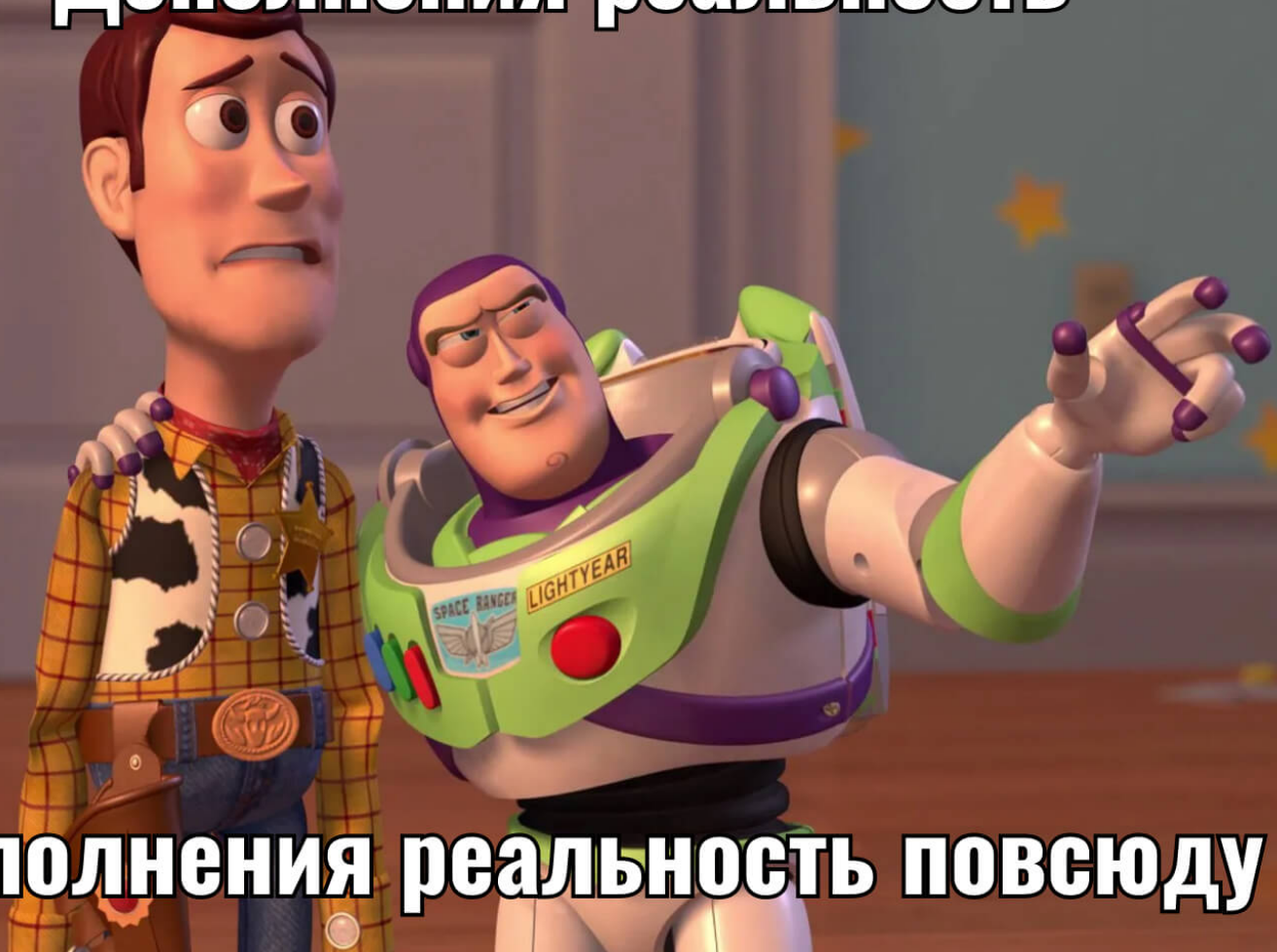


Дополнения реальность

Дополнения реальность повсюду



Презентация / гайд по AR

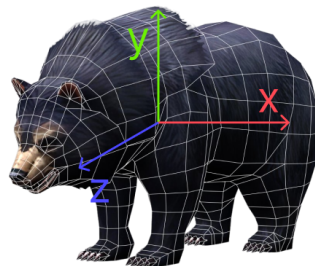
Попробую объяснить что такое AR, что он может, как его использовать и т.д. Первая половина будет полезна для дизайнеров и менеджеров, остальное уже техническая часть

- Что такое AR
- mind-ar-js
- Zap AR
- AR SDK for Three.js

Что такое AR

Эффект дополненной реальности достигается с помощью наложения 2х изображений.

