УДК 004.946

Рабданова Венера Владимировна,

канд. экон. наук, зав. кафедрой;

Елтунова Инга Баировна,

канд. пед. наук, начальник отдела

дистанционного и дополнительного профессионального образования;

Кокиева Галия Ергешевна,

канд. техн. наук, доцент кафедры,

Бурятский институт инфокоммуникаций (филиал)

ФГБОУ ВО «Сибирский государственный университет телекоммуникаций и информатики»,

г. Улан-Удэ, Республика Бурятия, Россия

АЛГОРИТМ СОЗДАНИЯ КОМПЬЮТЕРНОЙ ИГРЫ

В данной статье представлен алгоритм разработки 2D игры «Танчики». Рассмотрены этапы создания игры: описание идеи и жанра игры, выбор инструментария для создания игры; создание игровых объектов, оформление экрана и меню, выбор звуковых эффектов, программирование сюжета.

Ключевые слова: компьютерная игра, этапы создания игры, спрайт, фреймворк.

Venera V. Rabdanova,

candidate of economic sciences, head of the department;

Inga B. Eltunova,

candidate of pedagogical sciences, head of department

distance and additional vocational education,

Galiya E. Kokieva,

candidate of technical sciences, associate professor of the department,

Buryat Institute of Infocommunications (branch)

FGBU VO «Siberian State University of Telecommunications and Informatics»,

Ulan-Ude, Republic of Buryatia, Russia

THE ALGORITHM OF CREATING A COMPUTER GAME

In this article, an algorithm for developing the 2D game «Tanchiki» is presented. The stages of creating the game are considered: the description of the idea and genre of the game, the choice of

tools for creating the game; creation of game objects, design of the screen and the menu, a choice of sound effects, programming of a plot.

Keywords: computer game, the stages of creating a game, sprite, framework.

В наше динамичное время компьютерные игры заняли прочную нишу в сфере отдыха и развлечений. Игры создаются для разных целей – дать возможность отвлечься, получить удовольствие, а для кого-то это может быть обучение. Для обывателя создание игры не кажется сложным, но на самом деле – это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов, включающий в себя как технические, так и творческие моменты.

Разработка игры обычно ведется поэтапно:

- описание идеи игры, выбор жанра игры;
- выбор игрового движка для создания игры;
- создание игровых объектов;
- оформление экрана и меню;
- выбор звуковых эффектов;
- программирование сюжета;
- тестирование.

В данной статье приведен пример создания игры «Танчики». Цель данной игры – создание ремейка популярной в свое время игры «Battle City» или, как называют в простонародье, «Танчики». Под ремейком подразумевается копия какой-либо игры, но с некоторыми изменениями, портирование на другие операционные системы или изменения в самой игре, но с сохранением концепции оригинала. Жанр игры – аркадный.

Для разработки игры выбран кроссплатформенный свободно распространяемый фреймворк LÖWE (также известен как Love 2D), предназначенный для разработки компьютерных игр на скриптовом языке Lua, распространяемый по лицензии zlib и предусматривающий свободное использование, как в открытых, так и в коммерческих проектах с закрытым исходным кодом [2].

Для начала осуществления разработки нужно собрать и сделать контент, который будет использоваться в игре. В целях сохранения базовой эстетики игры спрайт танка был скачен из интернета (рис. 1).

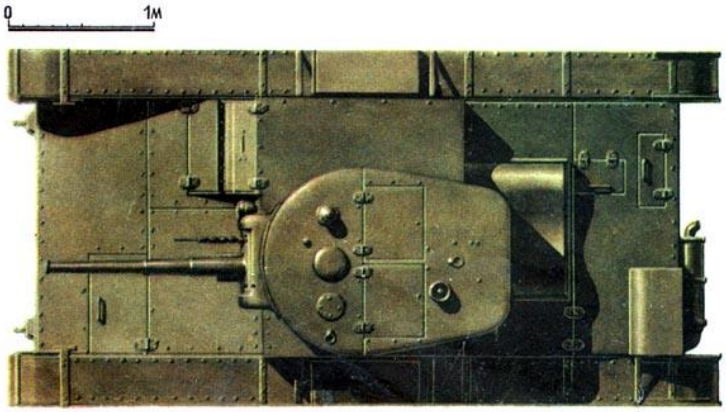

Рисунок 1 – Спрайт танка Т-26

Затем данный спрайт был отредактирован для двух игроков: для первого игрока был выбран красный цвет, для второго игрока – зелёный.

В любой игре есть меню, поэтому на следующем этапе были разработаны спрайты для меню. Был создан текст Battle City, залит красной заливкой с применением фильтра зерно. На выходе была выставлена прозрачность фонового слоя 0, рисунок был обрезан и получен следующий спрайт (рис. 2).

