Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

(ВолгГТУ)

Кафедра «Системы автоматизированного проектирования и поискового конструирования»

Отчет по лабораторной работе №1

по дисциплине

«Проектирование АСОиУ»

|  |
| --- |
| Выполнил: студент группы САПР-1н  Резуненко А.И. |
| Проверил(а): |
|  |

Волгоград, 2018

На рисунках 1-7 изображён результат построения BPMN-диаграмм. Самая верхняя диаграмма является общей, все последующие – поясняющими.

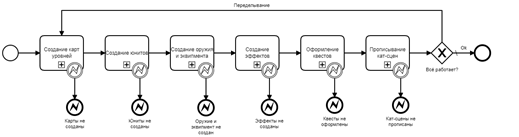


Рисунок 1. Диаграмма, поясняющая процесс портирования игры Код Доступа Рай на движок игры Власть Закона: Полицейские Истории.

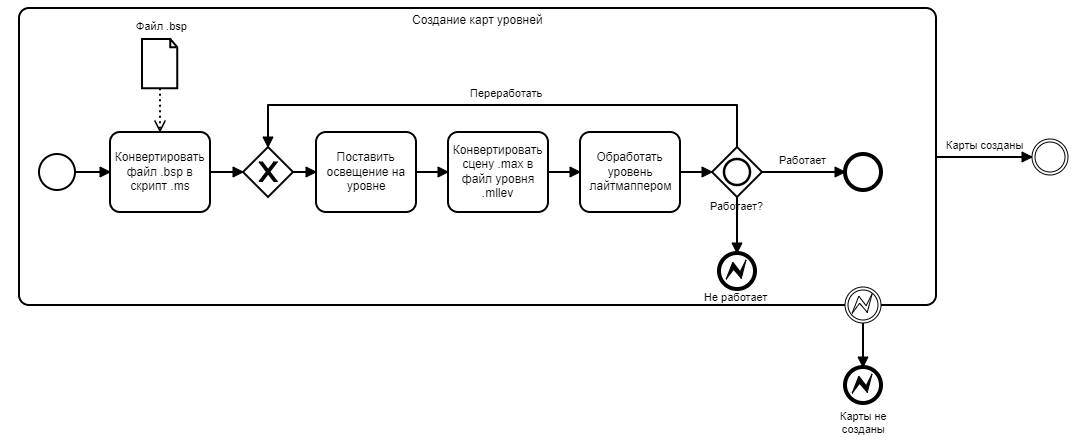


Рисунок 2. Диаграмма, описывающая процесс создания карт уровней для игры Власть Закона: Полицейские Истории.

Файл формата .bsp берётся из ресурсов игры Код Доступа: Рай. Освещение выставляется исключительно типа light spot. Конвертация происходит с помощью внутренних компонентов 3D Studio Max 8, а именно при помощи плагина max2mllev. Обработка уровня лайтмаппером необходима для того, чтобы прописать каждой поверхности уровня параметр и уровень освещённости. Процесс зависит от количества источников света и их расположения, занимает от 30 минут до 10 часов.

