

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА ПРОГРАММНОЙ ИНЖЕНЕРИИ

ЛАБОРАТОРНАЯ РАБОТА № 6

по курсу: «Профессиональный практикум программной инженерии»
тема «Создание руководства программиста»

Выполнил: студент группы

Пи-156

Мазалов Роман

Донецк 2018

1. Цель лабораторной работы

Познакомиться с форматом оформления документации DocBook.

2. Файл .xml

```
<?xml version="1.0" encoding="UTF-8"?>
<book xml:id="simple_book" xmlns="http://docbook.org/ns/docbook" version="5.0">
  <info><title>Документация к проекту "Battletoads and Double Dragon Remake Web
Application"</title>
  <author><personname><firstname>Роман</firstname>
    <surname>Мазалов</surname></personname></author>
  <copyright><year>2018</year><holder>Мазалов Роман</holder></copyright></info>
  <preface><address>dellight55@gmail.com</address></preface>
  <chapter xml:id="chapter_1">
    <title>Краткое описание приложения</title>
    <para>Ремейк видео-игры <emphasis role="bold">Battle And Double Dragon</emphasis> для
NES.</para>
    <para><emphasis role="bold">Battletoads & Double Dragon: The Ultimate
Team</emphasis> — видеоигра в жанре Beat 'em up, разработанная компанией Rare и изданная
Tradewest в 1993 году. Изначально игра была выпущена для игровой консоли NES,
впоследствии вышли версии для SNES, Sega Mega Drive и Game Boy.</para>
    <para>Ремейк имеет следующие особенности:</para>
    <itemizedlist>
      <listitem>
        <para><emphasis role="bold">Максимальная схожесть:</emphasis> игра создана с
максимальной приближенностью к версии игры на NES, используются атласы, звуки,
музыкальное сопровождение оригинальной Battletoads And Double Dragon.</para>
      </listitem>
      <listitem>
        <para><emphasis role="bold">Доступность:</emphasis> ремейк представляет собой
Web-приложение, которое можно разместить на сервере для предоставления общего
доступа.</para>
      </listitem>
      <listitem>
        <para><emphasis role="bold">Использование современных Web-
технологий:</emphasis> ремейк использует технологии HTML5 и CSS3, фреймворк Vue.js для
динамического обновления структуры DOM, код приложения на JavaScript написан с
использованием стандарт ES5.</para>
      </listitem>
    </itemizedlist>
  </chapter>
  <chapter xml:id="chapter_2">
    <title>Горячие клавиши</title>
    <para>Клавиша "Стрелка вправо" - движение вперёд</para>
    <para>Клавиша "Стрелка влево" - движение назад</para>
    <para>Клавиша "Стрелка вверх" - движение налево</para>
    <para>Клавиша "Стрелка вниз" - движение направо</para>
    <para>Клавиша "Пробел" - прыжок</para>
  </chapter>
  <chapter xml:id="chapter_3">
    <title>Сценарии игры</title>
```

```

<section>
  <title>Столкновение с объектом</title>
<para>Игрок сталкивается с одним из препятствий по пути.</para>
  <example><title>Столкновение с препятствием</title>
    <programlisting>
      <![CDATA[
javascript
// obstacleOverlap()

if (this.isObstaclesSpawned) {
  this.player.crush()
  this.player.carfallAnimation.onComplete.add(this.game.gameOver, this)
}

// player.crush()

this.body.velocity.y = -100
this.body.velocity.x = -50
this.body.gravity.y = 50
this.animations.play('carfall', 6, false)
this.dx = 0
this.dy = 0
this.dj = 0
this.isPlayerCrushed = true]]>
    </programlisting>
  </example>
  <para>Игрок столкнулся с лутом, при этом увеличивается счет игрока.</para>
  <example><title>Столкновение с лутом</title>
    <programlisting>
      <![CDATA[
javascript
// lootOverlap()
if (!this.isLooted) {
  this.lootSprite.renderable = false
  this.createScoreAnimation(this.lootSprite.x, this.lootSprite.y, 2000)
  this.isLooted = true
  this.coinSound.play()
}]]>
    </programlisting>
  </example>
</section>
<section>
  <title>Окончание игры с победой</title>
  <para>Игрок прошел все волны препятствий.</para>
  <example><title>Победа игрока</title>
    <programlisting>
      <![CDATA[
javascript
// spawn()

...
if (this.waveCount > 0) {
  ...

