**План**

**Цель:**

Познакомиться с разработкой Android-игр

**Результат:**

Созданный проект в Android Studio для игры и полученные начальные представления об Android-разработке игр

**Программа:**

1. Фреймворк LibGDX
2. Что такое API
3. Back-end
4. Модули
5. Жизненный цикл

**Структура организованной клубной деятельности:**

1. Знакомство с материалом (теория)

2. Описание результата (описание работы программы)

3. Разработка программы

4. Тестирование (испытание программы)

5. Подведение итогов

Ход знакомства с материалами:

1. Организационный момент
2. Материал для ознакомления

**Фреймворк LibGDX**

Фреймворк – это заготовка или шаблон, определяющий структуру программного обеспечения.

LibGDX – это фреймворк для визуализации и разработки игр, основанный на языке программирования Java. Он позволяет создавать игры для таких платформ, как Windows, Linux, MacOS, Android, IOS.

Чтобы скачать LibGDX, нужно перейти на сайт <https://libgdx.badlogicgames.com/>

**API**

API (Application Programming Interface) – это интерфейс программирования, интерфейс создания приложения. Если говорить более понятным языком, то API – это готовый код для упрощения жизни программисту.

API создавался для того, чтобы программист реально мог облегчить задачу написания того или иного приложения благодаря использованию готового кода (например, функций).

**Back-end**

LibGDX позволяет сосредоточиться на нескольких платформах. Каждая платформа имеет различные механизмы, когда дело доходит до настройки окна приложения, обработки пользовательского ввода, взаимодействия с файловой системой, проигрыванием и т.д.

LibGDX пытается абстрагировать эти различия через множество интерфейсов, которые складывают специфику платформы. Для каждой платформы LibGDX поддерживает так называемые back-end и реализует эти интерфейсы. Разработчик приложения не связан с back-end напрямую, вместо этого разработка ведется через интерфейсы.

В настоящее время LibGDX поддерживает 4 разных back-end:

* LWJGL
* JOGL
* Android
* HTML 5

**Модули**

В основе ядра LibGDX лежит пять интерфейсов, предоставляющие взаимодействие с операционной системой. Каждый back-end реализует следующие интерфейсы:

* Application
* Files
* Input
* Audio
* Graphics

**Жизненный цикл**

LibGDX-приложение имеет определенный жизненный цикл, регулирующий состояния приложения (создание, приостановка, возобновление, визуализация, утилизация)

Разработчик приложения перехватывает события жизненного цикла с помощью реализации ApplicationListener интерфейса и передачи экземпляра этой реализации в Application реализацию конкретного back-end. С этого момента LibGDX-приложение будет вызывать ApplicationListener каждый раз, когда будут события уровня приложения.

Можно наследоваться от ApplicationAdapter класса, если нужны не все методы интерфейса.

1. Практическое задание

Создайте LibGDX-проект на основе просмотренного ранее видео.