** with KTX2 (Basis Universal). They stay compressed on the GPU, so they use less memory and upload faster. Load them with `KTX2Loader`.
4. **Level of detail.** `THREE.LOD` swaps a simpler model when something is far away.
5. **Bake light.** Put lighting into textures in Blender for static scenes, and keep one real-time sun at most.

Steps to measure

1. Log `renderer.info.render.calls` and `renderer.info.render.triangles` once a second.
2. Check `renderer.info.memory.textures` and `.geometries`.
3. Profile on the real device. On a phone, use remote debugging from desktop DevTools.
4. Change one thing, measure again. Keep a short log.

A sane starting budget

- Pixel ratio capped at 2 on phones, 1 for heavy scenes.
- One shadow-casting light, small shadow map.
- Textures 1024 or 2048, KTX2 where possible.
- Merge or instance everything that repeats.

SHIP IT

A short performance log for your world: draw calls, triangles and frame rate before and after one fix.

☐ Shipped. Link or file: _____

Check yourself

1. What is the per-frame budget at 90 Hz?

2. When would you use InstancedMesh rather than merging?

3. Why do compressed GPU textures help on mobile?

LESSON 5.3 · PROJECT · 120 MIN

Ship: your browser world

Finish and publish your course piece: a drivable or walkable browser world of a real African place, built from real map data, that opens on a laptop, a phone and a Quest 3 or 3S.

Key points

What "done" looks like

Done is not "everything I imagined". Done is a world someone can open from a link, understand in ten seconds and enjoy for two minutes.

Respect the place

Name streets, neighbourhoods and cities accurately. Leave out faces, number plates and private homes unless people agreed.

Steps checklist

- [ ] **Freeze the scope.** Write one sentence: "A walk down X in Y, from A to B." Cut anything that doesn't serve it.
- [ ] **Polish the landmark.** The one place people will screenshot gets your best modelling, light and sound.
- [ ] **First ten seconds.** A loading state, a clear start position facing something good and a control hint.
- [ ] **Performance pass.** Hit your budget on a mid-range phone. Use your log from the last lesson.
- [ ] **Headset pass.** Enter VR on a Quest, travel the main route, fix anything uncomfortable.
- [ ] **Credits screen.** "© OpenStreetMap contributors", every model, texture and sound you didn't make, and the people who helped.
- [ ] **Publish** to a static host. Test the link in a private window on three devices.
- [ ] **Write it up.** Five short paragraphs: the place, why you chose it, how you built it, what you cut, what's next.

SHIP IT

Your public link plus the write-up, posted to the Metameta Nights gallery channel on the Metameta Club Discord, and a 30-second capture clip.

- [ ] Shipped. Link or file: _____

Check yourself

1. Does the world run smoothly on a phone and in a headset?

2. Is every source credited on screen and in the write-up?

3. Can a stranger understand what and where it is in ten seconds?

PRACTICE QUIZ

Check your module

Circle one answer for each question. The answer key is on the next page.

1. Roughly how long can a frame take at 90 Hz?

(A) About 33 ms

(B) About 16.7 ms

(C) About 11 ms

2. In a WebXR scene, how should you move the person through the world?

(A) Set the headset camera position directly every frame

(B) Put the camera in a rig Group and move the rig

(C) Reload the page at a new position

3. Which tool draws hundreds of identical streetlights in one draw call?

(A) THREE.InstancedMesh

(B) THREE.LOD

(C) A bigger shadow map

4. Why does WebXR need HTTPS (or localhost)?

(A) It is a powerful API, so browsers only allow it in a secure context

(B) HTTPS makes models load faster

(C) Quest browsers block all HTTP pages

ANSWER KEY

Answers and why

1 · C About 11 ms

1000 ms ÷ 90 ≈ 11.1 ms, and the headset renders two eyes in that time.

2 · B Put the camera in a rig Group and move the rig

The headset owns the camera pose. Moving a parent rig moves the person without fighting tracking.

3 · A THREE.InstancedMesh

InstancedMesh renders many copies of one geometry and material in a single call.

4 · A It is a powerful API, so browsers only allow it in a secure context

WebXR, like camera and location APIs, only runs in secure contexts.

NOTES

Build log and notes

SESSION	WHAT I MADE OR LEARNED	NEXT STEP