

# Создание пустого проекта

Исполняемые модули находятся в репозитории [lm2.engine.ce](https://github.com/lm2-engine/ce). Файлы для проекта находятся в репозитории [lm2.examples](https://github.com/lm2-engine/examples). Необходимый минимум доступен по ссылкам:

- [win\\_empty\\_project\\_start.zip](#)
- [rpi\\_empty\\_project\\_start.zip](#)

## Создание структуры директорий

Создайте папку `empty_project`. Это корень проекта. Скопируйте в корень исполняемый модуль (исполняемый файл `.exe` и динамические библиотеки `.dll` необходимые для старта платформы в случае Windows или скомпилированный бинарный файл в случаях систем на основе ядра Linux).

Создайте в корне папку `images`. Это каталог для хранения рендеров. Далее мы укажем к нему путь в главном конфигурационном файле. Скопируйте в этот каталог файл `logo.png`.

Создайте в корне папку `configs`. Это каталог для хранения конфигурационных файлов (кроме главного конфигурационного файла, он должен быть в корне проекта).

Создайте в корне папку `fonts`. Это каталог для используемых шрифтов. Далее мы подключим шрифты в главном конфигурационном файле. Скопируйте в этот каталог файл `LiberationSerif-Regular.ttf`.

## Создание файлов

### Главный конфигурационный файл

Основная статья «[Главный конфигурационный файл](#)». Создайте в корне проекта файл `config.xml`. Платформа по умолчанию к нему обратиться. Никаких дополнительных действий не требуется. Файл должен содержать следующий код:

```
<!-- объявление xml -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!-- корневой элемент -->
<root>
  <!-- элемент общих настроек, задаются следующие: язык, путь к рендерам,
   путь к конфигурационным файлам, флаг логирования -->
  <System
    Language="ru"
    ImagesPath="images/"
    ConfigsPath="configs/"
    Log="true"/>
  <!-- элемент логической машины, задаются следующие свойства: уникальный
   идентификатор, файл описание панелей, файл конфигурации окон, первый
   файл с lua-кодом для запуска -->
  <VM
    ID="main"
```

```

    Visual="panels.xml"
    StartupLayout="viewports.xml"
    StartupLogic="start.lua">
</VM>
<!-- элемент шрифта, задаются следующие свойства: уникальный идентификатор,
    полный путь к файлу -->
    <Font ID="Default" FileName="fonts/LiberationSerif-Regular.ttf" />
</root>

```

Напишите код сами, а не просто скопируйте его, так лучше запоминается.

## Файл конфигурации окон

Основная статья «[Конфигурация окон](#)». В папку configs создайте viewports.xml. Название файла должно совпадать с названием, указанным в главном конфигурационном файле. Определите окно на первом мониторе (если у вас несколько мониторов) с разрешением 800×480. Файл должен содержать следующий код:

```

<!-- объявление xml -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!-- корневой элемент, задается следующее свойство: разрешение -->
<root MonitorWidth="800" MonitorHeight="480">
    <!-- элемент вьюпорт, задаются следующие свойства: положение на мониторе,
        уникальный идентификатор -->
    <Viewport Monitor="0,0" ID="main_viewport" />
</root>

```

## Файл описания панелей

Основная статья «[Описание панелей](#)». Создайте в папке configs файл panels.xml. Название файла должно совпадать с названием, указанным в главном конфигурационном файле. Необходимо подключить библиотеку элементов, файл со строковыми данными, описать панель с фоном, логотипом и кнопкой. Файл должен содержать следующий код:

```

<!-- объявление xml -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!-- корневой элемент -->
<root>
    <!-- элемент подключающий строковые данные -->
    <Text FileName="text.xml"/>
    <!-- элемент подключающий библиотеку элементов -->
    <Library FileName="templates.xml"/>
    <!-- элемент родитель панелей -->
    <GUI>
        <!-- элемент описывающий панель, задается следующее свойство: размеры
            вьюпорта, на который будет выводиться панель -->
        <Panel ID="main" VPWidth="800" VPHeight="480">
            <Comment>Empty project main panel</Comment>
            <!-- элемент на панели, задаются следующие свойства: уникальный идентификатор,
                идентификатор элемента из библиотеки элементов -->
            <Entry ID="bg" TypeID="bg">

```

```

    <Comment>Panel background</Comment>
  </Entry>
  <!-- элемент на панели, задаются следующие свойства: уникальный идентификатор,
идентификатор элемента из библиотеки элементов, положение на панели -->
  <Entry ID="btExit" TypeID="button" Top="300" Left="300"/>
  <!-- элемент на панели, задаются следующие свойства: уникальный идентификатор,
идентификатор элемента из библиотеки элементов, положение на панели -->
  <Entry ID="Logo" TypeID="logo" Top="60" Left="60"/>
</Panel>
</GUI>
</root>