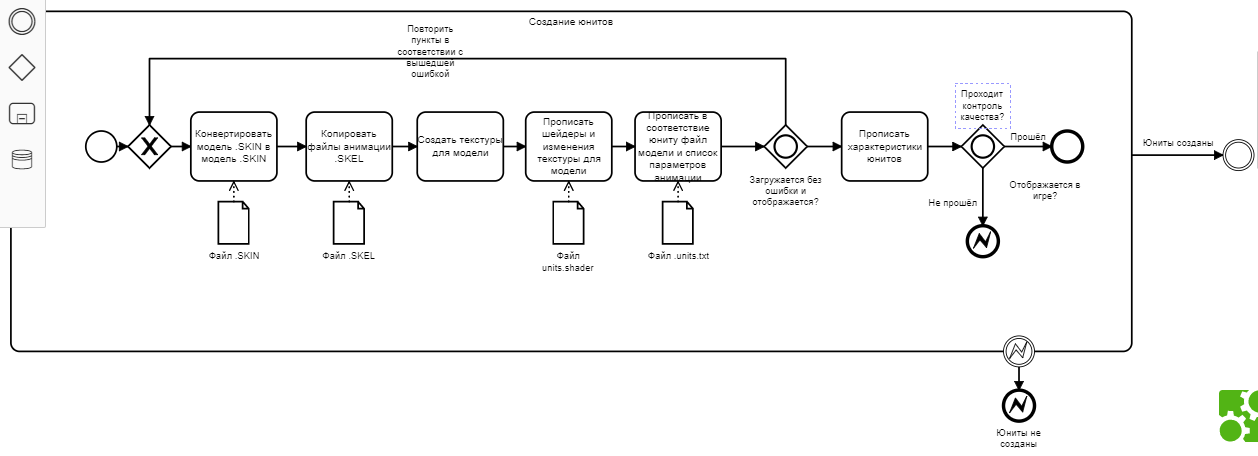


Рисунок 3. Диаграмма, описывающая процесс создания юнитов для игры Власть Закона: Полицейские Истории.

Все файлы форматов .SKIN, .SKEL берутся из ресурсов игры Код Доступа: Рай. Конвертация происходит с помощью программы Skin Converter 3.21. Текстуры для модели создаются при помощи любого графического редактора. На выходе должен быть файл квадратной (х\*х пиксел) размерности формата .dds или .tga, который должен совпать с UVW развёрткой модели в 3D Studio Max. Файл формата unit.shader содержит такую структуру привязки модели к текстуре:

dockglassShad / название шейдера

{ /блок шейдера

cull front / найстройка шейдера

deformVertexes autosprite

surfaceParam flammable

surfaceParam nolightmap

surfaceParam playerclip

polygonOffset

armour 5

marks hot

steps water

{ /блок первой текстуры

map textures/Power\_of\_Law/f\_x/blue\_refl.tga / путь к текстуре, команда map

tcGen enviroment /настройка текстуры

blendFunc SRC\_ALPHA INV\_SRC\_ALPHA

tcMod emboss 1 3 8

tcMod scroll 1 3

rgbGen wave sin 1 2 3 4

FillMode solid

depthWrite

depthFunc equal

zBias 98

nomipmap

nopicmip

}

{ /блок настройки шейдера

map $lightmap

blendFunc DST\_COLOR SRC\_COLOR

}

}

Этот файл берётся из игры Власть Закона: Полицейские Истории, и в него записываются все необходимые пункты, учитывая, какое количество каких текстур требуется наложить на модель. Также следует учитывать, что модели КДР содержат в себе 1 шейдер, или ~10. Модели ВЗ:ПИ содержат в себе только 2 шейдера – \_body и \_head.

Файл units.txt берётся также из игры Власть Закона: Полицейские Истории, и в неё прописывается системное наименование юнита, его имя, отображаемое в игре, характеристики, файл используемой модели для разных типов носимой брони (suit1..4+scuba). Контроль качества осуществляется самим же работником.