- Формируется кадр из камеры устройства
- Затем рендерится кадр с 3d сценой
- Потом на кадр из камеры накладывается кадр из 3d сцены





Магия дополнений реальности

Как это работает

Для создания ощущения того что наша сцена находится в реальном мире её нужно правильно позиционировать. И по сути AR это инструмент который умеет ориентироваться в пространстве и исходя из этого задаёт координаты сцене.

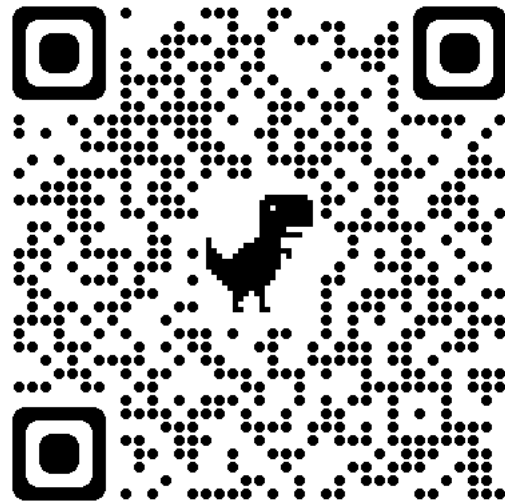
Как AR ориентируется

-  С помощью железа. Это доступ к специальной железяке в телефоне которая сама обрабатывает все данные с гироскопов, компасов, лидеров, камер и выдаёт нормальные координаты.
-  С помощью софта. Пытается получить данные к гироскопу, компасу, геолокации, камере и т.д.

С помощью железа

Мифический WebXR

Мы работаем с web а она работает через браузеры. И вот для этих браузеров создали восхитительный WebXR Device API который решить почти все наши проблемы с AR. За счет того что он получает все данные напрямую с устройства качество позиционирования очень хорошая, плюс производительность.



Но тут ключевое слово браузер и 🦕 поддержка данного API есть только на Chrome и только в Android



Когданибудь но не сейчас(((

С помощью софта

По сути есть 1.5 инструмента которые нормально работают

Это программа на js которая так же пытается получить все возможные данные с гироскопа, компаса и т.д. И за счёт это пытается сориентироваться в пространств.

✅ Самый большой плюс в том что это работает на js и мы не привязаны к браузеру. Это позволяет ей работать почти на любом устройстве.

❌ Из минусов качество позиционирования заметно хуже чем в WebXR, затрачивает приличное количество вычислительных мощностей из за того что это всё крутится на js. Ну и таких программ очень мало а самому их написать практически не реально.

- mind-ar-js
- Zap AR

mind-ar-js

Пока что единственный найденый бесплатный AR который в принципе работает но это скорее всего не кому не понравится.
Есть не плохая документация

Не плохой набор инструментов

- Image Tracking (Отслеживает определённые изображения и располагает сцену на нём)
- Face Tracking (Отслеживает лицо и на нём размещает сцену)

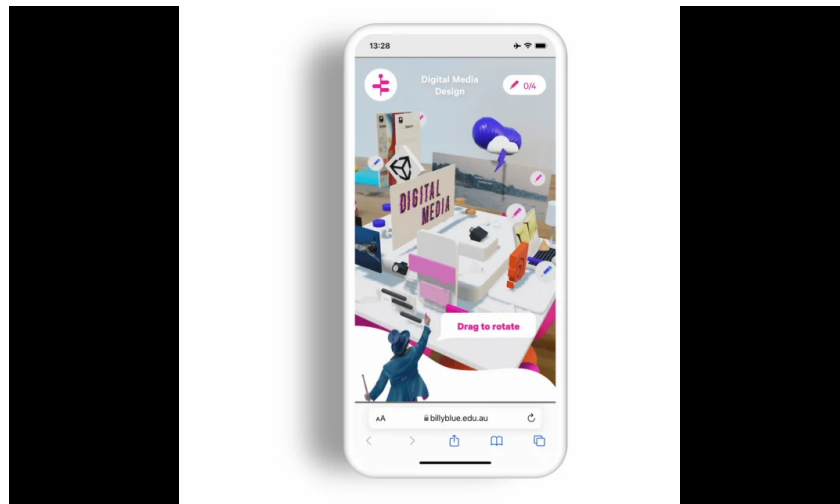
Поскольку его разрабатывает 1 чувак, работает это всё ну так себе, имеются куча ошибок.
Использовать его вы скорее всего не будете, но знать что такое есть стоит)



Zap AR

А вот это то с чем вы будете много работать)

Это платный инструмент, достаточно хорошо работающий. Имеет не плохой функционал и в принципе закрывает 100% задач которые есть у нас. Но самый большой минус он платный.



