



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации
информационных технологий

Отчет по практической работе №2

по дисциплине «Разработка кроссплатформенных мобильных
приложений»

Выполнил:

Студент группы ИКБО-32-22

Козеев Василий Александрович

Проверила:

Зырянова С.А.

МОСКВА 2025

Практическая работа 2

Установка Flutter и создания первого проекта

Тема

Виджеты. Дерево виджетов. Stateless Widget.

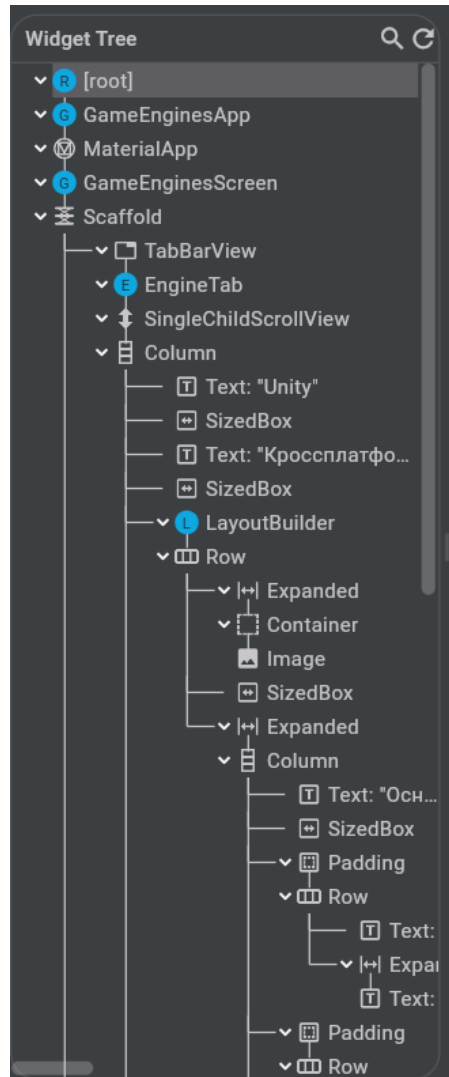
Задачи

Задание 1

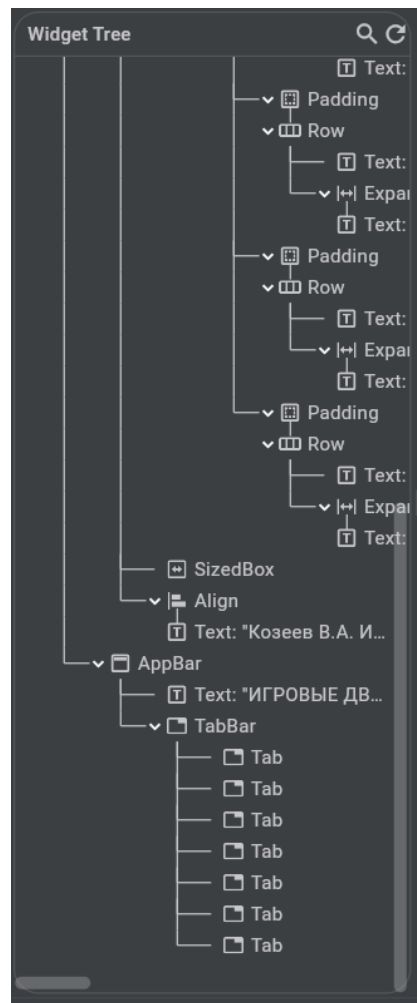
- 1) Выбрать тему (предметную область (ПО)) из предложенных ниже или свою и согласовать ее с преподавателем
- 2) Создать приложение для выбранной предметной области согласно примерному макету.
- 3) В отчет по практической работе внести программный код приложения, скриншот дерева виджетов приложения, скриншот результата.

