09.02.07, ИР-330

РЕВЬЮИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

«Модификация игрового или мультимедиа-приложения»

КУРСОВОЙ ПРОЕКТ

**по профессиональному модулю**

**ПМ.03 «Ревьюирование программных продуктов»**

|  |  |  |
| --- | --- | --- |
| Студент |  | Топко Н.В  Каргаполов А.Е  Сивоха И.М |
| Руководитель |  | Бабикова Т.М. |

**Оглавление**

[**Введение 2**](#_Toc131492806)

[**Цель 2**](#_Toc131492807)

[**1 Аналитическая часть: 2**](#_Toc131492808)

[**1.1 Функции программного продукта 2**](#_Toc131492809)

[**Описание предметной области и задачи ПП 2**](#_Toc131492810)

[**1.2 Анализ структуры данных существующего программного продукта и обоснование ее модификации: 2**](#_Toc131492811)

[**Классы 3**](#_Toc131492812)

[**ER-диаграмма с добавлением изменений 5**](#_Toc131492813)

[**1.3 Проектирование интерфейса клиентской части программного продукта 5**](#_Toc131492814)

[**Экономическая часть 10**](#_Toc131492815)

[**2.1 Описание программного продукта 10**](#_Toc131492816)

[**2.2 Описание качества программного продукта 10**](#_Toc131492817)

[**2.3 Расчет технико-экономической эффективности разработки ПС. Расчет коэффициента технического уровня 11**](#_Toc131492818)

[**2.4 Расчет затрат на разработку продукта 14**](#_Toc131492819)

[**Заключение 18**](#_Toc131492820)

[**Использованные источники информации 18**](#_Toc131492821)

**Содержание**

# **Введение**

Основанием для разработки проекта является учебный план специальности 09.02.07 Информационные системы и программирование (квалификация выпускников - Специалист по информационным ресурсам).

## Цель

Создать мультимедийную игру, по мотивам исторических событий из реальной жизни и погрузить пользователя в атмосферу того времени. Получить качественный и интересный программный продукт, который будет аналогом исторической битвы.

# **1 Аналитическая часть:**

## 1.1 Функции программного продукта

- хранение информации о очках и отображение игровых очков.

- пользователь должен иметь возможность управлять оружием героя.

- выбор уровня сложности игры.

- отображение состояния игрока.

- генерация непрерывного цикла игрового контента.

- формирования критерия окончания игры(проигрыш).

формирования критерия окончания игры(выйгрыш).

- выигрыш наступает в случае достижения определенного количества очков.

## Описание предметной области и задачи ПП

Разработка игрового приложения по мотивам исторических событий, а именно сражения, именуемого «Куликово поле».

В приложении присутствует описание предыстории битвы, подчеркнутое из различных источников, освещающих исторические события. Отличительной чертой являются графические элементы, выполненные собственноручно разработчиком игры. В игре присутствует возможность передвижения по локации и взаимодействие с объектами.

* 1. **Анализ структуры данных существующего программного продукта и обоснование ее модификации:**

## Классы

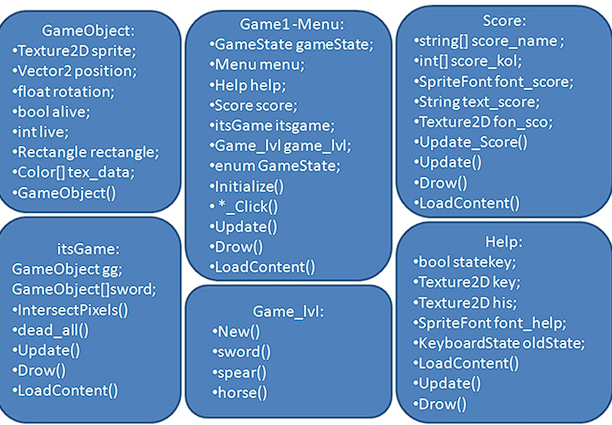
****

Рис 10. Классы

Все обьекты игры между которыми происходит взаимодействие отражает класс GameObject.