```

```

    } else if (!this.player.isPlayerFinished) {
        this.player.finish()
        this.player.omgAnimation.onComplete.add(this.game.playerWon, this)
    }
...]]>
</programlisting>
</example>
</section>
<section>
    <title>Выбор другого игрового персонажа</title>
    <para>Игрок другого игрового персонажа в стартовом меню.</para>
    <example><title>Выбор игрового персонажа</title>
        <programlisting>
            <![CDATA[
javascript
// StartMenuPage3.vue selectCharacter(e)
if (e.keyCode === 37) {
    if (this.selectedCharacter > 0) {
        --this.selectedCharacter
    }
}

if (e.keyCode === 39) {
    if (this.selectedCharacter < this.CHARACTERS_COUNT - 1) {
        ++this.selectedCharacter
    }
}

// Game.js create()
..
this.player = new Player({
    game: this.game,
    x: 100,
    y: 0.4 * this.game.world.height,
    asset: 'battletoads',
    scale: this.scaleFactor,
    ground: this.ground,
    shadow: playerShadow,
    character: this.game.character,
})
...]]>
        </programlisting>
    </example>
</section>
</chapter>
<chapter xml:id="chapter_4">
    <title>Примеры работы программы</title>
    <figure>
        <title>Один из вариантов выбора персонажа</title>
        <mediaobject>
            <imageobject>
                <imagedata fileref="1.png"

```

```
        format="PNG" width="500px" />
    </imageobject>
</mediaobject>
</figure>
<figure>
    <title>Ещё один из вариантов выбора персонажа</title>
    <mediaobject>
        <imageobject>
            <imagedata fileref="2.png"
                format="PNG" width="500px" />
        </imageobject>
    </mediaobject>
</figure>
<figure>
    <title>Подбор лута</title>
    <mediaobject>
        <imageobject>
            <imagedata fileref="3.png"
                format="PNG" width="500px" />
        </imageobject>
    </mediaobject>
</figure>
<figure>
    <title>Один из вариантов поражения (1 часть)</title>
    <mediaobject>
        <imageobject>
            <imagedata fileref="4.png"
                format="PNG" width="500px" />
        </imageobject>
    </mediaobject>
</figure>
<figure>
    <title>Один из вариантов поражения (2 часть)</title>
    <mediaobject>
        <imageobject>
            <imagedata fileref="5.png"
                format="PNG" width="500px" />
        </imageobject>
    </mediaobject>
</figure>
</chapter>
</book>
```

3. Пример конвертации DocBook в .html

Документация к проекту "Battletoads and Double Dragon Remake Web Application"

Роман Мазалов

Copyright © 2018 Мазалов Роман

Table of Contents

- [1. Краткое описание приложения](#)
- [2. Горячие клавиши](#)
- [3. Сценарий игры](#)
 - [Столкновение с объектом](#)
 - [Окончание игры с победой](#)
 - [Выбор другого игрового персонажа](#)
- [4. Примеры работы программы](#)

List of Figures

- [4.1. Один из вариантов выбора персонажа](#)
- [4.2. Ещё один из вариантов выбора персонажа](#)
- [4.3. Подбор лута](#)
- [4.4. Один из вариантов поражения \(1 часть\)](#)
- [4.5. Один из вариантов поражения \(2 часть\)](#)

List of Examples

- [3.1. Столкновение с препятствием](#)
- [3.2. Столкновение с лутом](#)
- [3.3. Победа игрока](#)

Example 3.1. Столкновение с препятствием

```
javascript
// obstacleOverlap()

if (this.isObstaclesSpawned) {
    this.player.crush()
    this.player.carfallAnimation.onComplete.add(this.game.gameOver, this)
}

// player.crush()

this.body.velocity.y = -100
this.body.velocity.x = -50
this.body.gravity.y = 50
this.animations.play('carfall', 6, false)
this.dx = 0
this.dy = 0
this.dj = 0
this.isPlayerCrushed = true
```

Игрок столкнулся с лутом, при этом увеличивается счет игрока.

Example 3.2. Столкновение с лутом

```
javascript
// lootOverlap()
if (!this.isLooted) {
    this.lootSprite.renderable = false
    this.createScoreAnimation(this.lootSprite.x, this.lootSprite.y, 2000)
    this.isLooted = true
    this.coinSound.play()
}
```

Chapter 4. Примеры работы программы

Figure 4.1. Один из вариантов выбора персонажа



Figure 4.2. Ещё один из вариантов выбора персонажа