Практическая часть

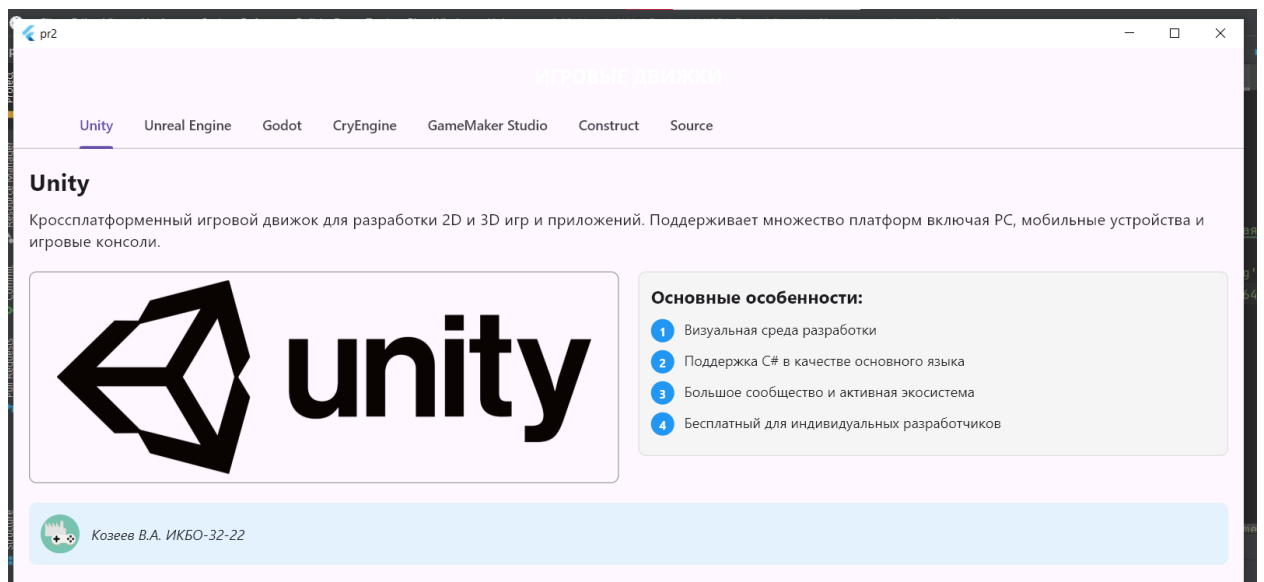
Тема – Игровые движки.



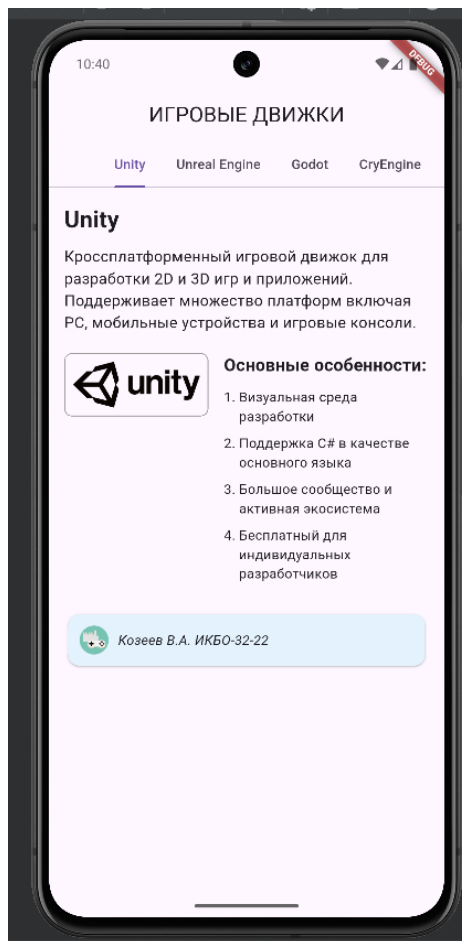
Скриншот 1 – Дерево виджетов приложения ч1



