

## Лабораторна робота №9. AR-інтерфейс та анімація: робота зі скелетною анімацією (GLTF) та просторовими інтерфейсами

### 1. Мета роботи

- Навчитися імпортувати та керувати анімованими 3D-моделями у форматі GLTF/GLB у середовищі Three.js.
- Опанувати роботу з AnimationMixer та системою станів анімації.
- Розробити просторовий користувацький інтерфейс (**Spatial/Floating UI**) для взаємодії з об'єктами в AR.
- Вивчити методи оптимізації 3D-контенту для мобільних AR-пристроїв.

### 2. Теоретична частина

#### Скелетна анімація (Skinning & Rigging)

Скелетна анімація базується на двох процесах:

1. **Rigging (Ригінг):** Створення ієрархічної структури «кісток» (Bones), які утворюють скелет моделі.
2. **Skinning (Скіннінг):** Прив'язка вершин (vertices) геометрії моделі до конкретних кісток. Кожна вершина має «вагу» — коефіцієнт впливу певної кістки на її рух.

#### Анімаційна система Three.js

Для відтворення анімації використовуються три основні класи:

- **AnimationClip:** Містить дані про анімацію (назва, тривалість, треки трансформацій). Це «плівка» з записом.
- **AnimationMixer:** Об'єкт, що програє анімації. Він пов'язує кліпи з конкретною моделлю.
- **AnimationAction:** Надає методи керування конкретним кліпом: `play()`, `stop()`, `pause()`, а також налаштування ваги (`weight`) та циклічності.

#### Просторові інтерфейси (Spatial UI)

В AR/VR традиційні 2D-інтерфейси (HUD), закріплені на екрані, часто викликають дискомфорт. **Diegetic UI** — це елементи інтерфейсу, що існують всередині ігрового світу.

- **CanvasTexture:** Метод малювання інтерфейсу на HTML5 Canvas з подальшою передачею його як текстури на 3D-площину (`PlaneGeometry`).

- **HTMLMesh:** Бібліотека для рендерингу живого HTML/CSS коду в WebGL (експериментально, але потужно).

### 3. Практичне завдання: Проєкт «Інтерактивний AR-персонаж»

#### Етапи виконання:

1. **Підготовка:** Використовуйте Vite для швидкого розгортання проєкту.
2. **Завантаження моделі:** Використовуйте GLTFLoader. Модель має містити скелет та кілька анімацій.
3. **Логіка Mixer:** Створіть екземпляр AnimationMixer та додайте `mixer.update(delta)` у головний цикл рендерингу.
4. **UI Панель:** Створіть 3D-площину, яка буде слугувати меню. Нанесіть на неї кнопки «Idle», «Walk», «Wave».
5. **Raycasting:** Реалізуйте вибір кнопок за допомогою променя (Raycaster), що виходить з центру екрана або контролера AR.

#### 4. Завдання для самостійного виконання

1. **Crossfade:** Замініть миттєве перемикання анімацій на плавний перехід за допомогою `action.fadeOut(duration)` та `action.fadeIn(duration)`.
2. **Billboarding:** Напишіть функцію, яка у кожному кадрі розвертає UI-панель обличчям до камери (`panel.lookAt(camera.position)`).
3. **Творче завдання:** Додайте 3D-слайдер (або дві кнопки +/-), які змінюють значення `mixer.timeScale`, регулюючи швидкість анімації.

#### 5. Поради щодо оптимізації для Mobile AR

- **Кількість кісток:** Для стабільної роботи на мобільних браузерях намагайтеся не перевищувати 64-72 кістки на одну модель.
- **Геометрія:** Оптимальна кількість полігонів для AR-персонажа — до 10,000–15,000 трикутників.
- **Матеріали:** Використовуйте один атлас текстур замість багатьох окремих файлів, щоб зменшити кількість Draw Calls.

#### 6. Контрольні запитання

1. У чому перевага формату GLB над іншими форматами при роботі з анімацією у вебі?

2. Навіщо потрібно викликати `mixer.update(deltaTime)` у циклі рендерингу?
3. Як реалізувати одночасне програвання двох анімацій (наприклад, махання рукою під час ходьби)?
4. Які існують обмеження при використанні HTML/CSS для створення інтерфейсів усередині WebXR сесії?