

# Технологии дополненной реальности

2022/23 учебный год

## Инженерный тур

### Общая информация

В заключительном этапе инженерного тура командам предстоит реализовать информационно-справочный веб-ресурс «Сумма технологий», посвященный WebAR-технологиям и включающий в себя:

- базовые компоненты AR-технологии, которые необходимы для реализации комплексной инженерной задачи на базе webAR;
- ссылки на все мультимедиа материалы, подготовленные командой (3D-модели, маркеры и доп материалы, демонстрационные видео работы всех реализуемых на примерах использования WebAR)
- демо-страницы с AR-сценариями (коды сценариев с комментариями), реализованными за время проведения инженерного тура заключительного этапа;
- игру, реализованную по сценарию разработчиков (с графическими материалами, видео-демонстрацией, кодом с комментариями);
- игру, созданную по собственному сценарию (с графическими материалами, видео-демонстрацией, кодом с комментариями).

### Легенда задачи

Компания «Альтернативное будущее» объявила конкурс на лучший образовательный интернет-ресурс «Сумма технологий», посвященный webAR разработке. На ресурсе необходимо разместить три модуля материалов, состоящих из примеров кода, видео-демонстраций работы, возможности проверить работоспособность примера опытным путем.

Первый модуль посвящен реализации отдельных базовых элементов webAR-разработки — от прикрепления к маркеру мультимедийных объектов, до программных компонентов реализующих взаимодействие с объектами дополненной реальности. В первом модуле, согласно правилам конкурса, компания требует выполнить трехмерные модели четко по представленному техническому заданию.

Во втором модуле компания просит команды участников конкурса подготовить игру по сценарию. Предлагается реализовать взаимодействие с объектами цифрового мира кубитоклобусами через конвейер. Конвейер также, как и существа кубитоклобусы, является цифровой моделью, управляется при помощи различных маркеров, может модифицировать объекты и выполнять их сортировку. Игра объединяет в одном приложении ранее описанные отдельные элементы, сценарий демонстрирует использование разных функциональных возможностей дополненной реальности в веб-пространстве.

Третий модуль материалов веб-ресурса — игра, предложенная участниками по собственному сценарию, или модификация ранее реализованной игры «Кубитоклобусы»

## Требования к команде и компетенциям участников

Количество участников в команде: 4.

Компетенции, которыми должны обладать члены команды:

(Роли, которые должны быть представлены в команде)

1. **Архитектор:** разработка архитектуры веб-ресурса, создание из отдельных AR-модулей единого программного продукта.
2. **Конструктор (дизайнер):** разработка маркеров, конструирование трехмерных моделей и анимации, подготовка дизайн-макета веб-ресурса, содержащего объекты дополненной реальности.
3. **Программист(ы):** создание отдельных программных компонентов — AR-модулей, разработка сценариев взаимодействия пользователя с объектами дополненной реальности, а также с веб-ресурсом.

## Оборудование и программное обеспечение

Наименование	Описание
<b>Компьютер или ноутбук</b> (Базовые характеристики Operating system version Windows: Windows 10; CPU: x86, x64 architecture with SSE2 instruction set support; Graphics API: DX10, DX11, DX12 capable)	Для разработки программных компонентов
<b>Планшет или смартфон</b> (Базовые характеристики Operating system: Android7, CPU: ARM 64-bit, Graphics API: OpenGL ES)	Для тестирования разработанных AR-сценариев и просмотра веб-ресурса
<b>Редактор кода или IDE</b> Например: VS Code ( <a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a> ) Sublime text ( <a href="https://www.sublimetext.com/">https://www.sublimetext.com/</a> ) или другие аналоги	Для разработки программных компонентов и их отладки Для развертки локального сервера
<b>Графический редактор</b> Например: Adobe Photoshop, Adobe Illustrator, Figma ( <a href="https://www.figma.com/">https://www.figma.com/</a> ), Procreate, Gimp, Paint или другие аналоги	Для создания маркеров, макета веб-ресурса, 2D материалов оверлея и сайта
<b>3D редактор</b> Например: Blender ( <a href="https://www.blender.org/">https://www.blender.org/</a> ) или другие аналоги	Для создания трехмерных моделей
<b>Интернет-браузер</b> Например: Яндекс Браузер ( <a href="https://browser.yandex.ru/">https://browser.yandex.ru/</a> ) или другие аналоги	Для просмотра и отладки решений
<b>Дополнительно:</b>	
Видеокамера, наушники, компьютерная мышь	Для удобства в работе и тестировании
Доступ в Интернет со скоростью около 100 Мбит/с.	Для тестирования и просмотра результатов работы, при разворачивании сервисов в сети Интернет

---

## Описание задачи

Инженерный тур разбит на 3 части:

- Часть 1. Секретный ингредиент.
- Часть 2. Конструкция по инструкции.
- Часть 3. Своя игра. Сборка итогового проекта.

### *Этап 1. Секретный ингредиент*

Решение задач повышенной сложности на реализацию отдельных элементов Web-AR технологий и моделирование трехмерных объектов с анимацией.

Этап дополнительно разбит на две части:

- Задания: оверлей AR-модулей.
- Задания: архитектура AR-модулей.

Эта часть задания заключительного этапа выполняется на платформе Степик и имеет автоматическую проверку и оценивание. В спорных ситуациях решения команд рассматривается разработчиками.

### *Этап 2. Конструкция по инструкции*

Задача на реализацию Web-AR проекта по сценарию разработчиков с использованием программных компонентов, реализованных в первый день соревнования.

Эта часть задания оценивается разработчиками на соответствие функциональным возможностям, описанным в сценарии игры с учетом дополнительных решений участников в области механики игры и дизайне интерфейсов взаимодействия с объектами дополненной реальности.

### *Этап 3. Своя игра. Сборка итогового проекта*

Реализация собственного сценария простейшей игры с использованием технологий дополненной реальности. Доработка итогового продукта в виде информационно-справочного веб-ресурса «Сумма технологий», предназначенного для обучения работе с WebAR-технологиями.

Результаты выполнения этой части задания заключительного этапа оценивается экспертным жюри вместе с итоговым проектом веб-ресурсом «Сумма технологий».

## Система оценивания

Оценка задачи финала профиля «Технологии виртуальной и дополненной реальности: дополненная реальность» делится на четыре составляющих:

- Этап 1. Секретный ингредиент.
- Этап 2. Конструкция по инструкции.
- Этап 3. Своя игра и итоговый проект.

- Оценка жюри.

Все критерии к данным блокам представлены в таблицах VI.2.1–VI.2.4.

Таблица VI.2.1: Критерии оценки Этапа 1

Название модуля и задач	Критерий	Макс. балл	Примечание
<b>1.1 Задания: оверлей AR-модулей</b>		<b>40</b>	
Задача. Разработка маркеров команды		3	В данном задании оценивается максимум 6 маркеров
	Устойчивый трекинг собственного маркера	по 0,5 за маркер	
Задача. Анимация в 3D		12	Задание с автоматической проверкой на соответствие эталону
Задача. Логотип в 3D		5	Задание с автоматической проверкой на соответствие эталону
Задача. Цифровые аватары команды		12	В данном задании оценивается максимум четыре 3D-модели
	Наличие корректной анимации	по 1 за модель	
	Наличие текстур/материалов на модели и ее частях	по 1 за модель	
	Корректный риг	по 1 за модель	
Задача. Воссоздание статического объекта		8	Задание с автоматической проверкой на соответствие эталону
<b>1.2 Задания: архитектура AR-модулей</b>		<b>60</b>	
Задача. Трехмерные фигуры на гранях куба		3	Задание с автоматической проверкой
Задача. Объекты внутри куба		4	Задание с автоматической проверкой
Задача. Множество динамических объектов на одном маркере		4	Задание с автоматической проверкой
Задача. Прыгающий мяч (анимация)		4	Задание с автоматической проверкой
Задача. Мяч прыгающий (компонент)		3	Задание с автоматической проверкой
Задача. Пин-понг		6	Задание оценивает командой разработчиков, с применением дисконтирования*
Задача. Цилиндрическая дырка в полу		3	Задание с автоматической проверкой
Задача. Портал в виртуальный мир		7	Задание оценивает командой разработчиков, с применением дисконтирования*
Задача. Управление виртуальным объектом при помощи клавиатуры		6	Задание оценивает командой разработчиков, с применением дисконтирования*
Задача 0. Тень с секретом		2	Задание с автоматической проверкой
Задача 1. 3D фигура отдельными точками		5	Задание оценивает командой разработчиков, с применением дисконтирования*
Задача 2. Управление виртуальной камерой при помощи клавиатуры		3	Задание с автоматической проверкой
Задача 3. Мультимаркер и его взаимодействие		10	Задание оценивает командой разработчиков, с применением дисконтирования*



\*Дисконтирование применяется:

- на количество попыток в получении рабочего кода задачи, с учетом указанного в условии задачи максимального количества попыток;
- отсутствие комментариев в представленном коде;
- представленное решение удовлетворяет лишь части условий описанных в задании.

Таблица VI.2.2: Критерии оценки Этапа 2

<b>Этап 2. Конструкция по инструкции</b>	<b>100</b>
<b>Составляющие базовой реализации</b>	<b>30</b>
Стартовый маркер — на котором генерируется кубитоклобус	10
Маркер цвета для направления кубитоклобуса в приемник соответствующего цвета	10
Счетчик отправленных кубитоклобусов	10
<b>Составляющие продвинутой реализации</b>	<b>15</b>
Добавление формы кубитоклобусам	5
Маркер распределения кубитоклобусов по форме	5
При попадании кубитоклобуса формы несоответствующей приемнику — игра заканчивается. При этом, если цвет не верный, то игроку просто не начисляются баллы.	5
<b>Реализация Некубитоклобусов</b>	<b>20</b>
Маркер для конвертирования кубитоклобуса в нужную форму и(или) цвет	8
Маркер уничтожения некубитоклобуса.	5
Счетчик жизней работающего конвейера (При неверной форме — 1 жизнь, всего жизней 5, присутствует отображение количества жизней пользователю)	5
Прекращение игры при попадании некубитоклобуса в приемник	2
Дополнительные «фишки» для любого уровня реализации	35
Ускорение скорости появления кубитоклобусов со временем	5
Ограничение процесса игры по времени от 1 до 2 минут	5
Вывод счета по окончании игры	5
Авторская стилизация маркеров, фигур и интерфейса	10
Добавление фоновой музыки (2 балла), музыкальных эффектов (3 балла)	5
Счетчик жизней — отображение пользователю на экране и дополнительные условия по их потери	5

Таблица VI.2.3: Критерии оценки Этапа 3

Название задачи и критериев	Макс. балл
<b>Этап 3. Своя игра и итоговый проект</b>	<b>160<sup>1</sup></b>
<b>1. Своя задача</b>	<b>60 баллов</b>
Концепция	25
Механика	10
Полнота решения	25
<b>2. Сайт</b>	<b>100 баллов</b>
Наличие всех задач финала (в том числе задач 2 и 3 дня)	по 1 за задачу
Наличие в задачах (по штучно)	
Кодов решения	1
Видео или изображения демонстрации работы сценария	3 — видео 1 — изображение
Наличие блока или ссылки на демо-страницу	1
Наличие ссылки на материалы	0,5
Присутствует название команды на сайте	1

Таблица VI.2.3: Критерии оценки Этапа 3

Название задачи и критериев	Макс. балл
Технологичность решения	2 — конструктор сайтов 6 — самостоят. разработка
Адаптивность	3

Таблица VI.2.4: Критерии оценки жюри

Критерии оценки жюри	Максимум 50 баллов
Общее впечатление	
Внешний вид сайта	15
Игр(ы)	15
Техника презентации проекта	9
Ответы на вопросы	9
Дополнительные баллы	2

Общая оценка решения командного тура формируется по формуле:

$$0,5 \cdot \text{Баллы Этап 1} + 0,2 \cdot \text{Баллы Этап 2} + 0,2 \cdot \text{Баллы Этап 3} + 0,1 \cdot \text{Баллы за презентацию проекта жюри.}$$

При верном (задание проверяется на соответствие требованиям), полном (выполнены все пункты задания) и качественном (экспертная критериальная оценка) выполнении всех заданий профиля участники могли набрать 100 баллов.

Важно! Оценка за выполнение задач нормируется к 100. В том случае, если ни одна команда участников заключительного этапа не набирает максимально возможного балла, к 100 нормируется результат лучшей команды, соответствующим образом пересчитываются результаты остальных команд.

## Решение задачи

### *Этап 1. Секретный ингредиент. Задания: оверлей AR-модулей*

#### *Задача VI.2.7.1. Разработка маркеров команды*

Разработайте шесть уникальных стилизованных под команду маркеров. Дайте названия маркерам: `pattern_1`, `pattern_2` и т. д.

Маркеры необходимо вставить в шаблон кода, приведенный ниже, для проверки устойчивости контента, привязанного к ним.

Проверьте, что контент не путается между маркерами, над каждым маркером выводится свой объект.

Шаблон кода.

```
<!DOCTYPE html>
<html>
<head>
```

<sup>1</sup>Возможно получение большего количества баллов, нормируется по лучшему результату к 100

---

```
<script src="https://aframe.io/releases/1.4.1/aframe.min.js"></script>
<script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>
<script src="https://cdn.jsdelivr.net/gh/c-frame/aframe-extras@6.1.1/dist/aframe-extras.min.js"></script>
</head>
<body>
  <a-assets>
    
  </a-assets>

  <a-scene>

    <a-marker id="marker_1" type="pattern"
      ↪ url="./src/markers/patt/pattern_1.patt">
      <a-box material="color: #4CC3D9; transparent: true; opacity: 0.3; src: #grid;"></a-box>
    </a-marker>

    <a-marker id="marker_2" type="pattern"
      ↪ url="./src/markers/patt/pattern_2.patt">
      <a-box material="color: #FFCC00; transparent: true; opacity: 0.3; src: #grid;"></a-box>
    </a-marker>

    <a-marker id="marker_3" type="pattern"
      ↪ url="./src/markers/patt/pattern_3.patt">
      <a-box material="color: #FF2400; transparent: true; opacity: 0.3; src: #grid;"></a-box>
    </a-marker>

    <a-marker id="marker_4" type="pattern"
      ↪ url="./src/markers/patt/pattern_4.patt">
      <a-box material="color: #6A5ACD; transparent: true; opacity: 0.3; src: #grid;"></a-box>
    </a-marker>

    <a-marker id="marker_5" type="pattern"
      ↪ url="./src/markers/patt/pattern_5.patt">
      <a-box material="color: #3F888F; transparent: true; opacity: 0.3; src: #grid;"></a-box>
    </a-marker>

    <a-marker id="marker_6" type="pattern"
      ↪ url="./src/markers/patt/pattern_6.patt">
      <a-box material="color: #3D2B1F; transparent: true; opacity: 0.3; src: #grid;"></a-box>
    </a-marker>

    <a-entity camera></a-entity>
```

```
</a-scene>  
</body>  
</html>
```

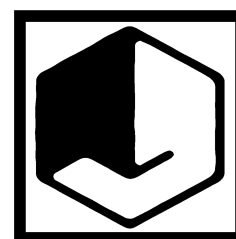
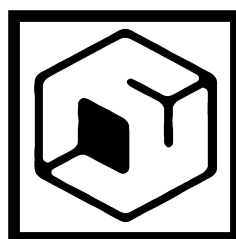
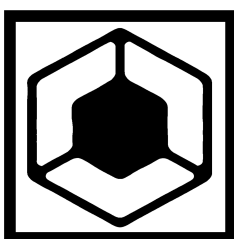
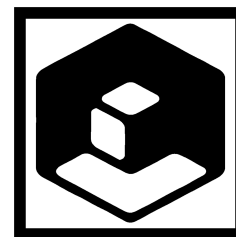
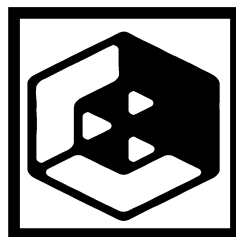
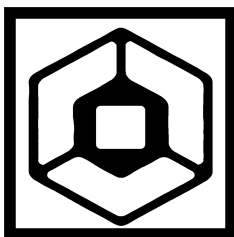
В результате необходимо прикрепить:

- ссылку на скриншот с результатами выполнения AR-сценария из шаблона, где будет видно сразу все 6 маркеров и AR-контент (пример на рисунке);
- ссылку на проект в glitch.



### *Решение*

Шесть маркеров, которые считываются как уникальные и контент не путается между ними.



## Задача VI.2.7.2. Анимация в 3D

Вам необходимо воссоздать анимацию, просмотрев видео: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/babr\\_animation.mp4](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/babr_animation.mp4).

Имея условия, описанные ниже.

- Всего в анимации 115 кадров, 116-й кадр дублирует 1-й кадр.
- В кадрах с 1-го по 49-й вы создаете walk cycle Бабра.
- В 60-м кадре он переходит в позицию покоя (rest pose).
- После чего на 75-м кадре встает в характерную особую позу.
- Со 105-го кадра до 116-го Бабр возвращается в исходную позицию ходьбы (1-й кадр).