Рисунок 2 – Спрайт Battle City

Таким же образом поступаем с надписями Player 1 и Player 2.

Рисунок 3 – Спрайт Player 1

Рисунок 4 – Спрайт Player 2

На следующем этапе была создана текстура, из которых будет состоять база игроков. Было создано изображение 32x30 пикселей, залито красной заливкой и применен фильтр текстуризатор, выбрана текстура «кирпич», был выполнена подгонка масштаба, выбран рельеф и задан свет, вниз и влево.

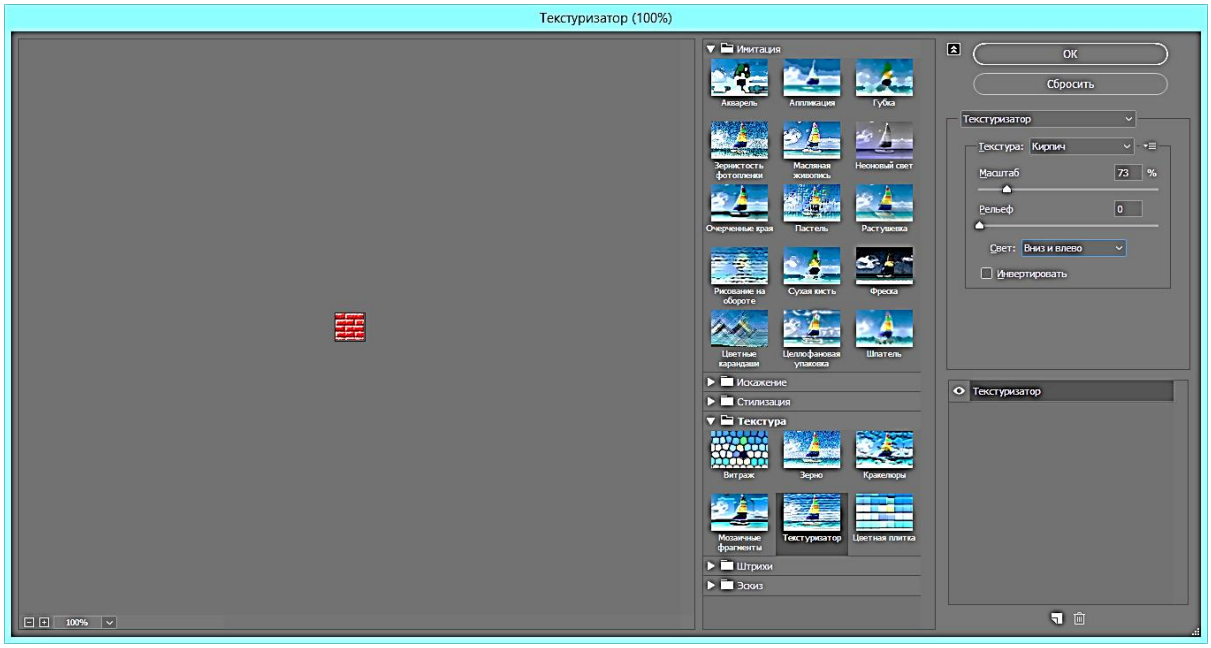

Рисунок 5 – Текстура кирпича

В качестве снаряда была скачана картинка из свободного доступа (рис. 8).

Рисунок 6 – Спрайт снаряда

Следующий этап – это программирование сюжета. Для этого была создана папка с фреймворком main.lua, где были созданы три главные функции фреймворка: love.load(), love.update(dt), love.draw(). В love.load() идёт инициализация и загрузка всех данных. Функция вызывается один раз при старте программы. В love.update(dt) идет постоянное обновление, в неё записывают всё, что должно происходить в режиме реального времени. В скобках (dt) указана единица времени delta time. В love.draw() происходит отрисовка объектов на экране. Также сразу был создан побочный файл conf.lua для определения наименования игры, ширины и высоты окна.

function love.conf(t)

```
t.window.title = «Tanks!»
t.window.width = 800
t.window.height = 600
end
```

В LÖWE понятия объектно-ориентированного программирования нет, поэтому все будет реализовываться с помощью метатаблиц. Была создана новая таблица с названием gamestate = {}. Она будет использоваться для переключения игровых состояний, меню или уровня. В главных функциях был написан следующий код:

```
function love.load()
if gamestate.current.load then gamestate.current.load() end
end
--обновляем
function love.update(dt)
if gamestate.current.update then gamestate.current.update(dt) end
end
--рисуем
function love.draw()
if gamestate.current.draw then gamestate.current.draw() end
end
--Проверяем нажатия клавиш
function love.keypressed(key)
if gamestate.current.keypressed then gamestate.current.keypressed(key) end
end
```