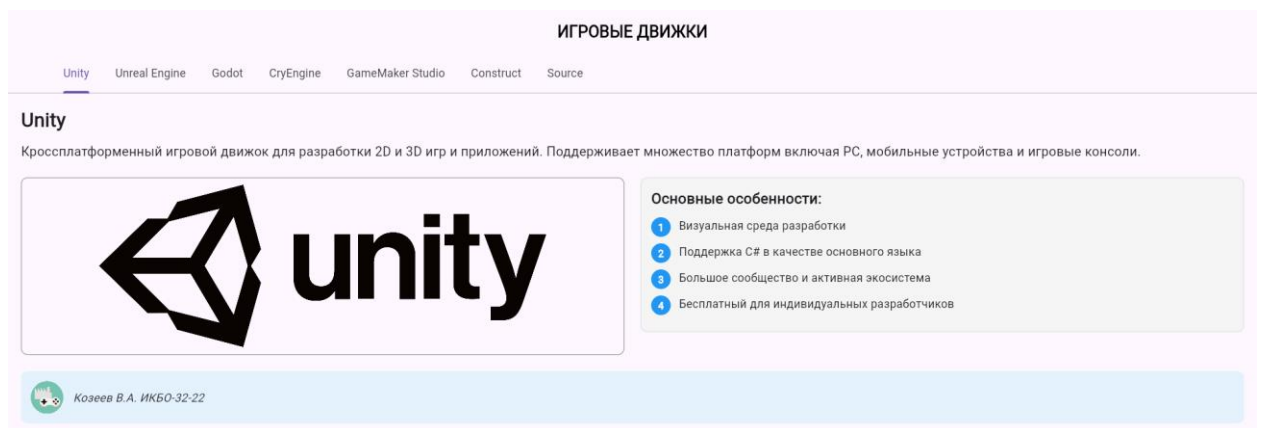
Скриншот 2 – Дерево виджетов приложения ч2



Скриншот 3 – Запуск в Windows



Скриншот 4 – Запуск на Android



Скриншот 5 – Запуск в web

Листинг 1 – main.dart:

```
import 'package:flutter/material.dart';

void main() {
  runApp(const GameEnginesApp());
}

class GameEngine {
  final String name;
  final String description;
  final String imageUrl;
  final List<String> features;
```

```

    GameEngine({
        required this.name,
        required this.description,
        required this.imageUrl,
        required this.features,
    });
}

class GameEnginesApp extends StatelessWidget {
    const GameEnginesApp({super.key});

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Игровые движки',
            theme: ThemeData(
                primarySwatch: Colors.blue,
            ),
            home: const GameEnginesScreen(),
        );
    }
}

class GameEnginesScreen extends StatefulWidget {
    const GameEnginesScreen({super.key});

    @override
    State<GameEnginesScreen> createState() => _GameEnginesScreenState();
}

class _GameEnginesScreenState extends State<GameEnginesScreen> with
SingleTickerProviderStateMixin {
    late TabController _tabController;

    final List<GameEngine> gameEngines = [
        GameEngine(
            name: 'Unity',
            description: 'Кроссплатформенный игровой движок для разработки 2D и 3D
игр и приложений. Поддерживает множество платформ включая PC, мобильные
устройства и игровые консоли.',
            imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Unity_Technologies
_logo.svg/2560px-Unity_Technologies_logo.svg.png',
            features: [
                'Визуальная среда разработки',
                'Поддержка C# в качестве основного языка',
                'Большое сообщество и активная экосистема',
                'Бесплатный для индивидуальных разработчиков',
            ],
        ),
        GameEngine(
            name: 'Unreal Engine',
            description: 'Мощный игровой движок с передовой графикой, используемый
для создания AAA-игр и визуализаций. Разработан Epic Games.',
            imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/2/20/Unreal_Engine_logo
_%282019%29.svg/1920px-Unreal_Engine_logo_%282019%29.svg.png',
            features: [
                'Высококачественная графика с Ray Tracing',
                'Система Blueprints для визуального программирования',
                'Поддержка C++',
                'Бесплатен с роялти после первого \$1 млн дохода',
            ],
        ),
    ],

