Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Уральский федеральный университет**

**имени первого Президента России Б.Н.Ельцина»**

**Институт фундаментального образования**

**Кафедра интеллектуальных информационных технологий**

**«Сайт-конструктор для VR-приложения (игры)»**

Пояснительная записка

Руководитель Свинцов Д.В.

Группа ФО-350001

Студент Габайдуллин Д.Р.

Екатеринбург – 2018

**Введение**

В современных компьютерных играх / приложениях – клиенты (пользователи) предпочитают настройку приложения без участия программиста (автора), т.е. возможность изменять содержимое программы с помощью другой программы (в нашем случае – сайта).

Цель работы: Создание сайта – конструктора с возможностью изменять содержимое программы (игры).

Исходя из поставленной цели, я предположил выполнить следующие подзадачи:

* Выбор фреймворка внешнего вида сайта (выбор стал на bootsrap)
* Набросать прототип дизайна сайта (функционал, расположение элементов)
* Реализация функционала сайта
  + возможность загрузки файлов (изображения, аудио, видео, текст) на сервер
  + обновление файлов
  + сохранение конфигурации файлов сервера.

**Описание**

Данный сайт позволит пользователю загружать файлы (изображения, аудио, видео, текст) на сервер, просматривать текущие файлы и изменять их.

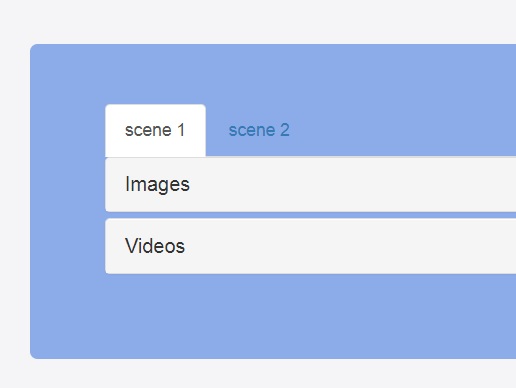
Чтобы понять смысл рассмотрим пример:

У нас есть игра, в которой есть помещения, в котором находятся определённые стэнды для Изображений/Видео (экраны, плакаты и тд.), к примеру можно увидеть на нижнем скриншоте место под видео/картинку (выделено красным цветом).



Пользователь (или владелец игры) хочет менять на данном плакате выводимую информацию без изменения исходников программы.

Сайт-конструктор позволяет загружать/изменять файлы на сервере, с которого игра затем будет подгружать (с интернета) и выводить их.



На сайте создаётся следующая иерархия:

Сцена 1:

-- Изображения:

---- Плакат слева на стене (загрузить)

---- Экран на потолке (загрузить)

-- Видео:

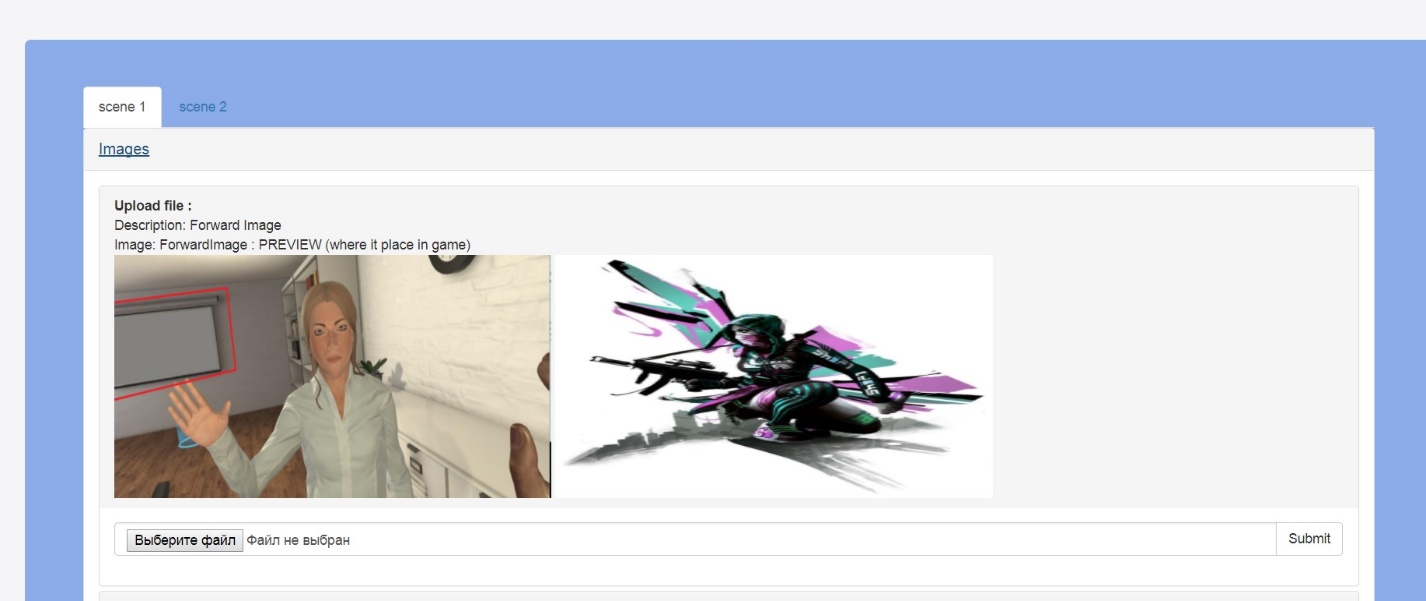
---- Телевизор на столе (загрузить)

-- Тесты(текст):

Сцена 2:

….

И т.д.



Как выглядит сайт:

Выбираем сцену, выбираем вкладку изображения и загружаем картинку которая будет затем показываться в игре в области выделенной на левой картинке. Загруженное изображение высвечивается справа.

После того как пользователь обновляет всю необходимую ему информацию, на сервере создаётся конфигурационный файл в формате .json, который будет хранить информацию: имя - ссылки на (изображения, видео и тд) и затем использоваться при загрузки игры. Игра на основе данного файла будет по web request’у подгружать все данные.

**Подробности реализации:**

Для создания данного сайта использовался Python 2.7, bootstrap, pyramid, jinja2.

Приложение имеет структуру данного вида:

datagame = { "scene 1" :

{

"Images": {

"LeftImage" : { ссылки, превью},

"RightImage" : {},

"ForwardImage" : {}

},

"Videos":

{

"LeftVideo" : {},

"RightVideo" : {},

"ForwardVideo" : {}

}

“Текст”

{}

И тд.

},

"scene 2" :

{

"Images": {

"LeftImage" : {},

"RightImage" : {},

"ForwardImage" : {}

},

"Videos":

{

"LeftVideo" : {},

"RightVideo" : {},

"ForwardVideo" : {}

}

}

}

На основе неё создается весь контент который затем преобразуется во внешний вид сайта.

**Выводы.**

Результатом курсовой работы стал сайт с возможность добавлять / изменять файлы (изображения, звуки, видео, текст) на сервере с помощью которых производится изменения «внешнего вида» другой программ.

**Список литературы.**

* [http://lectures.uralbash.ru](http://lectures.uralbash.ru/)
* <https://pylonsproject.org/>
* <https://www.python.org/>
* <http://google.com/>
* https://stackoverflow.com/