При этом:

- Кадры ходьбы: 1, 8, 14, 20, 26, 32, 38, 44, 49.
- Кадры 1, 8, 14, 20 соответствуют первой половине walk cycle.
- Кадры 26, 32, 38, 44 соответствуют второй половине walk cycle.
- Кадр 49 повторяет кадр 1.
- Кадры 75, 85, 95 и 105 являются вариациями особой позы Бабра с небольшими изменениями.
- Кадр 116 повторяет кадр 1.

Внутри приложенного файла вы найдете готовую модель Бабра, соболя и их риг: <https://disk.yandex.ru/d/vpdqEcpwHoK5Gg>.

Также в файле уже настроены материалы, свет, свойства рендера и положение камеры. Не меняйте эти настройки, так как они напрямую влияют на результат. Transition между кадрами оставьте по умолчанию.

*Эталонные изображения*

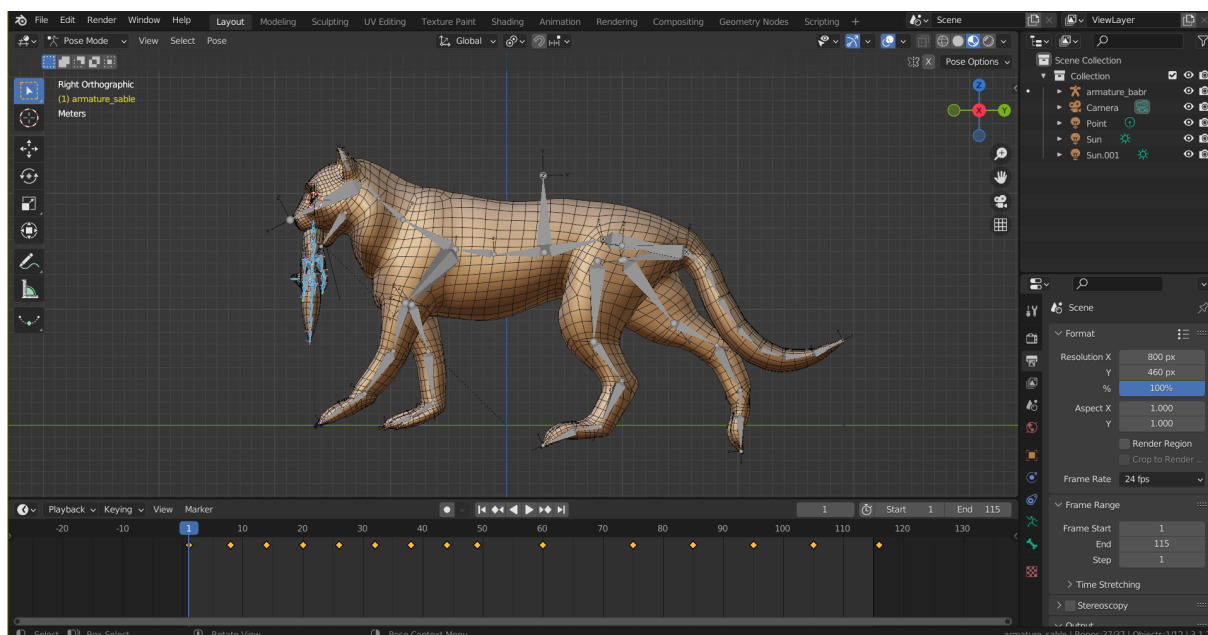


Рис. VI.2.1. Ключевые кадры соболя

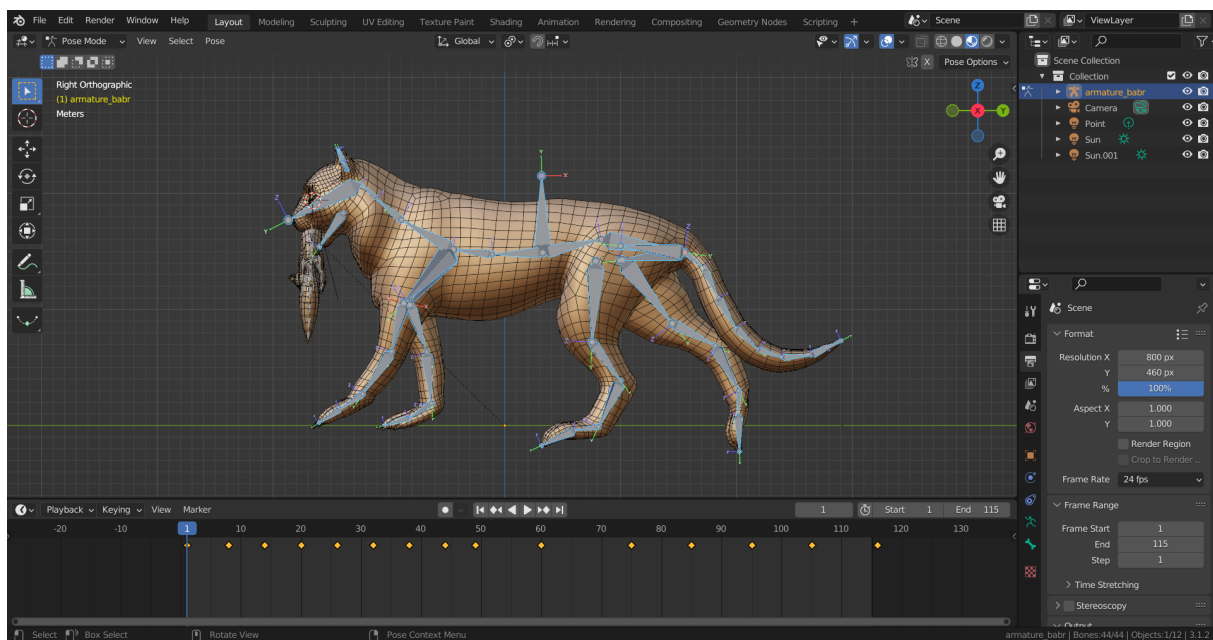


Рис. VI.2.2. Ключевые кадры бобра

### *Способ оценки работы*

Для проверки результата использовался следующий код на языке Python 3.

```
from skimage.metrics import structural_similarity as compare_ssim
import cv2

if __name__ == "__main__":
    imageA = cv2.imread("путь к оригинальному изображению")
    imageB = cv2.imread("путь к изображению из отклика")

    grayA = cv2.cvtColor(imageA, cv2.COLOR_BGR2GRAY)
    grayB = cv2.cvtColor(imageB, cv2.COLOR_BGR2GRAY)

    score = compare_ssim(grayA, grayB, full=True)[0] * 100
    if score > 95:
        print(True)
    else:
        print(False)
```

### *Решение*

Ссылка на эталонное решение: <https://disk.yandex.ru/d/980FAG0NozoL-Q>.

Ссылка на файл с эталоном в формате glb: <https://disk.yandex.ru/d/GWn20UgBf4L-6w>.

### *Задача VI.2.7.3. Логотип в 3D*

Создайте логотип НТО в 3D на основе предложенных референсов.

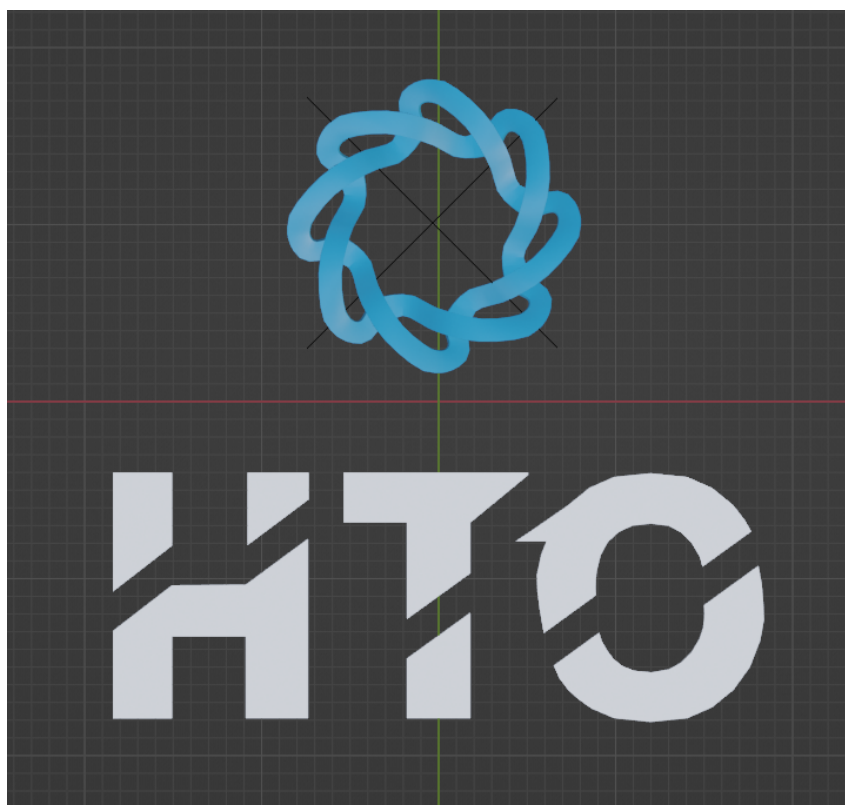


Рис. VI.2.3. Фронтальный вид

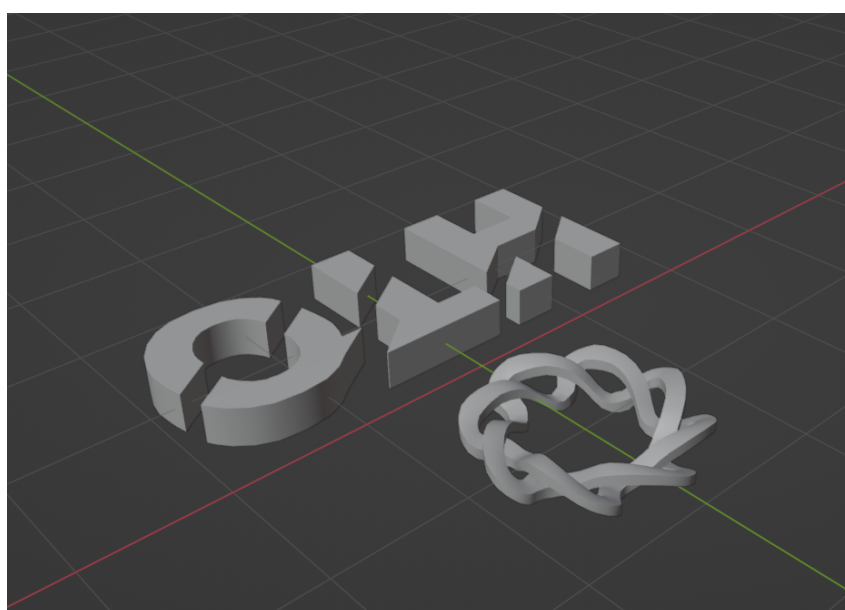


Рис. VI.2.4. Вид в проекции



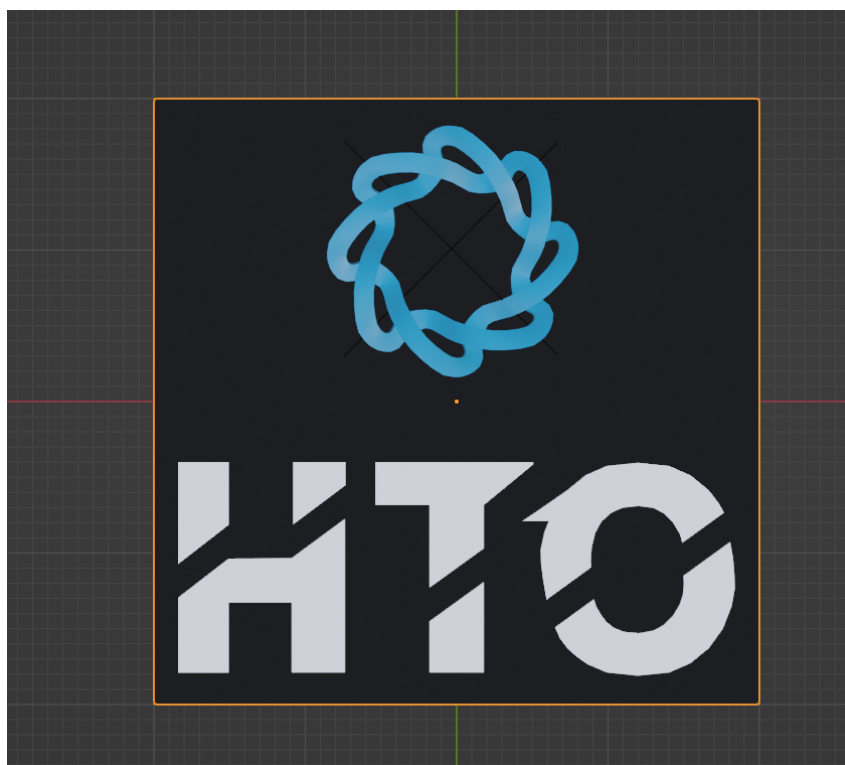


Рис. VI.2.5. Фронтальный вид с подложкой

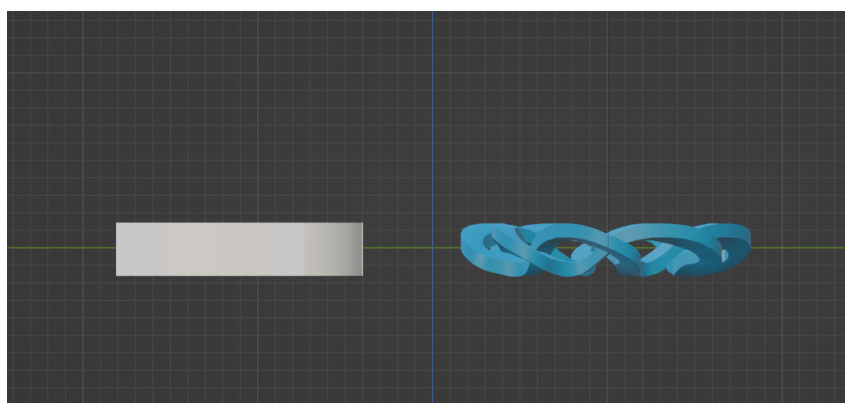


Рис. VI.2.6. Вид сбоку

Логотип располагается относительно центра осей. Задний план имеет размер  $4 \times 4$  юнита и добавлен на изображения для демонстрации масштаба логотипа.

Создайте свою собственную модель в приложенном файле: <https://disk.yandex.ru/d/Jer-Dya2VR-ZcA>.

В этом файле уже настроены материалы, свет, свойства рендера и положение камеры. Не меняйте эти настройки, так как они напрямую влияют на результат.

В отклик прикрепите рендер модели.

### *Способ оценки работы*

Для проверки результата использовался следующий код на языке Python 3.

```
from skimage.metrics import structural_similarity as compare_ssim
import cv2
```



```
if __name__ == "__main__":
    imageA = cv2.imread("путь к оригинальному изображению")
    imageB = cv2.imread("путь к изображению из отклика")

    grayA = cv2.cvtColor(imageA, cv2.COLOR_BGR2GRAY)
    grayB = cv2.cvtColor(imageB, cv2.COLOR_BGR2GRAY)

    score = compare_ssim(grayA, grayB, full=True)[0] * 100
    if score > 95:
        print(True)
    else:
        print(False)
```

Где `score` — регулируемая обсуждаемая величина. Для следующего пака изображений проверка выше проходит только для файла `002.png`, исходное изображение — `001.png`, остальные два изображения проверку по схожести не проходят. При величине `score > 94` проверку проходят все 3 изображения.

### *Решение*



Ссылка на файл с эталоном в формате glb: [https://disk.yandex.ru/d/pOzY4BuvHR\\_oPQ](https://disk.yandex.ru/d/pOzY4BuvHR_oPQ).

### *Задача VI.2.7.4. Цифровые аватары команды*

Разработайте цифровые аватары для всех участников вашей команды.

Цифровой аватар — трехмерная модель, копирующая наиболее значимые признаки внешности участника команды финалистов. При этом стилевое решение для оформления модели дизайнер может подобрать такое, чтобы подчеркнуть единство команды, а также идею концепции всего итогового проекта.

Для создания моделей мы рекомендуем использовать веб-конструктор Ready Player Me: <https://readyplayer.me/>.

---

Для формирования модели можно использовать фотографию, изображение с камеры или просто сделать похожее на участника персонажа на основе собственного опыта с использованием инструментов и заготовок конструктора. Для оживления модели используйте сервис Mixamo: <https://www.mixamo.com/?amp#/>.

Передайте готовые модели разработчикам для прикрепления их к граням командного AR-куба.