**Класс GameObject включает свойства**:

Sprite – текстура обьекта;

Position – позиция обьекта;

Rotation – угол поворота обьекта;

Alive – состояние активности;

Live – состояние количества жизней;

Rectangle – координаты текстуры;

Tex\_data – цвет невидемой части.

Логика взаимодействия между обьектами и игровая анимация описанна в классе itsGame.

**Класс itsGame включает свойства**:

Font – шрифт текста игровых очков и времени;

GG – обьект главного героя;

Sword[] – массив обьектов «Ратник»;

Spear[] – массив обьектов «Лучник»;

Horse[] – массив обьектов «Всадник»;

Trav – текстура игрового поля;

Corsor – обьект курсора

Точка входа в приложение и обьявление экземпляров ключевых классов находится в Game1-Menu.

**Класс Game1-Menu включает свойства:**

GameState – игровое состояние;

Menu – экземпляр класса Menu;

Help - экземпляр класса Help;

Score - экземпляр класса Score;

Отображение, подсчет и сортировку очков выполняет класс Score.

**Класс Score включает свойства:**

Score\_name – имена порядковых номеров таблицы очков;

Score\_kol – количество очков;

За раздел меню «Справка» и его подразделы «Управление» и «История» отвечает класс Help.

**Класс Help включает свойства:**

Statekey – положение выбранной позиции меню

Key – изображение справки о клавиатуре

## ER-диаграмма с добавлением изменений

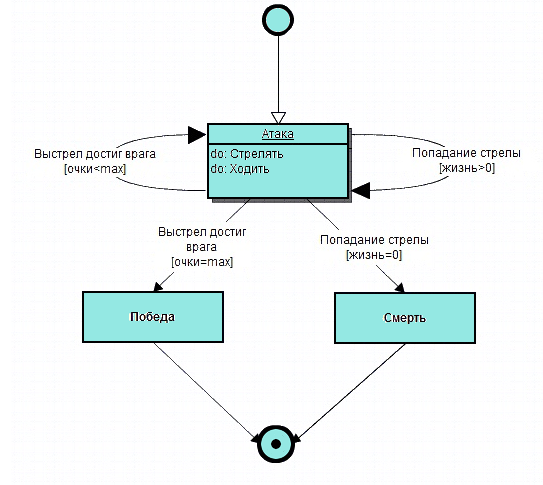
****

Рис 11. Блок схема диаграммы