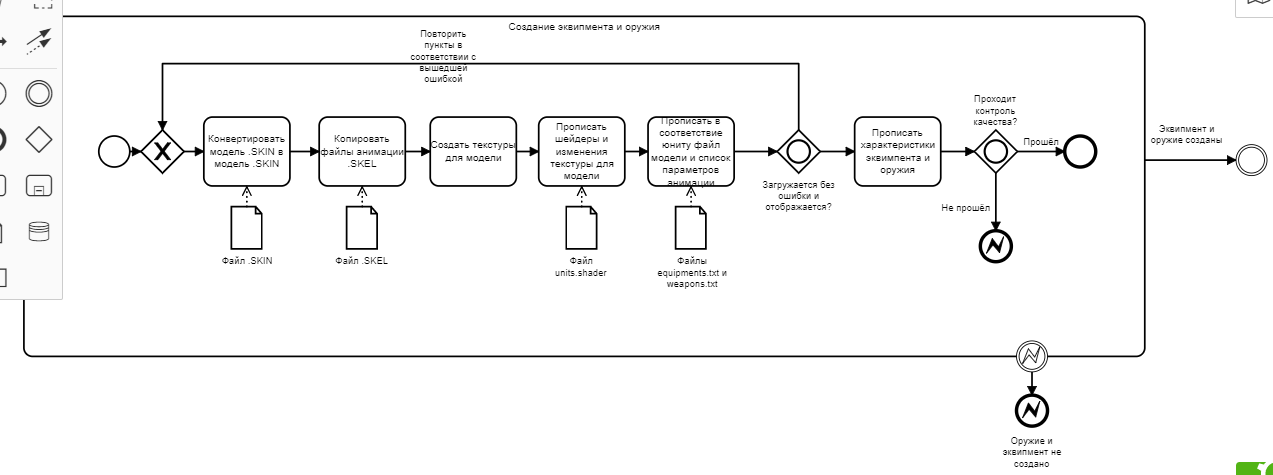


Рисунок 4. Диаграмма, описывающая процесс создания оружия и эквипмента для игры Власть Закона: Полицейские Истории.

Этот процесс практически полностью повторяет предыдущий шаг, за исключением разницы в тонкости работы. Используются другие файлы анимации, другие текстуры и другие характеристики. Также всё техническое «собирается» в других файлах – weapons.txt и equipments.txt.

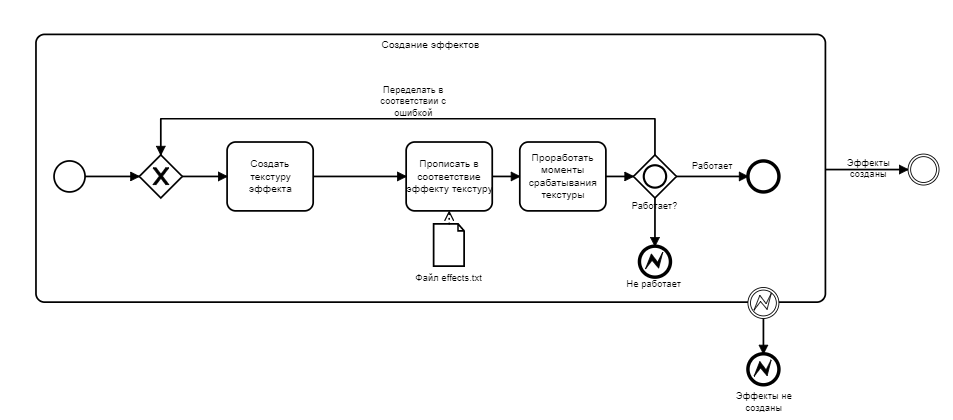


Рисунок 5. Диаграмма, описывающая процесс создания эффектов для игры Власть Закона: Полицейские Истории.

Моменты срабатывания текстуры прописываются в файлах, к которым относится сработавший эффект. Вспышка выстрела – в weapons.txt, полёт снаряда – в ammo.txt, использование обмундирования типа аптечка/силовой щит/имплантат – в equipment.txt.

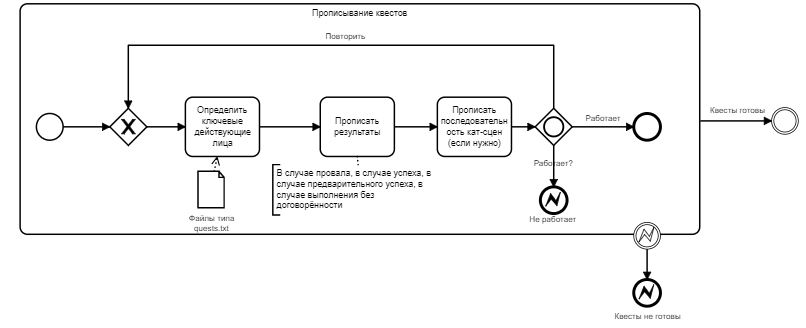


Рисунок 6. Диаграмма, описывающая процесс создания квестов для игры Власть Закона: Полицейские Истории.