**О форматах:** Конструктор Ready Player Me экспортирует модель аватара в формате glb (GLB — это формат 3D-файлов, используемый в виртуальной реальности (VR), дополненной реальности (AR)). Сервис Mixamo принимает модель в формате fbx (файлы fbx часто используются в разработке игр и анимации). Готовые анимированные цифровые аватары будут нужны в формате glb. Используйте редактор Blender для смены форматов ваших моделей.

**О названиях:** Всем моделям необходимо дать название `person_K.glb`, где К это номер от 1 до 4.

Примечание: воспользуйтесь инструкцией, если не разберетесь в процессе создания аватара самостоятельно. Инструкция содержит видеофрагмент преобразования модели цифрового аватара с использованием программы Blender: [https://youtu.be/yHY\\_9Fr2sg0](https://youtu.be/yHY_9Fr2sg0).

В качестве ответа прикрепите ссылку на архив с аватарами.

### *Решение*

В качестве решения представлена одна модель аватара с анимацией.

Ссылка на файл glb: <https://disk.yandex.ru/d/hhFNVCSZ4Rj0sw>.

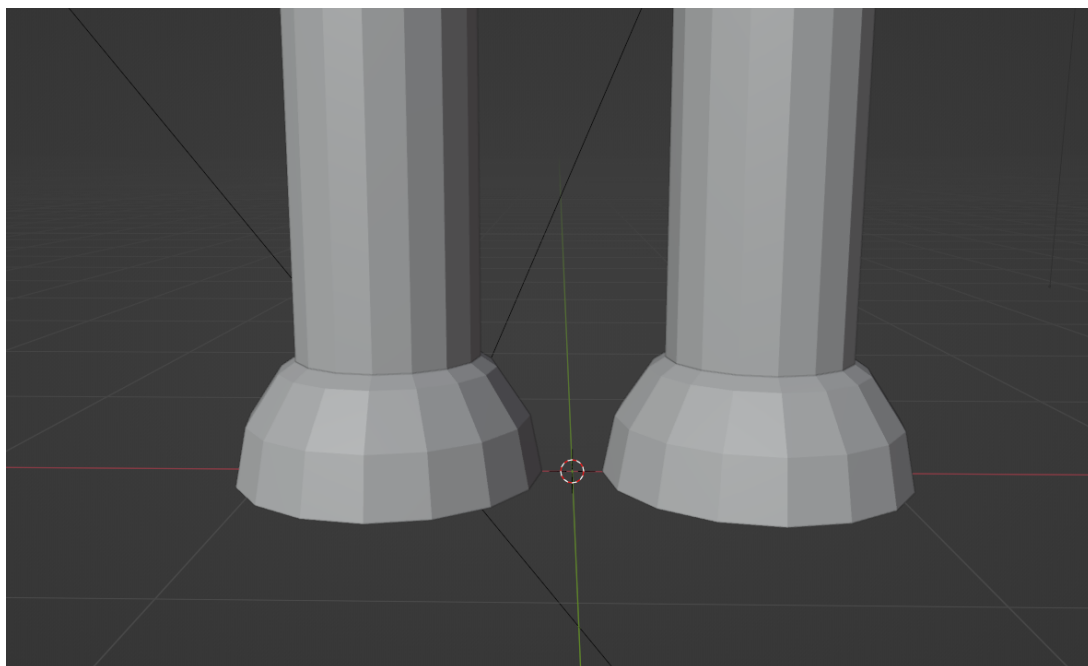


### *Задача VI.2.7.5. Воссоздание статического объекта*

Вам необходимо создать модель робота по образцу. Размеры робота представлены на анимированных изображениях ниже. У вас есть проекция вида спереди и вида сбоку для ориентации по деталям робота.

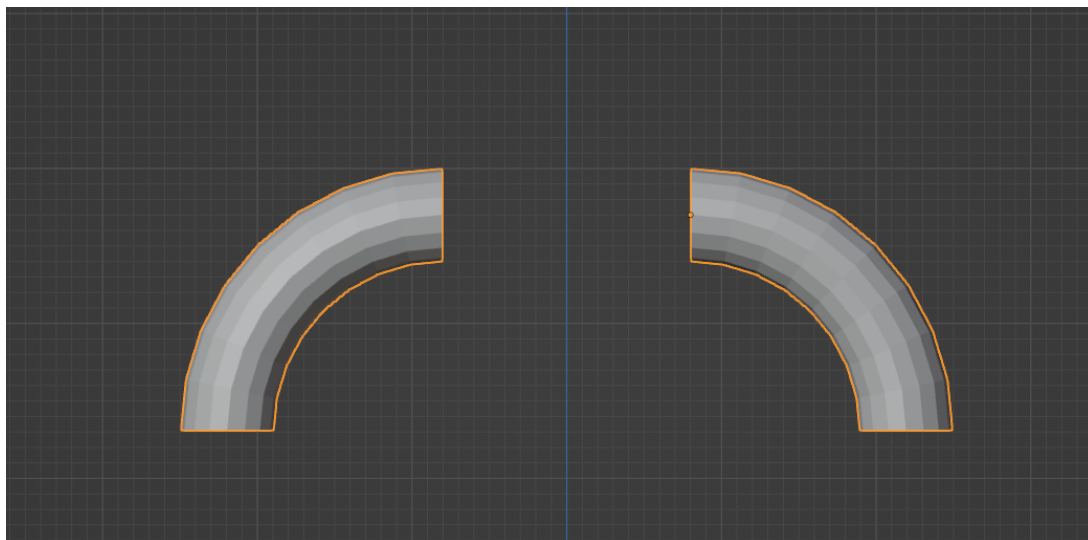
### Особенности модели

Все цилиндры и сферы имеют по 16 ребер. Модель находится основанием на поверхности осей, снизу на рисунке вы видите 3D-курсор, расположенный в точке  $(0, 0, 0)$  по центру нижней плоскости модели.



Для создания рук воспользуйтесь модификаторами **Solidify**, **Subdivision Surface** (simple, 3 шага) и **Simple Deform** (Bend, 90 градусов).

По необходимости пользуйтесь модификатором **Mirror**.



Создайте свою собственную модель в приложенном файле: В этом файле уже настроены свет, свойства рендера и положение камеры. Не меняйте эти настройки, так как они напрямую влияют на результат. Не добавляйте и не изменяйте материалы на модели.

Анимированное изображение с размерами деталей, вид спереди: <https://disk.yandex.ru/i/rIH68r0mqSEaSA>.

---

Анимированное изображение с размерами деталей, вид сбоку: <https://disk.yandex.ru/i/Z28-FotDikrZfg>.

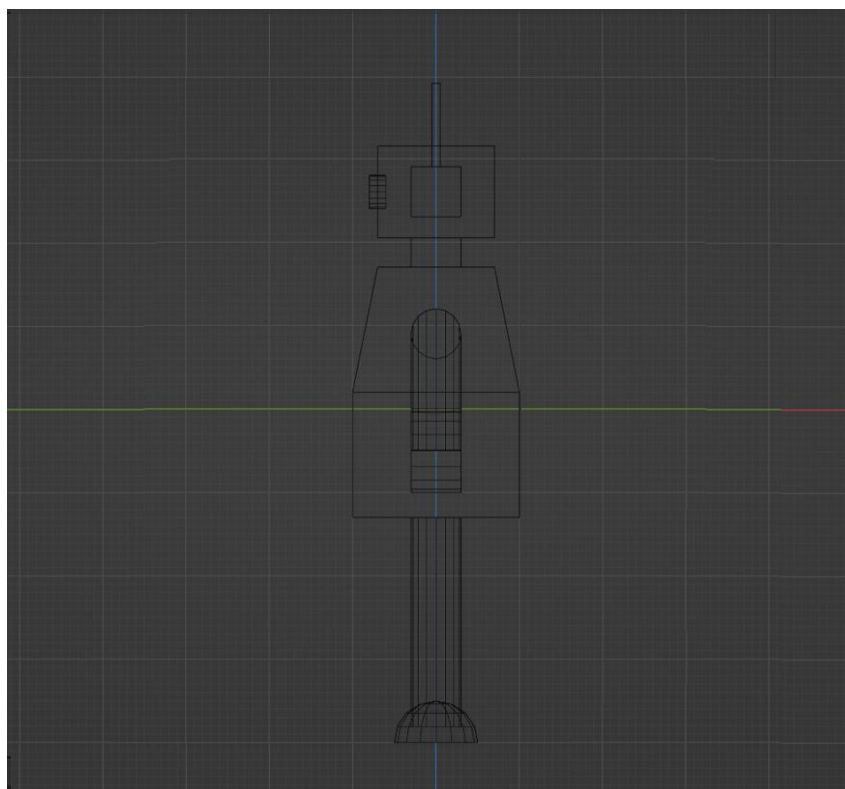


Рис. VI.2.7. Статичное изображение, вид сбоку, wireframe режим

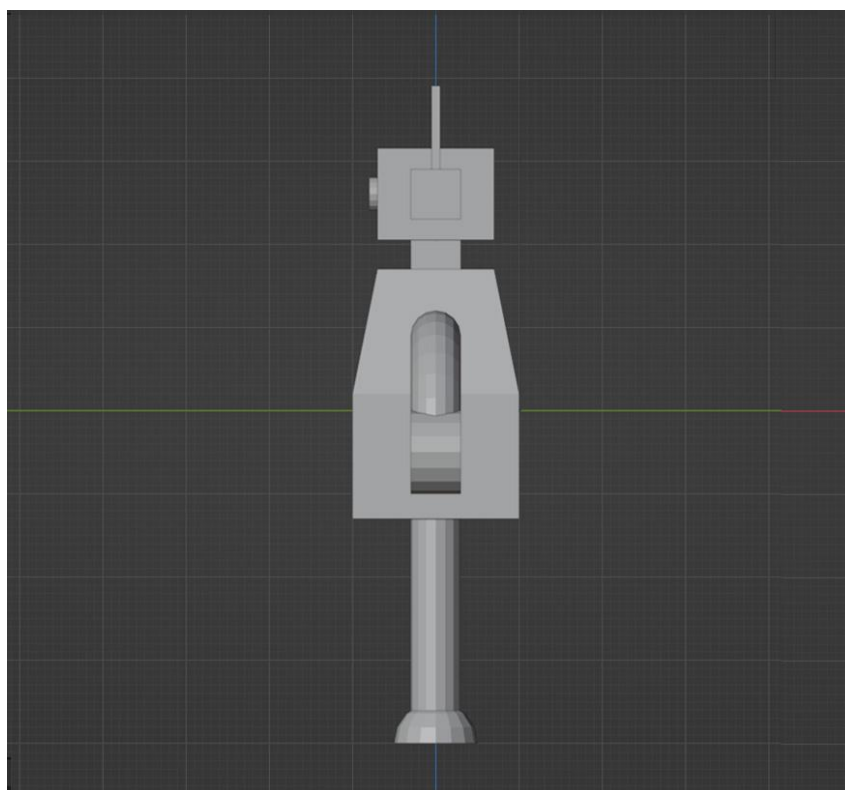


Рис. VI.2.8. Статичное изображение, вид сбоку, solid режим

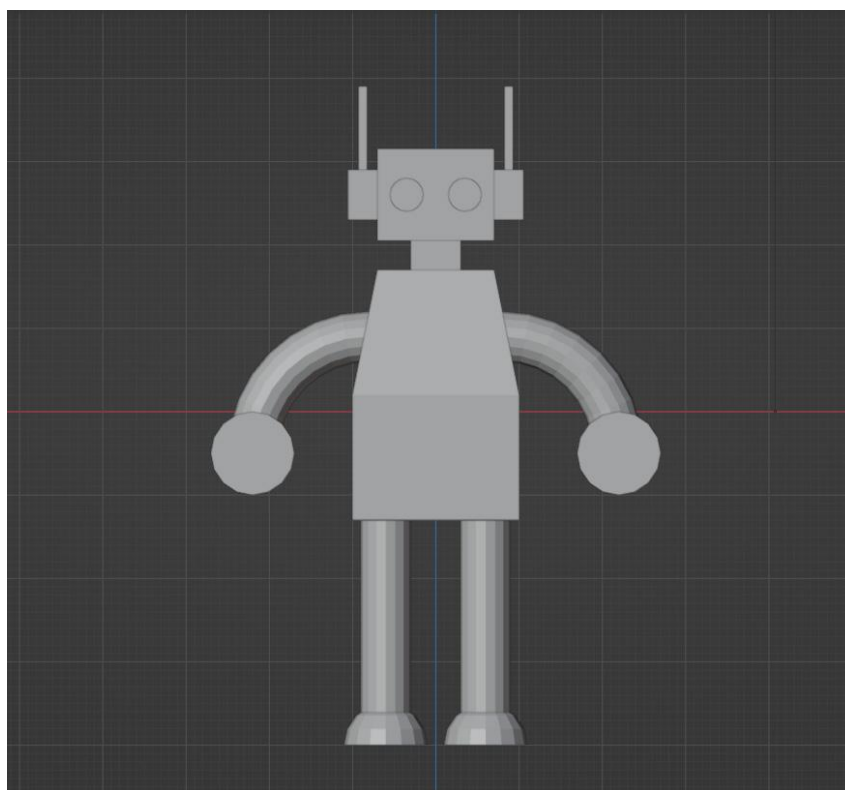


Рис. VI.2.9. Статичное изображение, вид спереди, solid режим

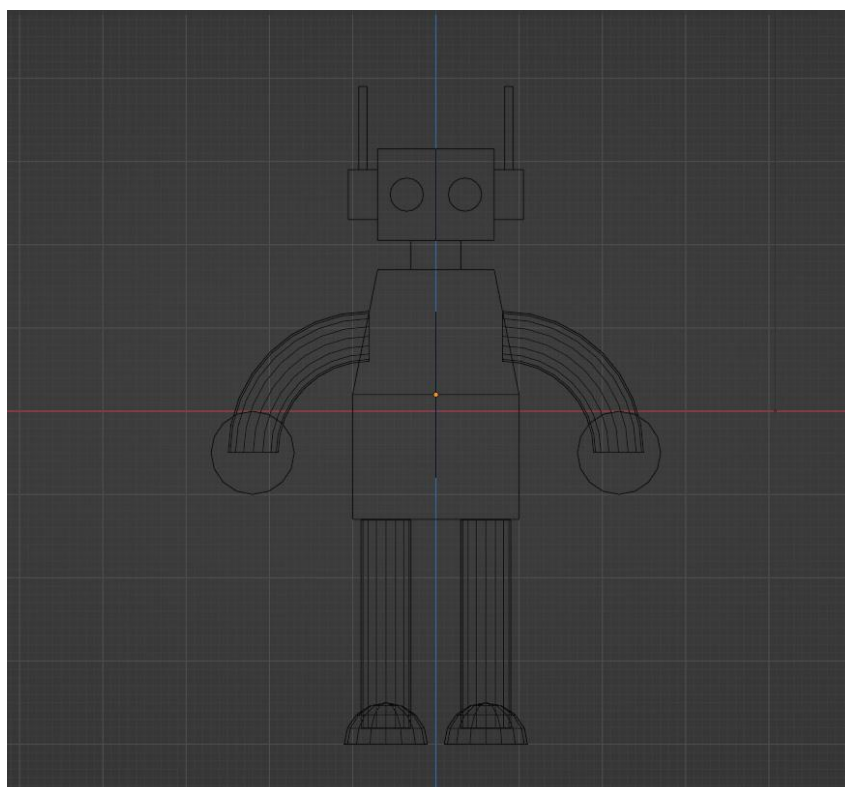


Рис. VI.2.10. Статичное изображение, вид спереди, wireframe режим

В отклик прикрепите рендер модели.