```

## Файл с библиотекой элементов

Основная статья «[Библиотека элементов](#)». Создайте в папке configs файл templates.xml. Название файла должно совпадать с названием, указанным в файле описания панелей. Необходимо описать элементы на панели. Файл должен содержать следующий код:

```

<!-- объявление xml -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!-- корневой элемент -->
<root>
  <!-- объявление библиотеки -->
  <Library>
    <!-- элемент библиотеки, задаются следующие свойства: уникальный идентификатор,
размеры элемента, -->
    <Element ID="bg" Width="800" Height="480">
      <Comment>Background</Comment>
      <!-- подэлемент прямоугольник, задаются следующие свойства: уникальный
идентификатор,
размеры и положение, цвет, цвет заливки, толщина контура, радиус закругления углов
-->
      <Rectangle ID="bg" Rect="50,50,700,380" Color="white" BGColor="black"
Thickness="1" CornerRadius="10"/>
      <!-- подэлемент текстовое поле, задаются следующие свойства: уникальный
идентификатор,
размер и положение, идентификатор, строковых данных, шрифт, кегль, выравнивание
-->
      <TextBox ID="text" Rect="full" TextID="txtEmptyProject"
Font="Default" Size="60" Align="Center" VAlign="Center"/>
    </Element>
    <!-- элемент библиотеки, задаются следующие свойства: уникальный идентификатор,
размеры элемента -->
    <Element ID="button" Width="200" Height="80">
      <!-- подэлемент прямоугольник, задаются следующие свойства: уникальный
идентификатор,
размеры и положение, цвет, цвет заливки, толщина контура, радиус закругления углов
-->
      <Rectangle ID="bg" Rect="full" Color="white" BGColor="black"
Thickness="2" CornerRadius="5"/>
      <!-- подэлемент текстовое поле, задаются следующие свойства: уникальный

```

```

идентификатор,
    размер и положение, идентификатор, строковых данных, шрифт, кегль, выравнивание -
->
    <TextBox ID="text" Rect="full" TextTID="txtExit" Font="Default"
Size="30" Align="Center" VAlign="Center"/>
</Element>
<!-- элемент библиотеки, задаются следующие свойства: уникальный идентификатор,
размеры элемента -->
<Element ID="logo" Width="369" Height="595" Scale="0.25">
    <!-- подэлемент изображение, задаются следующие свойства: размеры и положение,
путь к рендеру -->
    <Image Rect="full" Source="logo.png"/>
</Element>
</Library>
</root>

```

## Файл со строковыми данными

Основная статья «[Строковые данные](#)». Создайте в папке configs файл text.xml. Название файла должно совпадать с названием, указанным в файле описания панелей. Необходимо добавить все строковые поля. Файл должен содержать следующий код:

```

<!-- объявление xml -->
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!-- корневой элемент -->
<root>
    <!-- строковые данные, задается следующее свойство: уникальный идентификатор -->
    <Text ID="txtEmptyProject">
        <!-- языковая версия, задаются следующие свойства: язык, текст -->
        <Entry Lang="ru" Value="Пустой проект" />
        <Entry Lang="en" Value="Empty project" />
    </Text>
    <!-- строковые данные, задаются следующие свойства: уникальный идентификатор -->
    <Text ID="txtExit">
        <!-- языковая версия, задается следующее свойство: язык, текст -->
        <Entry Lang="ru" Value="Выход" />
        <Entry Lang="en" Value="Exit" />
    </Text>
</root>

```

## Lua-код

Основная статья «[Lua-код](#)». Создайте в папке configs файл start.lua. Название файла должно совпадать с названием, указанным в главном конфигурационном файле. Необходимо привязать панель к выводу, описать обработчик кнопки и привязать обработчик к событию. Файл должен содержать следующий код:

```

Panel('main_viewport', 'main')

function H_Exit(_ev)
    sys.Exit()

```

```
end  
AddEventHandler('main.btExit.Mouse.LeftPress', 'H_Exit')
```

## Запуск проекта

В следующих статьях описано как запускать проекты. Там же есть прямые ссылки на готовый пустой проект.

- [Установка и запуск платформы на Windows](#)
- [Установка и запуск платформы на Raspbian](#)

From:  
<https://wiki.lambda-mu.com/> - **Lambda-Mu Wiki**

Permanent link:  
[https://wiki.lambda-mu.com/lm2/ce/t\\_struct?rev=1596642085](https://wiki.lambda-mu.com/lm2/ce/t_struct?rev=1596642085)

Last update: **2020/11/30 14:12**