Затем было создано меню через создание таблицы menu = {}. Она будет хранить функции вызова следующего уровня, на двоих игроков или на одного: menu.load(), menu.update(dt), menu.draw(). В menu.load() записан следующий код:

```
function menu.load()
menu.current = 1
```

menu.ty = 230

menu.Label1 = love.graphics.newImage («BATTLE CITY.png»)

menu.Label2 = love.graphics.newImage («PLAYER 1.png»)

menu.Label3 = love.graphics.newImage («PLAYER 2.png»)

menu.T26 = love.graphics.newImage («T-26-1.png») – загрузили спрайт

menu.soundc = love.audio.newSource («choose.wav», «static»)

menu.music = love.audio.newSource («8bit.mp3») – если «static» не указано, LÖVE будет проигрывать файл с диска, подходит для длинных музыкальных треков

love.audio.play (menu.music)

end

Установление значения переменной menu.current = 1 означает, что по умолчанию режим будет на одного игрока; menu.ty = 230 – смещение спрайта танка по Y координате.

Важно сказать, что в Lua присутствует динамическая типизация переменных, тип переменной определяется автоматически, «на лету».

love.graphics.newImage – библиотека, загружающая картинки в проект.

menu.label1, menu.label2, menu.label3, menu.T26 – переменные принимают на себя загруженную картинку, чтобы в дальнейшем работать с ней.

Библиотека love.audio.newSource загружает звуки и музыку для проекта, принимает такие аудиоформаты как wav, ogg, mp3.

И любой формат трекерной музыки: XM, MIDI, MOD, всего более 12.

В функцию menu.keypressed(key) записан следующий фрагмент:

function menu.keypressed(key)

if key == 'up' or key == 'w'

then -- если нажата клавиша вверх, отрисовываем

love.audio.play(menu.soundc)

menu.current = 1

menu.ty = 230

love.graphics.draw (menu.T26,440,menu.ty,6.28,0.2,0.2,0,0)

```
end
if key == 'down' or key == 's'
then
love.audio.play(menu.soundc)
menu.ty = 330
menu.current = 2
love.graphics.draw(menu.T26,440,menu.ty,6.28,0.2,0.2,0,0)
end
-- На escape выходим
if key == 'escape'then love.event.quit() end
if key == 'return' then menu[menu.current].func() end
end
```

Библиотека love.keypressed (key), относится к love.keyboard. Эта функция вызывается, когда происходит нажатие клавиши. Далее описаны происходящие события.

Если нажата клавиша ‘w’ или ‘up’, стрелка вверх, прозвучит звук выбора, который был загружен (menu.soundc), установление значения переменной menu.current = 1, установление координаты Y для отрисовки спрайта танка и отрисован сам танк функцией love.graphics.draw (menu.T26, 440, menu.ty, 6.28,0.2,0.2,0,0).

Первый аргумент – загружаемый спрайт, второй аргумент – координата x, следующая координата y, четвертый угол поворота спрайта (следует сказать, что в LÖWE угол поворота измеряется не в градусах, а в радианах), пятый и шестой аргументы, масштаб спрайта, последние два аргумента смещение начала координат по x, y.

Если нажата клавиша ‘s’ или ‘down’, стрелка вниз, то события развиваются аналогичным образом, были изменены координаты спрайта menu.T26 и установлено значение menu.current = 2, то есть на двух игроков. Нажатие клавиши Esc вызывает функцию выхода love.event.quit() и происходит выход из игры.

Таким образом, в данной статье показан алгоритм создания небольшой игры «Танчики».

СПИСОК ЛИТЕРАТУРЫ

1. Battle City. – URL: https://ru.wikipedia.org/wiki/Battle_City (дата обращения: 25.04.2018).

2. Unity (игровой движок). – URL: https://ru.wikipedia.org/wiki/Unity (дата обращения: 25.04.2018).

3. Семь этапов создания игры: от концепта до релиза. – URL: https://habr.com/company/miip/blog/308286/ (дата обращения: 25.04.2018).

4. Шабалина О.А. Разработка обучающих компьютерных игр: как сохранить баланс между обучающей и игровой компонентой? // ОТО. – 2013. – №3. – URL: https://cyberleninka.ru/article/n/razrabotka-obuchayuschih-kompyuternyh-igr-kak-sohranit-balans-mezhdu-obuchayuschey-i-igrovoy-komponentoy (дата обращения: 25.04.2018).