---

## Способ оценки работы

Для проверки результата использовался следующий код на языке Python 3.

```
from skimage.metrics import structural_similarity as compare_ssim
import cv2

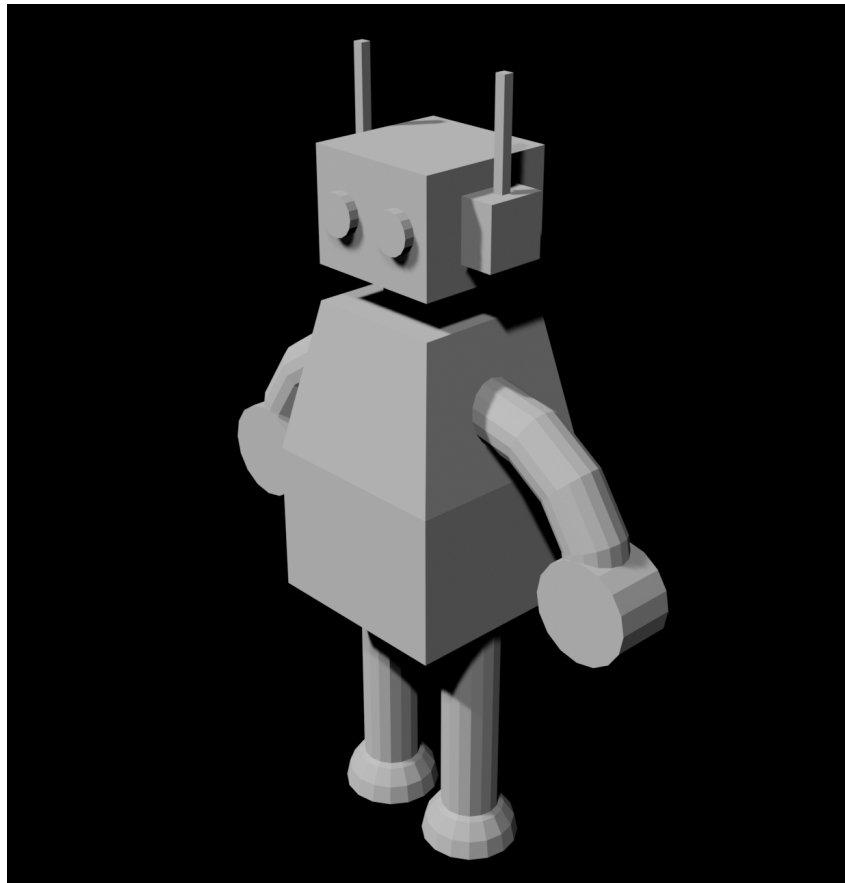
if __name__ == "__main__":
    imageA = cv2.imread("путь к оригинальному изображению")
    imageB = cv2.imread("путь к изображению из отклика")

    grayA = cv2.cvtColor(imageA, cv2.COLOR_BGR2GRAY)
    grayB = cv2.cvtColor(imageB, cv2.COLOR_BGR2GRAY)

    score = compare_ssim(grayA, grayB, full=True)[0] * 100
    if score > 95:
        print(True)
    else:
        print(False)
```

Где `score` — регулируемая обсуждаемая величина. Для следующего пака изображений проверка выше проходит только для файла `002.png`, исходное изображение — `001.png`, остальные два изображения проверку по схожести не проходят. При величине `score > 94` проверку проходят все 3 изображения.

## Решение



---

Ссылка на эталон в формате blend файла: <https://disk.yandex.ru/d/KGhBwrYLcwxxQg>.

## Этап 1. Секретный ингредиент. Задания: архитектура AR-модулей

### Задача VI.2.7.6. Трехмерные фигуры на гранях куба

Разработайте систему, которая позволит реализовать «эффект волшебного куба», когда объекты дополненной реальности накладывается на куб, покрытый шестью маркерами.

Ниже представлен шаблон кода, допишите фрагмент кода так, чтобы:

- на сторонах куба должны быть маркеры команды с названиями `pattern_1`, `pattern_2` и т. д.
- на маркере 1 должна появляться анимированная 3D фигура Бабра с названием «`babr_animate.gltf`» и `id = "babr"`
- маркер 2, 3, 4, 5 — анимированных 3D аватаров участников команд с названиями «`person_K.glb`» и `id="personK"`, где К это номер от 1 до 4.
- маркер 6 — 3D модель логотипа НТО с названием «`logo_nto.glb`» и `id="logo"`.

```
<html>
<script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
<script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe_
↪ me-ar.js"></script>
<body style="margin : 0px; overflow: hidden;">
  <a-scene arjs embedded>

    <!--ваш фрагмент кода-->

    <a-entity camera></a-entity>
  </a-scene>
</body>
</html>
```

В ответ напишите фрагмент кода.

Примечание: отступы и переносы строк могут быть в любом месте.

Примечание: для тестирования данной задачи воспользуйтесь любыми другими 3D моделями с названиями, указанными в условии к данной задаче, заменить их на нужные можно позже.

### Решение

```
1 <a-asset>
2   <a-asset-item id="babr" src="mod\babr_animate.gltf"></a-asset-item>
3   <a-asset-item id="person1" src="mod\person_1.glb"></a-asset-item>
4   <a-asset-item id="person2" src="mod\person_2.glb"></a-asset-item>
5   <a-asset-item id="person3" src="mod\person_3.glb"></a-asset-item>
6   <a-asset-item id="person4" src="mod\person_4.glb"></a-asset-item>
7   <a-asset-item id="logo" src="mod\logo_nto.glb"></a-asset-item>
```



```

8 </a-asset>
9
10 <a-marker preset="custom" type="pattern" url="pat\pattern_1.patt">
11   <a-gltf-model src="#babr"></a-gltf-model>
12 </a-marker>
13
14 <a-marker preset="custom" type="pattern" url="pat\pattern_2.patt">
15   <a-gltf-model src="#person1"></a-gltf-model>
16 </a-marker>
17
18 <a-marker preset="custom" type="pattern" url="pat\pattern_3.patt">
19   <a-gltf-model src="#person2"></a-gltf-model>
20 </a-marker>
21
22 <a-marker preset="custom" type="pattern" url="pat\pattern_4.patt">
23   <a-gltf-model src="#person3"></a-gltf-model>
24 </a-marker>
25
26 <a-marker preset="custom" type="pattern" url="pat\pattern_5.patt">
27   <a-gltf-model src="#person4"></a-gltf-model>
28 </a-marker>
29
30 <a-marker preset="custom" type="pattern" url="pat\pattern_6.patt">
31   <a-gltf-model src="#logo"></a-gltf-model>
32 </a-marker>

```

### Задача VI.2.7.7. Объекты внутри куба

Допишите фрагмент кода так, чтобы объект был размещен внутри куба. Пример работы после дополнения кода представлен в анимации.



Ссылка на анимацию: <https://disk.yandex.ru/i/1LMEQjsyHZx2pA>.

Фрагмент кода.

```
<a-box material="src: #tx; side: back"></a-box>
```

При этом:

- для куба можно использовать любую текстуру;



- 
- объект внутри куба должен быть 3D моделью, представляющий из себя логотип НТО (Задача Блок для дизайнеров.3).

Реализуйте данный сценарий и пришлите ссылку на glitch.

Примечание: для тестирования данной задачи воспользуйтесь любой другой 3D моделью.

## Решение

```
<!DOCTYPE html>
<html>
  <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
  <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>
  <script src="https://cdn.jsdelivr.net/gh/c-frame/aframe-extras@6.1.1/dist/aframe-extras.min.js"></script>
  <body style="margin : 0px; overflow: hidden;">
    <script>
      AFRAME.registerComponent('opacity', {
        init: function(){
          var el = this.el;
          this.el.addEventListener('model-loaded', function (e) {
            console.log(el.object3D.children[0].children[0].material.opacity)
            el.object3D.children[0].children.forEach(element => {
              element.material.opacity = 0.5
            })
          })
        }
      })
    </script>
    <a-scene
      embedded
      vr-mode-ui="enabled: false;"
      arjs="trackingMethod: best; debugUIEnabled: false;"
      renderer="logarithmicDepthBuffer: true;"
    >
      <a-assets>
        <a-asset-item id="obj" src="./primitives.obj"></a-asset-item>
        <a-asset-item id="mtl" src="./primitives.mtl"></a-asset-item>
      </a-assets>
      <a-marker id="marker_1" preset="kanji">
        <a-obj-model src="#obj" mtl="#mtl" opacity ></a-obj-model>

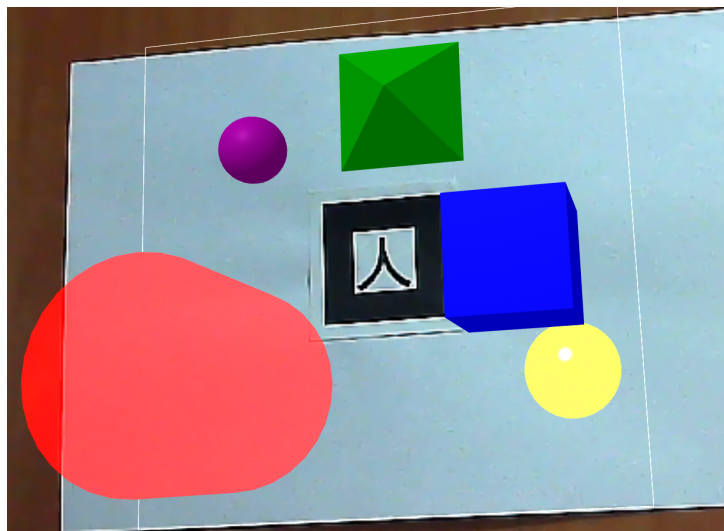
        <!--ваш код-->

      </a-marker>

      <a-entity camera></a-camera>
    </a-scene>
  </body>
</html>
```

## Задача VI.2.7.8. Множество динамических объектов на одном маркере

Ниже приведен код, который проецирует на маркер иллюзии примитивных объектов.



Пример проекций и фигур изображен на рисунке справа: полупрозрачным отображены проекции, плотным цветом — выставленные фигуры.

Добавьте в код эти объекты (каждый по отдельности) и расположите их на месте соответствующей проекции с нужным цветом. При этом для объекта используйте теги примитивов, а не `a-entity`.

Цвет задается свойством `color` при этом указывается ключевым словом из цветов стандартной палитры (например `blue`, а не `#0000ff`/`#00f`) и сразу после указания тега примитива или его формы (например `<a-box color="red" . . . >`). Не используйте `material` для задания цвета.

Позиция указывается сразу после цвета с точностью до десятых.

Все свойства примитивов напишите отдельными атрибутами.

Код с проекциями.

```
<!DOCTYPE html>
<html>
  <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
  <script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe-
  ↪ e-ar.js"></script>
  <script src="https://cdn.jsdelivr.net/gh/c-frame/aframe-extras@6.1.1/dist/aframe-
  ↪ me-extras.min.js"></script>
  <body style="margin : 0px; overflow: hidden;">
    <script>
      AFRAME.registerComponent('opacity', {
        init: function(){
          var el = this.el;
          this.el.addEventListener('model-loaded', function (e) {
            console.log(el.object3D.children[0].children[0].material.opacity)
            el.object3D.children[0].children.forEach(element => {
              element.material.opacity = 0.5
            })
          })
        }
      })
    </script>
  </body>
</html>
```

```

        })
    })
}
})
</script>
<a-scene
  embedded
  vr-mode-ui="enabled: false;"
  arjs="trackingMethod: best; debugUIEnabled: false;"
  renderer="logarithmicDepthBuffer: true;"
>
  <a-assets>
    <a-asset-item id="obj" src="./primitives.obj"></a-asset-item>
    <a-asset-item id="mtl" src="./primitives.mtl"></a-asset-item>

  </a-assets>
  <a-marker id="marker_1" preset="kanji">
    <a-obj-model src="#obj" mtl="#mtl" opacity ></a-obj-model>

    <!--ваш код-->

  </a-marker>

  <a-entity camera></a-camera>
</a-scene>
</body>
</html>

```

Дополнительно вам понадобятся файлы проекции:

- ссылка на primitives.mtl: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/primitives1.mtl](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/primitives1.mtl);
- ссылка на primitives.obj: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/primitives.obj](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/primitives.obj).

В ответ прикрепите все, что находится в теге `<a-marker id="marker_1" preset="kanji">` вместе с ним.

Примечание: сам маркер можно заменить.

Примечание: все примитивы должны быть вложены в `<a-marker>` не должно быть дополнительных блоков, в которых они будут вложены.

## Решение

```

1 <a-marker id="marker_1" preset="kanji">
2   <a-obj-model src="#obj" mtl="#mtl" opacity ></a-obj-model>
3   <a-box color="blue" position="2 -1 0" depth="2" height="2" width="2"></a-box>
4   <a-cylinder color="red" position="-2 2.5 2" radius="1.5"
   ↪ height="3"></a-cylinder>
5   <a-sphere color="yellow" position="3 0.7 2" radius="0.7"></a-sphere>
6   <a-sphere color="purple" position="-1.5 1.5 -1.5" radius="0.5"></a-sphere>
7   <a-octahedron color="green" position="0.4 0 -2.5" scale="1 0.71 1"
   ↪ radius="1.42" rotation="0 45 0"></a-octahedron>
8 </a-marker>

```

---

## Способ оценки работы

Для проверки результата использовался следующий код на языке Python 3.

```
from bs4 import BeautifulSoup
import bs4
from typing import Tuple
text = """
    <a-marker id="marker_1" preset="kanji">
    <a-obj-model src="#obj" mtl="#mtl" opacity ></a-obj-model>
    <a-box color="blue" position="2 -1 0" depth="2" height="2" width="2"></a-box>
    <a-cylinder color="red" position="-2 2.5 2" radius="1.5" height="3"></a-cylinder>
    <a-sphere color="yellow" position="3 0.7 2" radius="0.7"></a-sphere>
    <a-sphere color="purple" position="-1.5 1.5 -1.5" radius="0.5"></a-sphere>
    <a-octahedron color="green" position="0.4 0 -2.5" radius="1.42" rotation="0 45
↵ 0"></a-octahedron>
    </a-marker>
    """

ans = {}
for item in BeautifulSoup(text, 'html.parser').find('a-marker'):
    if isinstance(item, bs4.element.Tag) and item.get('color'):
        attrs = item.attrs
        ans[item.name + attrs['color']] = attrs

def check(reply):
    try:
        your_ans = {}
        for item in BeautifulSoup(reply, 'html.parser').find('a-marker'):
            if isinstance(item, bs4.element.Tag) and item.get('color'):
                attrs = item.attrs
                your_ans[item.name + attrs.get('color')] = attrs

        for key in ans.keys():
            # print(key)
            if key == 'a-octahedrongreen':
                color1 = your_ans[key].get('color')
                radius1 = float(your_ans[key].get('radius'))
                position1 = [float(el) for el in your_ans[key].get('position').split()]
                rotation1 = [float(el) for el in your_ans[key].get('rotation').split()]

                color2 = ans[key].get('color')
                radius2 = float(ans[key].get('radius'))
                position2 = [float(el) for el in ans[key].get('position').split()]
                rotation2 = [float(el) for el in ans[key].get('rotation').split()]

                if color1 != color2:
                    return False

                if abs(radius1 - radius2) > 0.1:
                    return False

                if abs(position1[0] - position2[0]) > 0.05 or abs(position1[1] -
↵ position2[1]) > 0.05 or abs(position1[2] - position2[2]) > 0.05:
                    return False

                if abs(rotation1[0] - rotation2[0]) > 0.05 or abs(rotation1[1] -
↵ rotation2[1]) > 0.05 or abs(rotation1[2] - rotation2[2]) > 0.05:
                    return False
```

```

else:
    for kk in ans.get(key).keys():
        # print(f"\t {kk} -> {your_ans.get(key).get(kk)} and
        ↪ {ans.get(key).get(kk)}")
        if your_ans.get(key).get(kk) != ans.get(key).get(kk):
            return False

    return True
except:
    return False

```

### Задача VI.2.7.9. Прыгающий мяч (анимация)

Ниже представлен код для имитации прыгающего мяча, реализованного через анимацию движения сферы вверх и вниз.

Посмотрите видео-демонстрацию: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/Task4.mp4](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/Task4.mp4).

Скачайте код, найдите ошибки и исправьте код. В ответе укажите номера строк через пробел, в которых были допущены ошибки, в порядке их обнаружения в коде.

- Если вы думаете, что ошибок нет, напишите 0.
- Название `id` соответствует содержимому тега.
- Не возникают ошибки во время исполнения сценария.

Пример ответа: 1 2 7

Что соответствует тому, что в коде были допущены ошибки на строках 1, 2 и 7.

### Решение

Верный код должен быть следующим.

```

1 <html>
2 <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
3 <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-a_j
  ↪ r.js"></script>
4 <script>
5     document.addEventListener("DOMContentLoaded", () =>{
6         let marker = document.querySelector("#marker")
7         let sp = document.querySelector("#sphere")
8
9         marker.addEventListener('markerFound', function() {sp.emit("up")
10     })
11     sp.addEventListener('animationcomplete__1',function(){
12         sp.emit("down")
13     })
14     sp.addEventListener('animationcomplete__2',function()sp.emit("up")
15     })
16 })
17 </script>
18
19 <body style="margin : 0px; overflow: hidden;">
20 <a-scene embedded arjs>
21 <a-assets>
22     
23 </a-assets>

```

```

24 <a-marker id= "marker" preset='hiro'>
25
26 <a-sphere id ="sphere" src="#img" position="0 0 0" rotation="-90 0 0"
27 animation__1="property: position; from:0 0 0; to: 0 5 0; dur: 500;
28 easing: easeOutQuad; startEvents: up"
29 animation__2="property: position; from:0 5 0; to: 0 0 0; dur: 500;
30
31 easing: easeInQuad; startEvents: down">#resumeEvents, startEvents, createEvents
32 </a-shpere>
33
34 </a-marker>
35 <a-entity camera></a-entity>
36 </a-scene>
37 </body>
38 </html>

```

Ошибки могут быть на строках: 9, 11, 14, 22, 24, 26, 27, 28, 29, 31.