```

```

    ),
    GameEngine(
        name: 'Godot',
        description: 'Открытый и бесплатный игровой движок с поддержкой 2D и 3D
графики. Имеет собственный язык программирования GDScript.',
        imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/6/6a/Godot_icon.svg/120
0px-Godot_icon.svg.png',
        features: [
            'Полностью открытый исходный код',
            'Собственный язык GDScript (похож на Python)',
            'Поддержка множества платформ',
            'Не требует роялти или подписки',
        ],
    ),
    GameEngine(
        name: 'CryEngine',
        description: 'Игровой движок с продвинутой графикой, известный по играм
серии Crysis. Разработан немецкой компанией Crytek.',
        imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/2/28/CryEngine_logo.svg
/1200px-CryEngine_logo.svg.png',
        features: [
            'Передовая технология глобального освещения',
            'Мощные инструменты для создания ландшафтов',
            'Поддержка виртуальной реальности',
            'Платная модель на основе подписки',
        ],
    ),
    GameEngine(
        name: 'GameMaker Studio',
        description: 'Игровой движок, ориентированный на 2D-разработку, с
упором на простоту использования и быстрое создание прототипов.',
        imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/b/b6/YoYo_Games_logo.sv
g/1200px-YoYo_Games_logo.svg.png',
        features: [
            'Визуальное программирование Drag & Drop',
            'Собственный язык GML (GameMaker Language)',
            'Быстрая разработка 2D игр',
            'Экспорт на множество платформ',
        ],
    ),
    GameEngine(
        name: 'Construct',
        description: 'Игровой движок, основанный на событиях, не требующий
знания программирования. Идеален для начинающих разработчиков.',
        imageUrl: 'https://www.construct.net/images/press/logo.svg',
        features: [
            'Программирование на основе событий',
            'Не требует знаний программирования',
            'Быстрое создание 2D игр',
            'Экспорт в HTML5 и другие платформы',
        ],
    ),
    GameEngine(
        name: 'Source',
        description: 'Игровой движок, разработанный Valve Corporation,
известный по играм Half-Life 2, Counter-Strike: Source и другим.',
        imageUrl:
'https://upload.wikimedia.org/wikipedia/commons/thumb/5/5c/Source_engine_logo
_and_wordmark.svg/1200px-Source_engine_logo_and_wordmark.svg.png',
        features: [
            'Продвинутая физика на основе Havok',

```



```

        child: Row(
          children: [
            Image.network(
              'https://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/Video-Game-Controller-Icon-IDV-green-industry.svg/800px-Video-Game-Controller-Icon-IDV-green-industry.svg.png',
              width: 30,
              height: 30,
              errorBuilder: (context, error, stackTrace) {
                return const Icon(Icons.sports_esports, size: 30,
color: Colors.blue);
              },
            ),
            const SizedBox(width: 10),
            const Expanded(
              child: Text(
                'Козеев В.А. ИКБО-32-22',
                style: TextStyle(fontSize: 14, fontStyle:
FontStyle.italic),
              ),
            ),
          ],
        ),
      ),
    ],
  ),
),
);
}

Widget _buildContentRow(GameEngine engine) {
  return Container(
    // Ограничиваем высоту контейнера
    constraints: BoxConstraints(
      maxHeight: 300, // Максимальная высота для всего блока
    ),
    child: Row(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        // Изображение с фиксированной шириной
        Container(
          width: 150, // Фиксированная ширина для изображения
          padding: const EdgeInsets.all(8.0),
          decoration: BoxDecoration(
            border: Border.all(color: Colors.grey),
            borderRadius: BorderRadius.circular(8.0),
          ),
          child: Image.network(
            engine.imageUrl,
            fit: BoxFit.contain,
            loadingBuilder: (context, child, loadingProgress) {
              if (loadingProgress == null) return child;
              return const Center(child: CircularProgressIndicator());
            },
            errorBuilder: (context, error, stackTrace) {
              return const Center(
                child: Icon(Icons.error, size: 50, color: Colors.red),
              );
            },
          ),
        ),
        const SizedBox(width: 16),
        // Список особенностей занимает оставшееся пространство

```

```

        Expanded(
            child: SingleChildScrollView(
                // Вертикальный скролл для списка, если он не помещается
                child: _buildFeatures(engine),
            ),
        ),
    ],
),
);
}

Widget _buildFeatures(GameEngine engine) {
    return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisAlignment: MainAxisAlignment.min,
        children: [
            const Text(
                'Основные особенности:',
                style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
            ),
            const SizedBox(height: 10),
            ...engine.features.asMap().entries.map((entry) {
                final index = entry.key;
                final feature = entry.value;
                return Padding(
                    padding: const EdgeInsets.only(bottom: 8.0),
                    child: Row(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            Text('${index + 1}. '),
                            Expanded(child: Text(feature)),
                        ],
                    ),
                );
            }).toList(),
        ],
    );
}
}

```