Функционал

- World tracking (Сцену можно поставить в любом месте и она будет привязана к точке в пространстве)
- Image Tracking (Отслеживает определенное изображения и располагает сцену на нём)
- Face Tracking (Отслеживает лицо и на нём размещает сцену)

Zapworks Studio

Сам не пробовал, пацаны рассказывали)

Так же в зар есть специальная программа с интерфейсом в который можно вполне полноценно работать. Как вариант особо упоротые дизайнеры смогут полноценно создавать ar, с возможностью полностью настроить 3d сцену, свет и выпиливать всякий функционал.

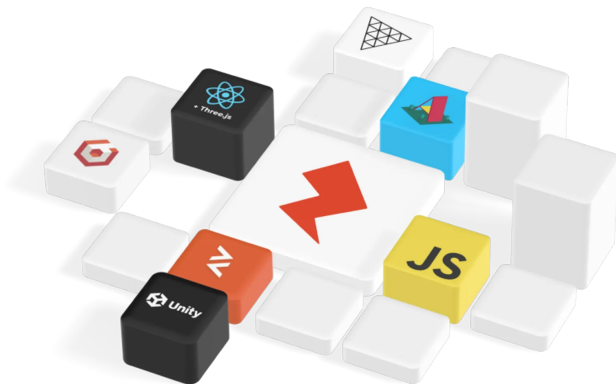


Universal AR SDK

Истинный путь самурая

Программы и интересы это кончено здоров, но истина power это создавать всё на js. AR SDK это набор js библиотек с помощью которых можно взаимодействовать с 3d сценой с помощью разных WebGL фреймворков.

Как говорилось раньше Zap AR это всего лишь инструмент для позиционирования сцены в пространстве. А уже рендер сцены, взаимодействие, логика, интерфейсы это всё делаем мы сами. Хотя Zap и предоставляет некоторые возможности но в основном всё придётся делать руками.



AR SDK for Three.js

Вы можете выбрать любой фреймворк который вам захочется, но я расскажу только о Three.js

В чём плюс Three.js Во первых это большая и хорошая документация в которой вы всё рано ничего не поймёте потому что мир WebGL примерно, полностью отличается от привычной web разработки). А так же у Three.js огромное комьюнити за счёт чего можно найти решения почти любого вопроса. Ну и отличная поддержка фреймворка и развитие.

- Документация для SDK Three.js
- Документация Three.js
- Примеры проектов

Старт проекта

Самый простой и быстрый способ

Можно конечно и самому сделать свою сборку мечты накатить туда пакет `npm install --save @zappar/zappar-threejs` но в этом случае вы столкнётесь с огромным количеством проблем. Проще всего в примерах найти подходящий для задачи и уже на его основе делать то что нужно

Возьмём к примеру glTF Loader

Тут у нас ссылка на гит хаб где находится полностью готовая сборка с примером работы SDK на Three.js. Её можно без препятственно скачать, установить модули `npm i` и пользоваться сколько влезет, но только локально. Всё это гавно зачем то работает на parcel.js что тоже может вызвать ряд проблем. Но к сожалению это самый удобный вариант на сегодня.

Структура проекта

Не должна у вас вызвать вопросов, но если вызывает то у меня для вас плохие новость))

Самым не понятным для вас тут будет файл /src/index.ts И я бы конечно мог описать что в нём происходит, но, но в этом нет смысла если вы не знаете Three.js А если знаете то и так всё будет понятно))

И тут мы приходим к самому неожиданному, что бы делать AR нужно разбираться в Three.js Потому что из Zap вы скопируете и вставите 10 строк кода и забудете о нём. А вот остальные 70-85% кода и мучений будут связаны с Three.js

Zap AR 2-5%



Вывод, учите Three.js