### Задача VI.2.7.10. Мяч прыгающий (компонент)

Ниже дан фрагмент кода для создания компонента A-Frame, внутри которого написана логика движения мяча вверх и вниз(прыжок).

Но в нем пропущены кусочки кода в событиях `init`, `update`, `tick`. Разместите следующие блоки кода в соответствующие события.

№	Блок кода
1.	<code>currentPosition = new THREE.Vector3() currentPosition.add(this.toPosition); currentPosition.multiplyScalar(t*t) currentPosition.add(inter);</code>
2.	<code>this.toPosition = this.target.position;</code>
3.	<code>this.target = this.data.t1.object3D;</code>
4.	<code>let inter = new THREE.Vector3( ); inter.add(this.toPosition); inter.multiplyScalar(0.7); inter.y += 5; inter.multiplyScalar(t*(1-t));</code>
5.	<code>this.el.setAttribute('position', currentPosition);</code>
6.	<code>let t = Math.abs((time*0.001)%2-1); t = Math.min(t,0.98); t = Math.max(t,0.02);</code>

Фрагмент кода.

```

<!DOCTYPE html>
<html>
  <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
  <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe.js"
  ↪ e-ar.js"></script>

  <body>
    <script>
      AFRAME.registerComponent('follow', {
        schema: {

```

```

    t1: {
      type: "selector"
    },

  },
  init: function () {
    <!-- пропуски-->
  },

  update: function(){
    <!-- пропуски-->
  },

  tick: function(time, timeDelta) {

    <!-- пропуски-->

  },
});

</script>
<a-scene>
  <a-entity>
    <a-marker preset="hiro">
      <a-sphere radius="0" id="t1" position="0 0 0"></a-sphere>
      <a-sphere radius="0.25" follow="t1: #t1;" color="blue" position="0 0.2
        ↪ 0"></a-sphere>
    </a-marker>
    <a-entity camera></a-entity>
  </a-entity>
</a-scene>
</body>
</html>

```

В каждом событии должен стоять хотя бы один блок.

Ответ напишите в следующем формате:

```

init: 1
update: 1
tick: 2 3

```

где 1, 2, 3 — номера блоков.

### *Решение*

Верный код.

```

1 <!DOCTYPE html>
2 <html>
3 <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
4 <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-
  ↪ ar.js"></script>

```

```

5
6 <body>
7 <script>
8   AFRAME.registerComponent('follow', {
9     schema: {
10
11       t1: {
12         type: "selector"
13       },
14
15     },
16     init: function () {
17       this.target = this.data.t1.object3D;
18     },
19
20     update: function(){
21       this.toPosition = this.target.position;
22     },
23
24     tick: function(time, timeDelta) {
25
26       let t = Math.abs((time*0.001)%2-1);
27       t = Math.min(t,0.98);
28       t = Math.max(t,0.02);
29
30
31       let inter = new THREE.Vector3( );
32       inter.add(this.toPosition);
33       inter.multiplyScalar(0.7);
34       inter.y += 5;
35       inter.multiplyScalar(t*(1-t));
36
37
38       currentPosition = new THREE.Vector3()
39       currentPosition.add(this.toPosition);
40       currentPosition.multiplyScalar(t*t)
41       currentPosition.add(inter);
42
43
44       this.el.setAttribute('position', currentPosition);
45     },
46   });
47
48 </script>
49 <a-scene>
50   <a-entity>
51     <a-marker preset="hiro">
52       <a-sphere radius="0" id="t1" position="0 0 0"></a-sphere>
53       <a-sphere radius="0.25" follow="t1: #t1;" color="blue" position="0 0.2
54         ↪ 0"></a-sphere>
55     </a-marker>
56     <a-entity camera></a-entity>
57   </a-entity>
58 </a-scene>
59 </body>
60 </html>

```

**Ответ:**



```
init: 3
update: 2
tick: 6 4 1 5
```

## Задача VI.2.7.11. Пин-понг

Используя код из прошлого шага доработайте его так, чтобы при появлении второго маркера, мячик начинал прыгать между этими маркерами, перепрыгивая с одного на другой.

Пример выполнения можно посмотреть тут: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/Task5.mp4](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/Task5.mp4).

Если в проекте во время исполнения будут возникать ошибки в консоли, то решение считается неверным.

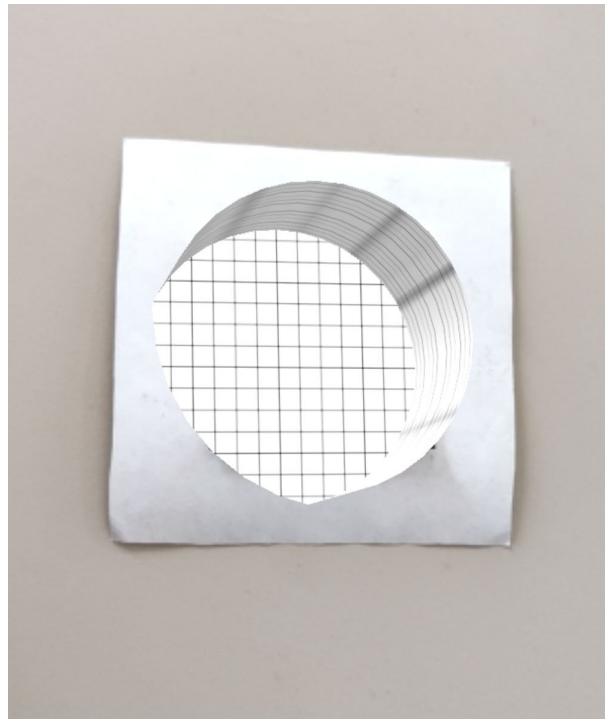
Разместите проект на glitch и пришлите ссылку на него.

## Решение

```
1  <!DOCTYPE html>
2  <html>
3  <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
4  <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-
   ↪ ar.js"></script>
5
6  <body>
7  <script>
8    AFRAME.registerComponent('follow', {
9      schema: {
10       t1: {
11         type: "selector",
12       },
13       t2: {
14         type: "selector",
15       },
16       speed: { type: 'number' }
17     },
18
19     init: function () {
20       this.targets = [this.data.t1.object3D, this.data.t2.object3D];
21     },
22
23     update: function(){
24       this.fromPosition = this.targets[0].position;
25       this.toPosition = this.targets[1].position;
26     },
27
28     tick: function (time, timeDelta) {
29
30       var t = Math.abs((time*0.001)%2-1);
31       t = Math.min(t,0.98);
32       t = Math.max(t,0.02);
33
34       let inter = new THREE.Vector3();
35
36       inter.add(this.fromPosition);
37       inter.add(this.toPosition);
```

```
38     inter.multiplyScalar(0.7);
39     inter.y += 5;
40
41     inter.multiplyScalar(t*(1-t));
42
43
44     startPosition = new THREE.Vector3();
45     startPosition.add(this.fromPosition);
46     startPosition.multiplyScalar(t*t);
47
48     stopPosition = new THREE.Vector3();
49     stopPosition.add(this.toPosition);
50     stopPosition.multiplyScalar((1-t)*(1-t));
51
52     currentPosition = new THREE.Vector3();
53     currentPosition.add(startPosition);
54     currentPosition.add(stopPosition);
55
56     currentPosition.add(inter)
57
58     this.el.setAttribute('position', currentPosition);
59 }
60 });
61
62 </script>
63 <a-scene embedded
64     ar
65     vr-mode-ui="enabled:false;"
66     arjs="trackingMethod:best; sourceType: webcam"
67     renderer="logarithmicDepthBuffer:true;">
68
69     <a-entity>
70         <a-marker type='pattern' url='trackers/letterA.patt'>
71             <a-sphere radius="0" id="t1" color="red" position="1 0 -3"></a-sphere>
72         </a-marker>
73
74         <a-marker type='pattern' url='trackers/letterB.patt'>
75             <a-sphere radius="0" id="t2" color="yellow" position="0 0 -3"></a-sphere>
76         </a-marker>
77
78         <a-sphere radius="0.25" follow="t1: #t1; t2:#t2; speed: 1" color="blue"
79             ↪ position="0 0.2 -3"></a-sphere>
80     </a-entity>
81     <a-entity camera></a-entity>
82 </a-scene>
83 </body>
84
85 </html>
```

## Задача VI.2.7.12. Цилиндрическая дырка в полу



Допишите фрагмент кода так, чтобы в дополненной реальности отображалась дырка в полу цилиндрической формы.

Пример представлен на картинке.

Фрагмент кода.

```
<!DOCTYPE html>
<html>
  <title>{Название задания}</title>
  <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
  <script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe-
  ↪ me-ar.js"></script>

  <body style="margin : 0px; overflow: hidden;">
    <a-scene
      embedded
      vr-mode-ui="enabled: false;"
      arjs="trackingMethod: best; debugUIEnabled: false;"
      renderer="logarithmicDepthBuffer: true;">

      <a-marker preset="custom" type="pattern"
        ↪ url="./src/markers/patt/letterA.patt">

        <!--ваш фрагмент кода-->

      </a-marker>

      <a-entity camera></a-entity>
    </a-scene>
```

---

```
</body>
</html>
```

В ответ напишите фрагмент кода.

Примечание: отступы и переносы строк могут быть в любом месте.

### *Решение*

```
<!DOCTYPE html>
<html>
  <title>{Название задания}</title>
  <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
  <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe.js"
    ↪ me-ar.js"></script>
  <script src="./src/threex-portal.js"></script>
  <script>
    AFRAME.registerComponent('arjs-portal-door', {
      schema: {
        url : {
          type: 'string',
        },
        doorWidth : {
          type: 'number',
          default: 1,
        },
        doorHeight : {
          type: 'number',
          default: 2,
        },
      },
      init: function () {
        var _this = this

        var doorWidth = this.data.doorWidth
        var doorHeight = this.data.doorHeight
        var imageURL = this.data.url

        var portalDoor = new THREEEx.Portal360(imageURL, doorWidth, doorHeight)
        this._portalDoor = portalDoor

        this.el.object3D.add(portalDoor.object3d)
      },
      tick: function(){
        this._portalDoor.update()
      }
    })

    AFRAME.registerPrimitive('a-portal-door', AFRAME.utils.extendDeep({},
    ↪ AFRAME.primitives.getMeshMixin(), {
      defaultComponents: {
        'arjs-portal-door': {},
      },
    })
```

```

    },
    mappings: {
      'url': 'arjs-portal-door.url',
    }
  )))
</script>
<body style="margin : 0px; overflow: hidden;">
  <a-scene embedded
    vr-mode-ui="enabled: false;"
    arjs="trackingMethod: best; debugUIEnabled: false;"
    renderer="logarithmicDepthBuffer: true;">

    <a-marker preset="custom" type="pattern"
      ↪ url="./src/markers/patt/letterA.patt" >

      <a-portal-door url="./src/babr.jpg" position='0 0 0' scale='1 1 1'
        ↪ rotation='0 0 0'><a-portal-door>

    </a-marker>

    <a-entity camera></a-entity>
  </a-scene>
</body>

</html>

```

### Задача VI.2.7.13. Портал в виртуальный мир



В этом задании мы предлагаем создать вам портал в другой мир.

Разработайте следующую систему:

- При использовании устройства должна включаться камера, отображая пространство вокруг пользователя, а над маркером должен появляться портал, в котором можно увидеть кусочек другой реальности.
- При нажатии на портал пользователь должен оказаться внутри него (портала) и увидеть панораму другого места.
- Так же, должна быть реализована возможность пользователя возвращаться в начальную точку.

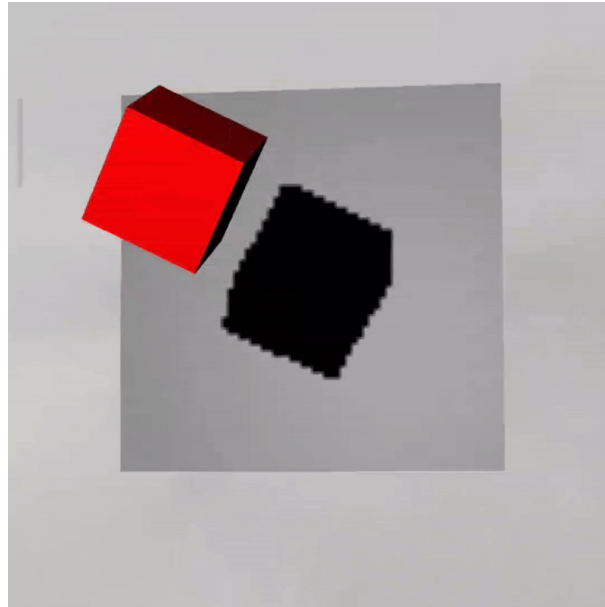
---

В качестве ответа прикрепите ссылку на glitch.

## Решение

```
1  <!DOCTYPE html>
2  <html>
3    <title>{Название задания}</title>
4    <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
5    <script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe.js"
6      ↪ -ar.js"></script>
7    <script>
8      var mesh = document.getElementById('model');
9      document.addEventListener("DOMContentLoaded", () => {
10       mesh = document.getElementById('model').object3D;
11       document.body.addEventListener('keydown', keyPressed);
12       console.log(mesh);
13       function keyPressed(e){
14         let key = e.key;
15         if (key == 'ArrowUp'){
16           mesh.rotateX(0.1);
17         }
18         if (key == 'ArrowDown'){
19           mesh.rotateX(-0.1);
20         }
21         if (key == 'ArrowLeft'){
22           mesh.rotateY(0.1);
23         }
24         if (key == 'ArrowRight'){
25           mesh.rotateY(-0.1);
26         }
27       }
28     })
29   </script>
30   <body style="margin : 0px; overflow: hidden;">
31     <a-scene shadow="autoUpdate: true"
32       embedded
33       vr-mode-ui="enabled: false;"
34       arjs="trackingMethod: best; debugUIEnabled: false;"
35       renderer="logarithmicDepthBuffer: true;"
36     >
37     <a-marker id="marker" preset="kanji">
38       <a-plane color="grey" rotation="90 0 0" position="0 0 0" width="1"
39         ↪ height="1" shadow="receive:true; cast:false"
40         ↪ material="side:double"></a-plane>
41
42     <a-box id='model' scale="0.25 0.25 0.25" color="red" position="0 2 0"
43       ↪ shadow></a-box>
44
45     <a-entity light="castShadow: true; type:point" color="white" position="0 5
46       ↪ 0"></a-light>
47   </a-marker>
48   <a-entity camera></a-camera>
49 </a-scene>
50 </body>
51 </html>
```

## Задача VI.2.7.14. Управление виртуальным объектом при помощи клавиатуры



На анимированном изображении представлен AR-сценарий, где:

- при наведении на маркер возникает 3D модель (на картинке красный куб), холст и свет, для отбрасывания тени на холст;
- управление 3D моделью происходит по нажатию клавиш на клавиатуре;
- при манипулировании 3D-моделью, ее тень изменяется.

Реализуйте данный сценарий и пришлите ссылку на glitch.