* 1. **Проектирование интерфейса клиентской части программного продукта**

Загрузочный экран

****

Рис 1. Загрузочный экран

Настройки игры



Рис 2. Настройки игры

Заставка перед началом игры



Рис 3. Заставка перед началом игры

Вторая заставка перед началом игры



Рис 4. Вторая заставка перед началом игры

История игры

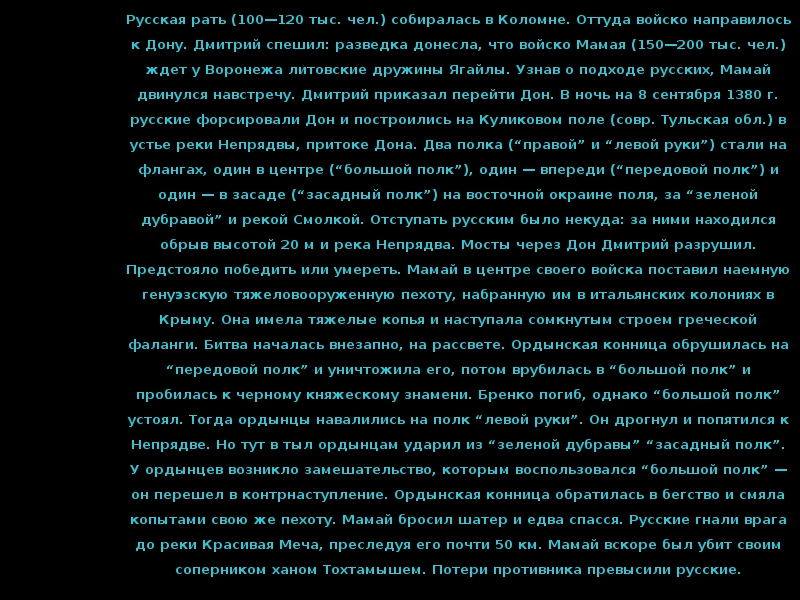


Рис 5. История игры

Все виды персонажа: с луком; с копьем; с мечом; на лошади с луком

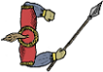


Рис 6. Все виды персонажа: с луком; с копьем; с мечом; на лошади с луком

Курсор



Рис 7. Курсор

Игровая карта. Поле боя

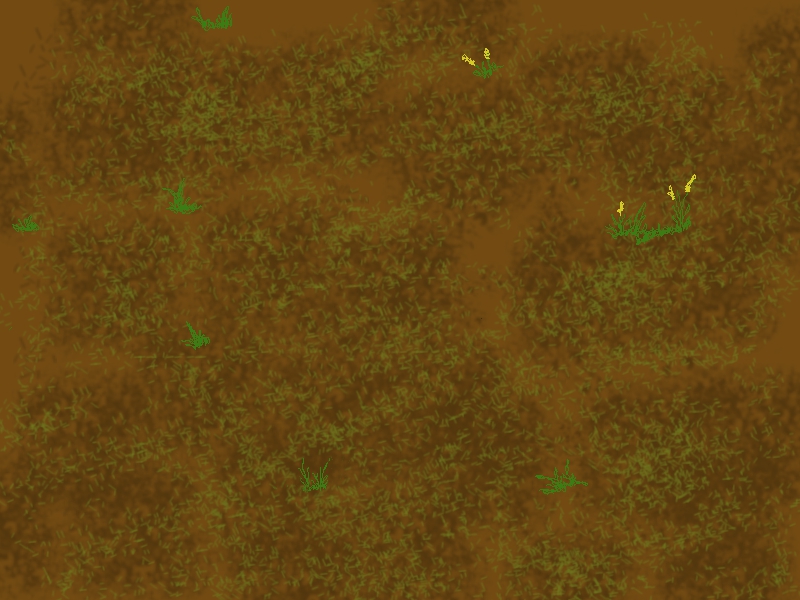


Рис 8. Игровая карта. Поле боя

Окончание игры при смерти(поражении)



Рис 9. Окончание игры при смерти(поражении)

# **Экономическая часть**

## 2.1 Описание программного продукта

- игра “Куликово поле” 28.03.2012 года выпуска, версия приложения Visual Studio 2009;

- возможность запуска приложения, управление персонажем в игре, взаимодействие с объектами;

- для запуска приложения необходима программа Visual Studio 2009 года выпуска или ниже;

- графический интерфейс приложения для взаимодействия с элементами;

- инсталляция продукта пользователем производится не должна;

- документация не поставляется;

- поддержка для продукта не предлагается;

- возможность провести диагностику, изменить код программы и выполнить проверочное тестирование программы;