Квесты создаются по определённой структуре. Данная структура представляет собой таблицу в формате .txt, в которой указаны:

Quest Name - Название квеста, используется в журнале.

System Name – Системное название квеста, используется для условий срабатывания квеста.

Autostart – Возможность запуска квеста при начале игры.

AutoFinish – Возможность автомотического завершения квеста при его взятии.

Quest Holder – Кто даёт квест.

Description – Описание квеста.

Phase – Фаза, при которой этот квест можно получить.

Quest Tasks – Условие запуска квеста.

Dependencies – Условие, при котором этот квест возможно запустить (прописывается порядок выполнения квестов).

Start Bonus – Бонус при взятии квеста.

Complete Bonus – Бонус при завершении квеста.

Failed Bonus – Бонус при провале квеста.

Passed Bonus – Бонус при завершении квеста ДО его взятия при попытке взять квест.

Passed OK Bonus – Бонус при повторном выполнении условия запуска квеста.

Start message – Сообщение при запуске квеста.

Complete message – Сообщение при выполнении квеста.

Inprogress message – Сообщение в процессе выполнения квеста.

Passed message – Сообщение при выполнеии квеста ДО взятия и при выполнении условий взятия квеста.

Failed message – Сообщение при провале квеста.

Passed OK message – Сообщение при повторном выполнении условия запуска квеста.

Journal message – Информация, сохранённая в Журнале.

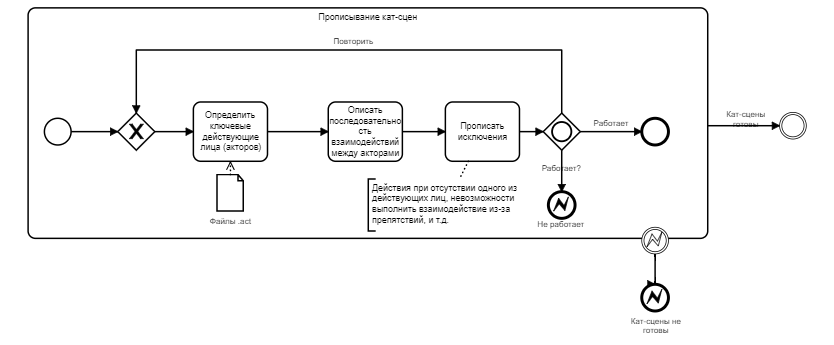


Рисунок 7. Диаграмма, описывающая процесс создания кат-сцен для игры Власть Закона: Полицейские Истории.

Файлы типа .act создаётся разработчиком по такой структуре:

quest mission\_gm = //Название квеста

holder : mission\_officer; //Кто даёт квест

depends\_on : hero(alex); //Кто участвует в квесте

task : bring(giga\_order,mission\_officer); //Задача

end\_task : talk(mission\_officer); //Условие для окончательного завершения квеста

remind : on\_talk(mission\_officer); //Напоминание о квесте

on pre\_success : say(mission\_officer, police\_access); //Результат при предварительном завершении (выполнение задачи ДО взятия квеста)

on success : reset(nturns) & level(west\_giga); //Результат при завершении

.

Последовательность кат-сцен описывается в strigger и sscene, имеющих такую структуру:

strigger pd\_alex = context(pd\_5floor,1): alan\_alex. //[Название триггера] = [Условие срабатывания триггера]: [Название кат-сцены]

alan\_alex = sscene(

(camfollow(player) | walkz(player, alan\_arrest\_a)) //Фокусировка камеры с выполнением действия без слежения за действующим

&camfocus(alan)&wait(1) //Фокусировка камеры с одновременным ожиданием

&say(alan, alan\_magvaer\_1) //Открытие информационного окна-диалога с прописанным в local2.txt высказыванием. Окно закрывается при нажатии на соответствующую кнопку.

&give(giga\_order) //Получение какого-либо предмета.

&addrec(mission\_gm,mission\_gm\_header,mission\_gm\_entry) //Добавление и системная запись

);