### Решение

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <script src="https://aframe.io/releases/1.4.1/aframe.min.js"></script>
5  <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
6  <script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe_
   ↪ -ar-nft.js"></script>
7
8  <script>
9    AFRAME.registerComponent("mybox", {
10     init: function () {
11       var material = new THREE.MeshBasicMaterial({ color: 0x00ff00 });
12       var geometry = new THREE.BoxGeometry(0.05, 0.05, 0.05);
13       this.el.setObject3D("mesh", new THREE.Mesh(geometry, material));
14     },
15     });
16 </script>
17 </head>
18
19 <body>
20 <a-scene embedded arjs>
21   <a-marker preset="hiro">
```

```

23     <script>
24     async function fetchJSON() {
25         const response = await fetch("/data.json");
26         const voxels_arr = await response.json();
27         // console.log(voxels_arr['voxels']);
28         create_models(voxels_arr["voxels"]);
29         return voxels_arr;
30     }
31
32     fetchJSON().then((voxels_arr) => {
33         voxels_arr;
34     });
35
36     function create_models(arr) {
37         for (i of arr) {
38             // console.log(i["x"]/5);
39             let b = document.createElement("a-box");
40             b.setAttribute(
41                 "position",
42                 `${i["x"] / 15} ${i["y"] / 15} ${i["z"] / 15}`
43             );
44             b.setAttribute("mybox", "");
45             let ul = document.querySelector("a-marker");
46             ul.append(b);
47         }
48     }
49     </script>
50
51     </a-marker>
52     <!-- После описания маркера и оверлея обязательным компонентом для дополненной
53     ↪ реальности является камера-->
54     <a-entity camera></a-entity>
55     </a-scene>
56     </body>
57 </html>

```

## Задача 0. Тень с секретом

С помощью реализованного на прошлом шаге сценария посмотрите, что за предмет скрыт в тени 3D модели.

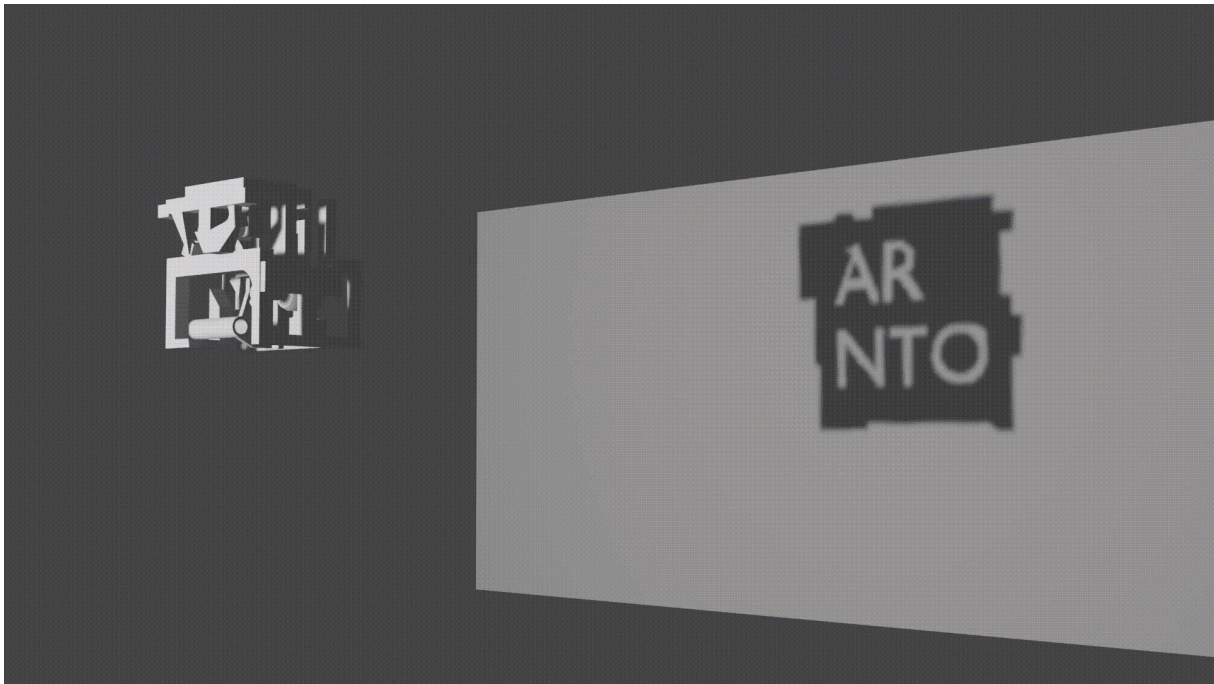
Ссылка на 3D модель: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/shadow.glb](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/shadow.glb).

Для ответа на данный вопрос необходимо показать организаторам (представителям разработчиков профиля на площадке) скриншот тени объекта и назвать объект, который вы увидели. Если ответ верный, то организатор передаст вам код, который необходимо ввести в качестве ответа к данному шагу задания.



---

## Решение



### Задача 1. 3D фигура отдельными точками

Отображение 3D фигур в дополненной реальности возможно и другим способом, где важно то, с какой стороны вы смотрите на объект.

Реализуйте сценарий, где в дополненной реальности отображается 3D-модель, которая дана вам в виде json файла: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/chaos.json](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/chaos.json).

При этом:

- отобразите модель точками — в виде куба с зеленой (#00FF00) текстурой;
- точка является разработанным вами компонентом AFrame или примитивом AFrame;
- нельзя переименовывать файл json, менять его расширение и содержимое;
- нельзя помещать содержимое json в скрипт;
- реализуйте сценарий без использования фреймворков и библиотек, на «голом» js.

Разместите проект на glitch и пришлите ссылку на него в качестве ответа.

## Решение

```
1 <html>
2   <head>
3     <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
4     <script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe_j
   ↪ e-ar.js"></script>
5     <style>
6       .button{
7         background-color: aquamarine;
```

```
8     width: 10vh;
9     height: 10vh;
10    position: fixed;
11  }
12  .up{
13    top: 0px;
14    left: 45%;
15  }
16  .down{
17    bottom: 0px;
18    left: 45%;
19  }
20  .left{
21    top: 45%;
22    left: 0px;
23  }
24  .right{
25    top: 45%;
26    right: 0px;
27  }
28  </style>
29  </head>
30  <body>
31    <a-scene
32      vr-mode-ui="enabled: false;"
33      renderer="logarithmicDepthBuffer: true;"
34      embedded
35      arjs="trackingMethod: best; sourceType: webcam;debugUIEnabled: false;"
36    >
37      <a-assets>
38        
39        
40      </a-assets>
41      <a-marker preset="hiro">
42        <a-box id="control" src="#tx"></a-box>
43      </a-marker>
44      <a-entity camera></a-entity>
45    </a-scene>
46    <div class="up button"></div>
47    <div class="down button"></div>
48    <div class="right button"></div>
49    <div class="left button"></div>
50  </body>
51  <script>
52    var obj = document.getElementById("control")
53    let upb = document.getElementsByClassName("up")[0]
54    let downb = document.getElementsByClassName("down")[0]
55    let rightb = document.getElementsByClassName("right")[0]
56    let leftb = document.getElementsByClassName("left")[0]
57    var dx = 25
58
59    <!-- ваш код-->
60
61  </script>
62  </html>
```

---

## Задание 2. Управление виртуальной камерой при помощи клавиатуры

Вам дан шаблон кода, где реализован сценарий управления виртуальной камерой при помощи клавиатуры. Данную функцию удобно использовать, когда реальная камера статична, а объект требуется рассматривать с разных: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/Task12.mp4](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/Task12.mp4).

```
upb.addEventListener("click",function(){obj.object3D.rotation.x += dx})
downb.addEventListener("click",function(){obj.object3D.rotation.x -= dx})
rightb.addEventListener("click",function(){obj.object3D.rotation.y += dx})
leftb.addEventListener("click",function(){obj.object3D.rotation.y -= dx})
```

Допишите шаблон кода так, чтобы камера управлялась верно.

В отклик разместите дописанный вами фрагмент кода.

### Решение

```
<html>
<head>
<script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
<script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>
<script>

document.addEventListener("DOMContentLoaded", () =>{
  var map = document.querySelector("#map")
  var apply = document.querySelector("#apply")
  var build = document.querySelector("#build")

  apply.addEventListener('markerFound', function() {
    if (build.object3D.visible){
      let temp = build.children[0].cloneNode(true)
      let wloc = new THREE.Vector3()
      let xyz = build.object3D.getWorldPosition(wloc)
      let a = map.object3D.worldToLocal(wloc)
      temp.object3D.position.x = a.x
      temp.object3D.position.y = a.y
      temp.object3D.position.z = a.z
      temp.setAttribute('color', '#fff00')
      map.appendChild(temp)
    }
  })
})

</script>
</head>
<body>
<a-scene
  vr-mode-ui="enabled: false;"
  renderer="logarithmicDepthBuffer: true;"
```

```
embedded
arjs="trackingMethod: best; sourceType: webcam;debugUIEnabled: false;
↳ detectionMode: mono_and_matrix; matrixCodeType: 3x3;"
>

<a-marker id="map" preset="kanji">
  <a-box scale="0.3 0.3 0.3" ></a-box>
</a-marker>

<a-marker id="apply" preset="hiro">
  <a-box scale="0.3 0.3 0.3" color="#00FF00" ></a-box>
</a-marker>

<a-marker id="build" type="barcode" value="5">
  <a-box color="#FF0000" ></a-box>
</a-marker>

<a-entity camera></a-entity>
</a-scene>
</body>
</html>
```

### Задача 3. Мультимаркер и его взаимодействие

Реализуйте AR-сценарий, который представлен на видео: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/Task13.mp4](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/Task13.mp4).

Описание сценария:

- Имеются три различных маркера.
- Один маркер нужен для фиксации позиции будущих объектов на плоскости и их переноса в пространстве (белый куб на видео).
- Второй маркер позволяет предварительно поставить объект в пространстве (красные кубы).
- Третий маркер фиксирует выставленный объект в пространстве (желтые кубы — зафиксированные объекты).
- При исчезновении маркера из поля зрения объекты не забываются, а появляются в том виде, в котором были зафиксированы, когда маркер снова начинает отслеживаться.

Разместите проект на glitch и пришлите ссылку на него.

### Решение

```
1 <html>
2   <head>
3     <style>
4       .buttons{
5         position: fixed;
6         bottom: 0px;
7         left: 0%;
8       }
9     </style>
```

```
10     .buttons div{
11         background-color: aquamarine;
12         width: 10vh;
13         height: 10vh;
14
15     }
16     .buttons div span{
17         position: relative;
18         top:50%;
19         left:20%;
20         font-size: 20pt;
21     }
22     #score{
23         background-color: aquamarine;
24         width: 10vw;
25         height: 20vh;
26         position: fixed;
27         top:0px;
28         left:0px;
29     }
30     #score span{
31         position: relative;
32         top:50%;
33         left:20%;
34         font-size: 25pt;
35     }
36     #startbutton{
37         background-color: aquamarine;
38         width: 10vw;
39         height: 10vh;
40         position: fixed;
41         bottom:5vh;
42         left:50vw;
43     }
44     #startbutton span{
45         position: relative;
46         top:25%;
47         left:20%;
48         font-size: 25pt;
49     }
50
51
52     </style>
53     <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
54     <script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe.js"
55     ↪ e-ar.js"></script>
56     <script>
57     var spawntime = 5000
58     var reduce_spawntime = 100
59     var reduce_sp_counttoreduce = 5
60     var is_started = false
61     var is_gameover = false;
62     var life = 5;
63     var score = 0;
64     var drop = false;
65     var sphere = false;
66     var red = false;
67     var green = false;
68     var blue = false;
69     var bgm = new Audio('bgm.mp3');
```

```

69     bgm.volume=0.05
70     var lbgm = new Audio('lbgm.mp3')
71     lbgm.volume = 0.05
72     var sfxscore = new Audio('getscore.mp3')
73     var sfxlife = new Audio('looselife.mp3')
74     var ticker
75     document.addEventListener("DOMContentLoaded",() =>{
76         var startB = document.querySelector("#startbutton")
77         var game = document.querySelector("#game")
78         var cubeF = document.querySelector("#box_fetch")
79         var sphereF = document.querySelector("#sphere_fetch")
80         var dropG = document.querySelector("#drop_gate")
81         var formG = document.querySelector("#form_gate")
82         var scoreTab = document.querySelector("#score")
83         function updateColor(){
84             console.log(cubeF)
85             let color = "#"+((red)?("ff"):(00))+((green)?("ff"):(00))+((blue)?("f
↵ f"):(00));
86             color = (color == "#ffffff")?("#aaaaaa"):(color
↵ =="#000000")?("#555555"):(color)
87             cubeF.setAttribute("material","color",color)
88             sphereF.setAttribute("material","color",color)
89         }
90         function updateDrop(){
91             dropG.setAttribute("rotation","0 0 "+((drop)?("180"):(90)))
92         }
93         function updateForm(){
94             console.log(sphere)
95             formG.setAttribute("rotation","0 0 "+((sphere)?("135"):(45)))
96         }
97         function updateScore(){
98             scoreTab.children[0].innerHTML = "Score:"+score+"<br>"+"Life:"+life
99         }
100        var redM = document.querySelector("#redM");
101        var greenM = document.querySelector("#greenM");
102        var blueM = document.querySelector("#blueM");
103        var shapeM = document.querySelector("#shapeM");
104        var dropM = document.querySelector("#dropM");
105        var sp = document.querySelector("#startPoint")
106
107        game.addEventListener("markerFound",function(){
108            is_started = true
109        })
110        game.addEventListener("markerLost",function(){
111            is_started = false
112        })
113
114        redM.addEventListener("markerFound",function(){
115            red = true
116            updateColor()
117        })
118        redM.addEventListener("markerLost",function(){
119            red = false
120            updateColor()
121        })
122        greenM.addEventListener("markerFound",function(){
123            green = true
124            updateColor()
125        })
126        greenM.addEventListener("markerLost",function(){

```

```
127     green = false
128     updateColor()
129 }
130 blueM.addEventListener("markerFound",function(){
131     blue = true
132     updateColor()
133 })
134 blueM.addEventListener("markerLost",function(){
135     blue = false
136     updateColor()
137 })
138 shapeM.addEventListener("markerFound",function(){
139     sphere = true
140     updateForm()
141 })
142 shapeM.addEventListener("markerLost",function(){
143     sphere = false
144     updateForm()
145 })
146 dropM.addEventListener("markerFound",function(){
147     drop = true
148     updateDrop()
149 })
150 dropM.addEventListener("markerLost",function(){
151     drop = false
152     updateDrop()
153 })
154
155 function check(obj_color,obj_form,base_color,base_form){
156     if(is_started){
157         console.log(obj_color, base_color)
158         if(obj_form == 'a-torus')
159         {
160             life = 0
161             gameover()
162         }
163         else
164             if(obj_form != base_form){
165                 sfxlife.play()
166                 life -= 1
167                 if (life<= 0)
168                 {
169                     gameover()
170                 }
171             }
172         else
173             if (obj_color == base_color){
174                 sfxscore.play()
175                 score += 1
176             }
177         updateScore()
178     }
179 }
180 function gameover(){
181     if(!is_gameover){
182         is_gameover = true
183         clearInterval(ticker)
184         console.log("LOSE")
185         game.children[0].children[0].setAttribute('color','#FF0000')
186         bgm.pause()
```



```

187         lbgm.play()
188         sp.innerHTML=""
189     }
190 }
191 function generator(gap,sp){
192     let form = (Math.random()>=0.5)?('a-box'):( 'a-sphere')
193     form = (Math.random()<=0.2)?('a-torus'):(form)
194     let color = "#"+((Math.random()>=0.5)?("FF"):( "00"))+((Math.random()>=0.5)
↵ )?("FF"):( "00"))+((Math.random()>=0.5)?("FF"):( "00"))
195     var obj = document.createElement(form)
196     sp.appendChild(obj)
197     obj.setAttribute('color',color)
198     obj.setAttribute('scale',"0.5 0.5 0.5")
199     obj.setAttribute('animation__start',"property:position;from:0 0 0;to:6
↵ 0 0;dur:"+gap+";loop:false;easing:linear")
200     obj.setAttribute('animation__1',"property:position;from:6 0 0;to:10 0
↵ 0; startEvents:p1;dur:"+gap+";loop:false;easing:linear")
201     obj.setAttribute('animation__drop1',"property:position;from:6 0 0;to:6
↵ 6 0; startEvents:d1;pauseEvents:pause;resumeEvents:resume;dur:"+gap
↵ "+";loop:false;easing:linear")
202     obj.setAttribute('animation__drop2',"property:position;from:6 6 0;to:0
↵ 6 0; startEvents:d2;pauseEvents:pause;resumeEvents:resume;dur:"+gap
↵ "+";loop:false;easing:linear")
203     obj.setAttribute('animation__drop3',"property:position;from:0 6 0;to:0
↵ 0 0; startEvents:d3;pauseEvents:pause;resumeEvents:resume;dur:"+gap
↵ "+";loop:false;easing:linear")
204     obj.setAttribute('animation__cube',"property:position;from:10 0 0;to:14
↵ -4.5 0; startEvents:cube;pauseEvents:pause;resumeEvents:resume;dur:
↵ "+gap+";loop:false;easing:linear")
205     obj.setAttribute('animation__sp',"property:position;from:10 0 0;to:14
↵ 4.5 0; startEvents:sph;pauseEvents:pause;resumeEvents:resume;dur:"+
↵ gap+";loop:false;easing:linear")
206
207     obj.addEventListener("animationcomplete__start",()=>{obj.emit(((drop)?(
↵ "d1"):( "p1")))}))
208     obj.addEventListener("animationcomplete__drop1",()=>{obj.emit("d2")})
209     obj.addEventListener("animationcomplete__drop2",()=>{obj.emit("d3")})
210     obj.addEventListener("animationcomplete__drop3",()=>{obj.remove()})
211     obj.addEventListener("animationcomplete__1",()=>{obj.emit(((sphere)?("s
↵ ph"):( "cube")))}))
212     obj.addEventListener("animationcomplete__sp",()=>{obj.remove();check(ob
↵ j.getAttribute("color"),form,("#"+((red)?("FF"):( "00"))+((green)?("
↵ FF"):( "00"))+((blue)?("FF"):( "00"))), 'a-sphere')})
213     obj.addEventListener("animationcomplete__cube",()=>{obj.remove();
↵ check(obj.getAttribute("color"),form,("#"+((red)?("FF"):( "00"))+((g
↵ reen)?("FF"):( "00"))+((blue)?("FF"):( "00"))), 'a-box')})
214
215     obj.emit("start")
216
217     return obj
218 }
219
220 var is_game_started = false
221 startB.addEventListener("click",function(){
222     if(!is_game_started){
223         updateColor()
224         updateDrop()
225         updateForm()
226         updateScore()
227         var temptime = 0

```



```

228     var tempcount = 0
229     function tick(){
230         temptime += 100
231         if (temptime >=spawntime){
232             generator(spawntime,sp)
233             temptime = 0
234             tempcount += 1
235         }
236         if (tempcount >= reduce_sp_counttoreduce)
237         {
238             spawntime-= reduce_spawntime
239             tempcount = 0
240         }
241     }
242     bgm.play()
243     ticker = setInterval(() => {
244         tick(100)
245     }, 100);
246     is_game_started=true
247 }
248 })
249 })
250
251 </script>
252 </head>
253 <body>
254     <a-scene follow
255         vr-mode-ui="enabled: false;"
256         renderer="logarithmicDepthBuffer: true;"
257         embedded
258         arjs="trackingMethod: best; sourceType: webcam;debugUIEnabled: false;
259         ↪ detectionMode: mono_and_matrix; matrixCodeType: 3x3;"
260     >
261     <a-marker id="game" type="barcode" value="3">
262     <a-entity scale = "0.2 0.2 0.2">
263     <a-plane color="#ffffff" rotation="-90 0 0 " width="20" height="20">
264     <a-box color="#bbbbbb" width="1.5" height="1.5" position="-9 0 0.5" id
265     ↪ ="startPoint">
266
267     </a-box>
268
269     <a-entity>
270     <a-box color="#888888" position="-3.5 0 0.1" scale="10 1.2 0.1"></a-box>
271     <a-box color="#888888" position="-6 6 0.1" scale="7.2 1.2 0.1"></a-box>
272     <a-box color="#888888" position="-3 3 0.1" scale="1.2 5 0.1"></a-box>
273     <a-box color="#888888" position="-9 3 0.1" scale="1.2 5 0.1"></a-box>
274     <a-box color="#888888" position="3 2 0.1" rotation="0 0 -45" scale="1.2
275     ↪ 5 0.1"></a-box>
276     <a-box color="#888888" position="3 -2 0.1" rotation="0 0 45" scale="1.2
277     ↪ 5 0.1"></a-box>
278     </a-entity>
279
280     <a-entity id="drop_gate" position = "-2.2 0.8 0" rotation="0 0 90">
281     <a-sphere scale="0.1 0.1 2"></a-sphere>
282     <a-box color="#ff0000" position="0 0.7 0" width="0.2" height="1.7"
283     ↪ depth="1.7" rotation="0 0 0"></a-box>
284     </a-entity>
285
286     <a-entity id="form_gate" position = "1.8 0 0" rotation="0 0 45">

```

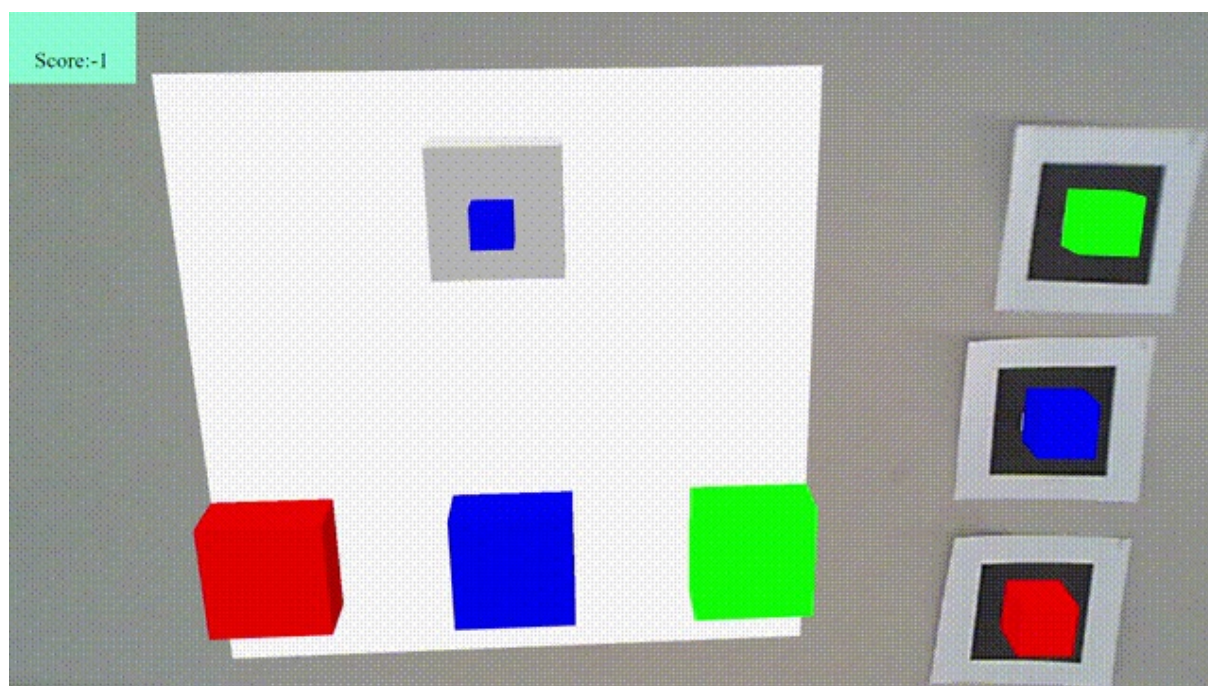
```

283     <a-sphere scale="0.1 0.1 2"></a-sphere>
284     <a-box color="#ff0000" position="0 0.7 0" width="0.2" height="1.7"
      ↪ depth="1.7" rotation="0 0 0"></a-box>
285 </a-entity>
286
287 <a-box color="#bbbbbb" position="5 -4 0.5" width="1.7" height="1.7"
      ↪ depth="1.7" rotation="0 0 -45">
288     <a-box id ="box_fetch" position="2.5 0 2" color="#FF0000" scale="4 4
      ↪ 5"></a-box>
289 </a-box>
290 <a-cylinder color="#bbbbbb" position="5 4 0.5" width="1.7" height="1.7"
      ↪ rotation="0 0 -45">
291     <a-box id ="sphere_fetch" position="0 2.5 2" color="#FF0000" scale="4 4
      ↪ 5"></a-box>
292 </a-box>
293
294 </a-plane>
295 </a-entity>
296
297 </a-marker>
298
299
300 <a-marker id="greenM" type="barcode" value="4">
301     <a-box scale="0.5 0.5 0.5" color="#00FF00" ></a-box>
302 </a-marker>
303 <a-marker id="redM" type="barcode" value="5">
304     <a-box scale ="0.5 0.5 0.5" color="#FF0000" ></a-box>
305 </a-marker>
306 <a-marker id="blueM" type="barcode" value="6">
307     <a-box scale="0.5 0.5 0.5" color="#0000ff" ></a-box>
308 </a-marker>
309 <a-marker id="shapeM" type="barcode" value="8">
310     <a-cylinder scale="1 3 1" color="#000000" ></a-box>
311 </a-marker>
312 <a-marker id="dropM" type="barcode" value="9">
313     <a-cylinder scale="1 3 1" color="#0000ff" ></a-box>
314 </a-marker>
315
316 <a-entity camera></a-entity>
317 </a-scene>
318 <div id="score"><span>Score:0</span></div>
319 </div>
320 <div id="startbutton"><span>Start</span></div>
321 </div>
322 </body>
323 <script>
324
325 </script>
326 </html>

```

## Этап 2. Конструкция по инструкции

### Игра по сценарию «Конвейер для кубитоклобусов»



Представим себе, что виртуальный мир населен некоторыми существами — кубитоклобусами.

Кубитоклобусы бывают круглые и некруглые (кубические, пирамидальные), а также у них есть цвет: красный, желтый, синий.

Кубитоклобусы попадают в реальный мир, через портал между двумя мирами цифровым и физическим в момент их конвергенции.

Наша задача состоит в отправке кубитоклобусов обратно в цифровой мир через специальные приемники. Приемники настроены на форму (круглые и некруглые) и цвет (красный, синий, желтый).

Кубитоклобус появляется в нашем мире через специальный портал (стартовый маркер). Используя маркеры формы и цвета, мы должны направить кубитоклобусы в нужный тип приемника (см. видео пример базовой реализации игры без учета счетчиков).

Рядом с приемником мы видим очки — кол-во отправленных назад кубитоклобусов правильного цвета и формы. При направлении в определенный приемник кубитоклобуса несоответствующей формы и (или) цвета. Конвейер ломается, все счетчики обнуляются, процесс запускается заново.

Кроме кубитоклобусов в наш мир могут попытаться проникнуть некубитоклобусы: существа неопределенной формы и непонятного (нечистого) цвета. Чтобы заработать больше баллов, мы вводим маркеры исправления формы и цвета для таких существ.

Однако скорость появления цифровых аборигенов в нашем мире возрастает и иногда бывает выгоднее уничтожить неопределенное существо, чтобы спасти остальных кубитоклобусов.

---

### Составляющие базовой (30 баллов):

1. Стартовый маркер — на котором генерируется кубитоклобус.
2. Маркер цвета для направления кубитоклобуса в приемник соответствующего цвета.
3. Счетчик отправленных кубитоклобусов (при неправильном отправлении кубитоклобуса игроку начисляется с 0 баллов, за верно отправленный 1).

*\*пример данной реализации в видео выше.*

### Составляющие продвинутой реализации (15 баллов):

1. Добавление формы кубитоклобусам.
2. Маркер распределения кубитоклобусов по форме.
3. При попадании кубитоклобуса формы несоответствующей приемнику — игра заканчивается. При этом, если цвет не верный, то игроку просто не начисляются баллы.

*\*должна быть выполнена полностью базовая составляющая.*

### Реализация Некубитоклобусов (20 баллов):

1. Маркер для конвертирования кубитоклобуса в нужную форму и(или) цвет.
2. Маркер уничтожения некубитоклобуса.
3. Счетчик жизней работающего конвейера (При неверной форме — 1 жизнь, Всего жизней 5, присутствует отображение количества жизней пользователю).
4. Прекращение игры при попадании некубитоклобуса в приемник.

### Дополнительные «фишки»:

- Ускорение скорости появления кубитоклобусов со временем (5 баллов).
- Ограничение процесса игры по времени от 1 до 2 минут (5 баллов).
- Вывод счета по окончании игры (5 баллов).
- Авторская стилизация маркеров, фигур и интерфейса (10 баллов).
- Добавление фоновой музыки (2 балла), музыкальных эффектов (3 баллов).
- Счетчик жизней — отображение пользователю на экране и дополнительные условия по их потери (5 баллов).

*\*для реализаций не предусматривающих счетчик.*

Ссылка на презентацию: [https://disk.yandex.ru/d/\\_0P3dJ3zsbjGDA/\T2A\CYRK\T2A\cyro\T2A\cyrn\T2A\cyrv\T2A\cyre\T2A\cyrishrt\T2A\cyre\T2A\cyrr.pptx](https://disk.yandex.ru/d/_0P3dJ3zsbjGDA/\T2A\CYRK\T2A\cyro\T2A\cyrn\T2A\cyrv\T2A\cyre\T2A\cyrishrt\T2A\cyre\T2A\cyrr.pptx).

В ответ необходимо прислать: код проекта, маркеры и видео работающей версии.

### Решение

Ссылка на репозиторий с решением задачи: <https://github.com/SergeyRusak/ARSortGame>.

`./index.html`

```
1 <html>
2   <head>
3     <style>
4       .buttons{
5         position: fixed;
6         bottom: 0px;
```

```
7     left: 0%;
8
9   }
10  .buttons div{
11    background-color: aquamarine;
12    width: 10vh;
13    height: 10vh;
14
15  }
16  .buttons div span{
17    position: relative;
18    top:50%;
19    left:20%;
20    font-size: 20pt;
21  }
22  #score{
23    background-color: aquamarine;
24    width: 10vw;
25    height: 20vh;
26    position: fixed;
27    top:0px;
28    left:0px;
29  }
30  #score span{
31    position: relative;
32    top:50%;
33    left:20%;
34    font-size: 25pt;
35  }
36  #startbutton{
37    background-color: aquamarine;
38    width: 10vw;
39    height: 10vh;
40    position: fixed;
41    bottom:5vh;
42    left:50vw;
43  }
44  #startbutton span{
45    position: relative;
46    top:25%;
47    left:20%;
48    font-size: 25pt;
49  }
50
51
52  </style>
53  <script src="https://aframe.io/releases/1.4.0/aframe.min.js"></script>
54  <script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe.js"
55  ↪ e-ar.js"></script>
56  <script>
57  var spawntime = 5000
58  var reduce_spawntime = 100
59  var reduce_sp_counttoreduce = 5
60  var is_started = false
61  var is_gameover = false;
62  var life = 5;
63  var score = 0;
64  var drop = false;
65  var sphere = false;
66  var red = false;
```

```

66     var green = false;
67     var blue = false;
68     var bgm = new Audio('bgm.mp3');
69     bgm.volume=0.05
70     var lbgm = new Audio('lbgm.mp3')
71     lbgm.volume = 0.05
72     var sfxscore = new Audio('getscore.mp3')
73     var sfxlife = new Audio('looselife.mp3')
74     var ticker
75     document.addEventListener("DOMContentLoaded",() =>{
76         var startB = document.querySelector("#startbutton")
77         var game = document.querySelector("#game")
78         var cubeF = document.querySelector("#box_fetch")
79         var sphereF = document.querySelector("#sphere_fetch")
80         var dropG = document.querySelector("#drop_gate")
81         var formG = document.querySelector("#form_gate")
82         var scoreTab = document.querySelector("#score")
83         function updateColor(){
84             console.log(cubeF)
85             let color = "#"+((red)?("ff"):( "00"))+((green)?("ff"):( "00"))+((blue)?("f
↵ f"):( "00"));
86             color = (color == "#ffffff")?("#aaaaaa"):(color
↵ == "#000000")?("#555555"):(color)
87             cubeF.setAttribute("material","color",color)
88             sphereF.setAttribute("material","color",color)
89         }
90         function updateDrop(){
91             dropG.setAttribute("rotation","0 0 "+((drop)?("180"):( "90")))
92         }
93         function updateForm(){
94             console.log(sphere)
95             formG.setAttribute("rotation","0 0 "+((sphere)?("135"):( "45")))
96         }
97         function updateScore(){
98             scoreTab.children[0].innerHTML = "Score:"+score+"<br>"+"Life:"+life
99         }
100        var redM = document.querySelector("#redM");
101        var greenM = document.querySelector("#greenM");
102        var blueM = document.querySelector("#blueM");
103        var shapeM = document.querySelector("#shapeM");
104        var dropM = document.querySelector("#dropM");
105        var sp = document.querySelector("#startPoint")
106
107        game.addEventListener("markerFound",function(){
108            is_started = true
109        })
110        game.addEventListener("markerLost",function(){
111            is_started = false
112        })
113
114        redM.addEventListener("markerFound",function(){
115            red = true
116            updateColor()
117        })
118        redM.addEventListener("markerLost",function(){
119            red = false
120            updateColor()
121        })
122        greenM.addEventListener("markerFound",function(){
123            green = true

```

```
124     updateColor()
125   })
126   greenM.addEventListener("markerLost",function(){
127     green = false
128     updateColor()
129   })
130   blueM.addEventListener("markerFound",function(){
131     blue = true
132     updateColor()
133   })
134   blueM.addEventListener("markerLost",function(){
135     blue = false
136     updateColor()
137   })
138   shapeM.addEventListener("markerFound",function(){
139     sphere = true
140     updateForm()
141   })
142   shapeM.addEventListener("markerLost",function(){
143     sphere = false
144     updateForm()
145   })
146   dropM.addEventListener("markerFound",function(){
147     drop = true
148     updateDrop()
149   })
150   dropM.addEventListener("markerLost",function(){
151     drop = false
152     updateDrop()
153   })
154
155   function check(obj_color,obj_form,base_color,base_form){
156     if(is_started){
157       console.log(obj_color, base_color)
158       if(obj_form == 'a-torus')
159       {
160         life = 0
161         gameover()
162       }
163       else
164         if(obj_form != base_form){
165           sfxlife.play()
166           life -= 1
167           if (life<= 0)
168           {
169             gameover()
170           }
171         }
172       else
173         if (obj_color == base_color){
174           sfxscore.play()
175           score += 1
176         }
177       updateScore()
178     }
179   }
180   function gameover(){
181     if(!is_gameover){
182       is_gameover = true
183       clearInterval(ticker)
```



```

184     console.log("LOSE")
185     game.children[0].children[0].setAttribute('color','#FF0000')
186     bgm.pause()
187     lbgm.play()
188     sp.innerHTML=""
189   }
190 }
191 function generator(gap,sp){
192   let form = (Math.random()>=0.5)?('a-box'):( 'a-sphere')
193   form = (Math.random()<=0.2)?('a-torus'):(form)
194   let color = "#"+((Math.random()>=0.5)?("FF"):( "00"))+((Math.random()>=0.5)
↵ )?("FF"):( "00"))+((Math.random()>=0.5)?("FF"):( "00"))
195   var obj = document.createElement(form)
196   sp.appendChild(obj)
197   obj.setAttribute('color',color)
198   obj.setAttribute('scale',"0.5 0.5 0.5")
199   obj.setAttribute('animation__start',"property:position;from:0 0 0;to:6
↵ 0 0;dur:"+gap+";loop:false;easing:linear")
200   obj.setAttribute('animation__1',"property:position;from:6 0 0;to:10 0
↵ 0; startEvents:p1;dur:"+gap+";loop:false;easing:linear")
201   obj.setAttribute('animation__drop1',"property:position;from:6 0 0;to:6
↵ 6 0; startEvents:d1;pauseEvents:pause;resumeEvents:resume;dur:"+gap
↵ "+";loop:false;easing:linear")
202   obj.setAttribute('animation__drop2',"property:position;from:6 6 0;to:0
↵ 6 0; startEvents:d2;pauseEvents:pause;resumeEvents:resume;dur:"+gap
↵ "+";loop:false;easing:linear")
203   obj.setAttribute('animation__drop3',"property:position;from:0 6 0;to:0
↵ 0 0; startEvents:d3;pauseEvents:pause;resumeEvents:resume;dur:"+gap
↵ "+";loop:false;easing:linear")
204   obj.setAttribute('animation__cube',"property:position;from:10 0 0;to:14
↵ -4.5 0; startEvents:cube;pauseEvents:pause;resumeEvents:resume;dur:
↵ "+gap+";loop:false;easing:linear")
205   obj.setAttribute('animation__sp',"property:position;from:10 0 0;to:14
↵ 4.5 0; startEvents:sph;pauseEvents:pause;resumeEvents:resume;dur:"+
↵ gap+";loop:false;easing:linear")
206
207   obj.addEventListener("animationcomplete__start",()=>{obj.emit(((drop)?(
↵ "d1"):( "p1")))}))
208   obj.addEventListener("animationcomplete__drop1",()=>{obj.emit("d2")})
209   obj.addEventListener("animationcomplete__drop2",()=>{obj.emit("d3")})
210   obj.addEventListener("animationcomplete__drop3",()=>{obj.remove()})
211   obj.addEventListener("animationcomplete__1",()=>{obj.emit(((sphere)?("s
↵ ph"):( "cube")))}))
212   obj.addEventListener("animationcomplete__sp",()=>{obj.remove();check(ob
↵ j.getAttribute("color"),form,("#"+((red)?("FF"):( "00"))+((green)?("
↵ FF"):( "00"))+((blue)?("FF"):( "00"))), 'a-sphere')})
213   obj.addEventListener("animationcomplete__cube",()=>{obj.remove();
↵ check(obj.getAttribute("color"),form,("#"+((red)?("FF"):( "00"))+((g
↵ reen)?("FF"):( "00"))+((blue)?("FF"):( "00"))), 'a-box')})
214
215   obj.emit("start")
216
217   return obj
218 }
219
220 var is_game_started = false
221 startB.addEventListener("click",function(){
222   if(!is_game_started){
223     updateColor()
224     updateDrop()

```



```

225     updateForm()
226     updateScore()
227     var temptime = 0
228     var tempcount = 0
229     function tick(){
230         temptime += 100
231         if (temptime >=spawntime){
232             generator(spawntime,sp)
233             temptime = 0
234             tempcount += 1
235         }
236         if (tempcount >= reduce_sp_counttoreduce)
237         {
238             spawntime-= reduce_spawntime
239             tempcount = 0
240         }
241     }
242     bgm.play()
243     ticker = setInterval(() => {
244         tick(100)
245     }, 100);
246     is_game_started=true
247 }
248 })
249 })
250
251 </script>
252 </head>
253 <body>
254     <a-scene follow
255         vr-mode-ui="enabled: false;"
256         renderer="logarithmicDepthBuffer: true;"
257         embedded
258         arjs="trackingMethod: best; sourceType: webcam;debugUIEnabled: false;
259         ↪ detectionMode: mono_and_matrix; matrixCodeType: 3x3;"
260     >
261     <a-marker id="game" type="barcode" value="3">
262     <a-entity scale = "0.2 0.2 0.2">
263     <a-plane color="#ffffff" rotation="-90 0 0 " width="20" height="20">
264     <a-box color="#bbbbbb" width="1.5" height="1.5" position="-9 0 0.5" id
265     ↪ ="startPoint">
266
267     </a-box>
268
269     <a-entity>
270     <a-box color="#888888" position="-3.5 0 0.1" scale="10 1.2 0.1"></a-box>
271     <a-box color="#888888" position="-6 6 0.1" scale="7.2 1.2 0.1"></a-box>
272     <a-box color="#888888" position="-3 3 0.1" scale="1.2 5 0.1"></a-box>
273     <a-box color="#888888" position="-9 3 0.1" scale="1.2 5 0.1"></a-box>
274     <a-box color="#888888" position="3 2 0.1" rotation="0 0 -45" scale="1.2
275     ↪ 5 0.1"></a-box>
276     <a-box color="#888888" position="3 -2 0.1" rotation="0 0 45" scale="1.2
277     ↪ 5 0.1"></a-box>
278     </a-entity>
279
280     <a-entity id="drop_gate" position = "-2.2 0.8 0" rotation="0 0 90">
281     <a-sphere scale="0.1 0.1 2"></a-sphere>
282     <a-box color="#ff0000" position="0 0.7 0" width="0.2" height="1.7"
283     ↪ depth="1.7" rotation="0 0 0"></a-box>

```

```
280     </a-entity>
281
282     <a-entity id="form_gate" position = "1.8 0 0" rotation="0 0 45">
283         <a-sphere scale="0.1 0.1 2"></a-sphere>
284         <a-box color="#ff0000" position="0 0.7 0" width="0.2" height="1.7"
285             ↪ depth="1.7" rotation="0 0 0"></a-box>
286     </a-entity>
287
288     <a-box color="#bbbbbb" position="5 -4 0.5" width="1.7" height="1.7"
289     ↪ depth="1.7" rotation="0 0 -45">
290         <a-box id ="box_fetch" position="2.5 0 2" color="#FF0000" scale="4 4
291         ↪ 5"></a-box>
292     </a-box>
293
294     <a-cylinder color="#bbbbbb" position="5 4 0.5" width="1.7" height="1.7"
295     ↪ rotation="0 0 -45">
296         <a-box id ="sphere_fetch" position="0 2.5 2" color="#FF0000" scale="4 4
297         ↪ 5"></a-box>
298     </a-box>
299
300     </a-plane>
301 </a-entity>
302
303 </a-marker>
304
305
306 <a-marker id="greenM" type="barcode" value="4">
307     <a-box scale="0.5 0.5 0.5" color="#00FF00" ></a-box>
308 </a-marker>
309 <a-marker id="redM" type="barcode" value="5">
310     <a-box scale ="0.5 0.5 0.5" color="#FF0000" ></a-box>
311 </a-marker>
312 <a-marker id="blueM" type="barcode" value="6">
313     <a-box scale="0.5 0.5 0.5" color="#0000ff" ></a-box>
314 </a-marker>
315 <a-marker id="shapeM" type="barcode" value="8">
316     <a-cylinder scale="1 3 1" color="#000000" ></a-box>
317 </a-marker>
318 <a-marker id="dropM" type="barcode" value="9">
319     <a-cylinder scale="1 3 1" color="#0000ff" ></a-box>
320 </a-marker>
321
322     <a-entity camera></a-entity>
323 </a-scene>
324 <div id="score"><span>Score:0</span></div>
325 </div>
326 <div id="startbutton"><span>Start</span></div>
327 </div>
328 </body>
329 <script>
330
331 </script>
332 </html>
```

---

## *Этап 3. Своя игра. Сборка итогового проекта*

### *Своя игра*

Командам участников необходимо предложить сценарий с частичной или полной реализацией своей игры. К игре предъявляются следующие требования:

1. игра должна демонстрировать функциональные возможности технологий дополненной реальности в веб-пространстве;
2. игра должна быть построена с использованием ранее разработанных (во время этапа 1) программных компонентах;
3. игра должна быть реализована по единому сценарию, а не представлять из себя набор отдельных компонентов;
4. игра должна иметь оригинальный дизайн всех составляющих ее элементов.

В качестве своей игры команды участников могут сделать расширение сценария игры, предложенной разработчиками профиля на втором этапе.

В качестве игры может быть предложен новый сценарий и дизайн игровых элементов для механики, предложенной разработчиками на втором этапе.

### *Сборка итогового проекта*

Все программные компоненты и готовые приложения, собранные на их основе, должны быть собраны командами участниками в единый веб-ресурс «Сумма технологий». На ресурсе необходимо разместить три модуля материалов, состоящих из примеров кода, видео-демонстраций работы, возможности проверить работоспособность примера опытным путем. Первый модуль посвящен реализации отдельных базовых элементов webAR-разработки — от прикрепления к маркеру мультимедийных объектов, до программных компонентов реализующих взаимодействие с объектами дополненной реальности. Во втором модуле должны быть размещены видеодемонстрация, коды с комментариями, маркеры и собственно приложение для игры «Кубитоклобусы». Третий модуль материалов веб-ресурса — игра, предложенная участниками по собственному сценарию, или модификация ранее реализованной игры «Кубитоклобусы».

В качестве примера прикрепляется решение команды «Примитив ограниченной дереализации».

Ссылка на репозиторий: [https://gitlab.com/nto\\_ar\\_2023/pod/](https://gitlab.com/nto_ar_2023/pod/).

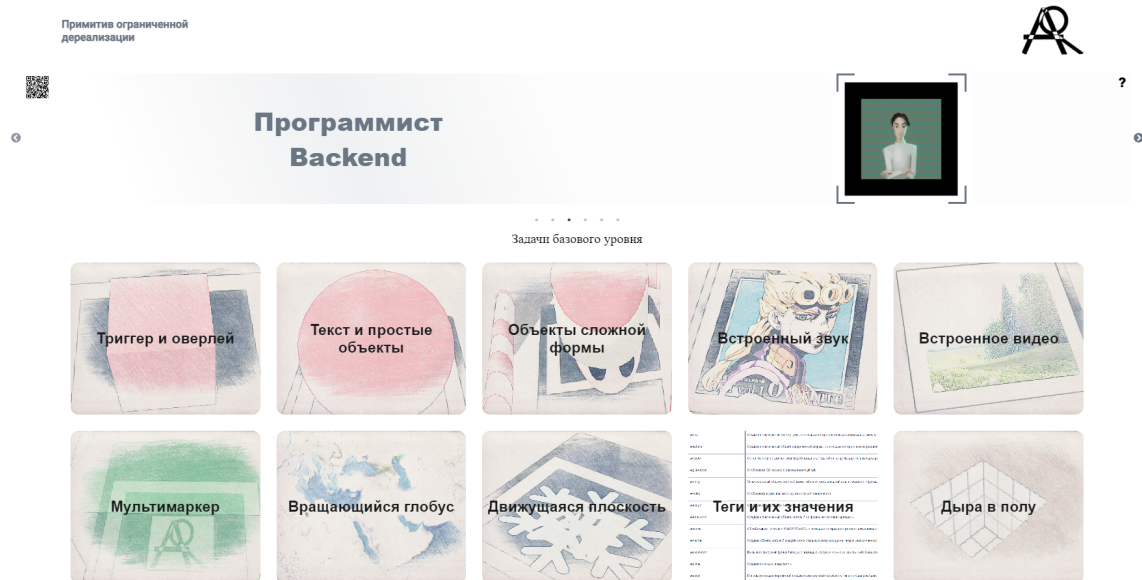


Рис. VI.2.11. Главная страница

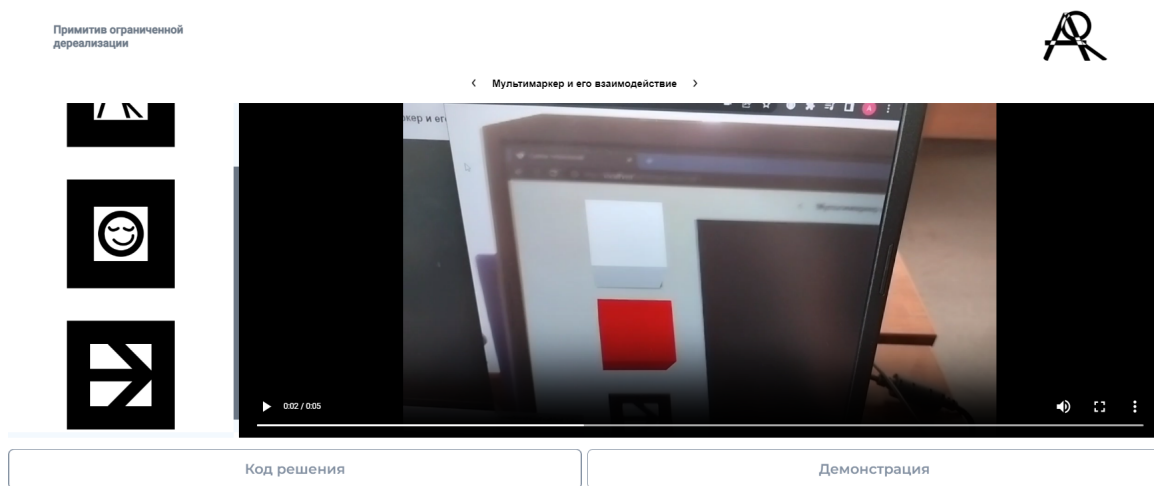


Рис. VI.2.12. Страница задачи

`./src/index.js`

```

1 import './App.css';
2 import './css/main.css';
3 import { BrowserRouter, Routes, Route } from "react-router-dom";
4 import HomePage from './pages/home';
5 import TaskPage from './pages/task';
6 import HtmlPage from './pages/html';
7 import Header from './components/Header';
8
9 function App() {
10   return (
11     <>
12
13     <BrowserRouter>
14     <Header></Header>
15
16     <Routes>
17       <Route path="/" element={<HomePage />} />
18       <Route path="/task" element={<TaskPage
19         />} />
20     </Routes>
21   </>
22 )
23 }

```

```

18         <Route path='/html' element={<HtmlPage
           ↪ />} />
19     </Routes>
20 </BrowserRouter>
21 </>
22     );
23 }
24
25 export default App;

```

*./src/App.js*

```

1  import './App.css';
2  import './css/main.css';
3  import { BrowserRouter, Routes, Route } from "react-router-dom";
4  import HomePage from './pages/home';
5  import TaskPage from './pages/task';
6  import HtmlPage from './pages/html';
7  import Header from './components/Header';
8
9  function App() {
10     return (
11         <>
12
13             <BrowserRouter>
14     <Header></Header>
15
16                 <Routes>
17     <Route path='/' element={<HomePage />} />
18     <Route path='/task' element={<TaskPage
           ↪ />} />
19     <Route path='/html' element={<HtmlPage
           ↪ />} />
20                 </Routes>
21             </BrowserRouter>
22         </>
23     );
24 }
25 export default App;

```

*./src/pages/home.jsx*

```

1  import { Link, useSearchParams } from "react-router-dom";
2  import Description from '../components/Description';
3  import Header from '../components/Header';
4
5  export default function TaskPage() {
6
7     const [params, setParams] = useSearchParams();
8
9     return (
10         <>
11
12             <Description
13     num={params.get("num")}
14     leftPage={params.get("left")}
15     rightPage={params.get("right")}
16     url={params.get("solutionUrl")}
17     video={params.get("videoUrl")}
18         ></Description>

```

```
19     );
20 }
```

*./src/pages/tasks.jsx*

```
1  import * as React from "react";
2  import { Link, useSearchParams } from "react-router-dom";
3  import Html from '../components/Html';
4
5
6  export default function HtmlPage() {
7
8    const [params, setParams] = useSearchParams();
9
10   return (
11     <>
12       <Html num={params.get('num')}></Html>
13     </>
14   );
15 }
```

*./src/pages/html.jsx*

```
1  import * as React from "react";
2  import { Link, useSearchParams } from "react-router-dom";
3  import Html from '../components/Html';
4
5
6  export default function HtmlPage() {
7
8    const [params, setParams] = useSearchParams();
9
10   return (
11     <>
12       <Html num={params.get('num')}></Html>
13     </>
14   );
15 }
```

## Материалы для подготовки

- Документация AR.js: <https://ar-js-org.github.io/AR.js-Docs/>.
- Документация AFrame: <https://aframe.io/>.
- Инструкция по аватарам: [https://youtu.be/yHY\\_9Fr2sg0](https://youtu.be/yHY_9Fr2sg0).
- Ссылка на сборник прошлого года: <https://ntcontest.ru/docs/ar-assignments.pdf>.
- Ссылка на образовательный курс: <https://stepik.org/course/123252/promo>.
- Ссылка на видео по структуре задаче финала: <https://youtu.be/4JfxLmhFtgQ>.
- Разбор задач второго тура: ссылка на видео и документ: <https://youtu.be/sOgyTZ4fFNE>.