## 2.2 Описание качества программного продукта

* **Функциональные возможности**:
* Функциональная полнота и согласованность (приводится набор функций, реализуемых в данной версии и соответствие этих функций стандартам, соглашениям, методическим рекомендациям, степень достаточности и правильности функций в соответствии в назначением продукта);
* граничные значения (минимальные или максимальные значения; длины ключей; максимальное число записей в файле; максимальное число критериев поиска; минимальный объем выборки и т.д.);
* только пользователь, у которого есть приложение, имеет доступ к защищаемой информации и технологии защиты данных и неправомерного постороннего доступа к приложению.
* **Надежность -** приводится описание набора программных и организационных средств, обеспечивающих возможность сохранять качество функционирования программного продукта при установленных условиях за установленный период времени:
* средства проверки достоверности исходных данных;
* средства зашиты от ошибок пользователя;
* средства восстановления работоспособности программной системы при программных ошибках;
* средства восстановления работоспособности программной системы при сбое аппаратуры.
* **Практичность –** приводится описание свойств, способствующих быстрому освоению, применению и эксплуатации программного продукта с минимальными трудозатратами с учетом характера решаемых задач и требований к квалификации обслуживающего персонала:
* описание интерфейса пользователя (тип интерфейса: командная строка; меню; окна, функциональные клавиши; функция подсказки, справочная система, руководство пользователя; притягательность и удобство интерфейса);
* требуемая квалификация пользователя (набор знаний необходимых пользователю для эксплуатации программного продукта: знание соответствующей технической области; знание операционной системы; знания, получаемые в результате специального обучения и т.д.);
* возможность настройки или адаптации продукта к требованиям пользователя (если продукт может настраиваться пользователем, то должны быть указаны инструментальные средства для проведения такой настройки и условия их применения: изменение параметров, алгоритмов вычислений; назначение функциональных клавиш и т.д.);
* наличие защиты от нарушения авторских прав (техническая защита от копирования; запрограммированные даты окончания использования продукта; интерактивные напоминания об оплате за копии и пр.);
* **Эффективность** – приводится информация о характере поведения продукта, его степени удовлетворения потребности пользователя в обработке данных с учетом экономических, вычислительных и людских ресурсов:
* время отклика системы на запрос пользователя
* время выполнения заданных функций в установленных условиях - для заданных конфигураций системы и профилей загрузки и др.
* оценка производительности при реализации заданных функций в установленных условиях - для заданных конфигураций системы и профилей загрузки и др.
* **Сопровождаемость** - характеризует технологические аспекты, обеспечивающие простоту устранения ошибок в программе и программных документах и поддержания в актуальном состоянии:
* возможность диагностики в случае отказов;
* определение условий для модификации, либо изменения режимов эксплуатации;
* возможность тестирования модифицированных частей программного продукта.
* **Мобильность** - означает способность продукта быть системой, переносимой из одной аппаратно-программной среды в другую:
* возможность адаптации продукта к различным условиям эксплуатации без применения дополнительных сервисов;
* простота внедрения в новых условиях;
* взаимозаменяемость с другим (аналогичным) программным продуктом.
  1. **Расчет технико-экономической эффективности разработки ПС. Расчет коэффициента технического уровня**

Расчет технико-экономической эффективности разработки ПП выполняется в тех случаях, когда на предприятии уже эксплуатируется система, выполняющая аналогичные разрабатываемой ПП функции или в случаях, когда на рынке ПП имеются подобные системы, а также в случае доработки, усовершенствования программного продукта.

На первом этапе необходимо выбрать оцениваемые показатели качества продукта. Их описания желательно предварительно упорядочить по приоритетам с учетом назначения и сферы применения конкретного проекта программного средства. Далее необходимо ранжировать выбранные показатели качества по приоритетам потребителей, которым необходимы определенные характеристики программного проекта с учетом их специализации и профессиональных интересов. Ранг (рейтинг) показателя определяется весовым коэффициентом от 0 до 1. Сумма весовых коэффициентов, относящихся к одному и тому же показателю, принимают постоянной величиной, равной единице: ∑ Vi=1.

Второй этап выбор шкал для описания характеристик качества программных средств. Далее для каждого из отобранных показателей должна быть установлена и согласована шкала оценок. Для показателей, представляемых качественными признаками, желательно определить и зафиксировать в спецификациях описания условий, при которых следует считать, что данная характеристика реализуется в программном средстве. Выбранные значения характеристик качества и их атрибутов должны быть предварительно проверены разработчиками на их реализуемость с учетом доступных ресурсов конкретного проекта и при необходимости откорректированы. Оценка показателя должна быть дана в соответствии со следующими категориями ответов, касающихся проявления конкретных показателей:

* Хi = 1, проявление показателя случайное;
* Хi = 2, проявление небольшое, эпизодическое;
* Хi = 3, проявление среднее;
* Хi = 4, проявление важное, значительное, устойчивое;
* Хi = 5, проявление основное, существенное, весьма устойчивое.

В случае количественной оценки показателя вычисленное значение переводится в систему баллов от 1 до 5:

* 0 – 20% –1 балл;
* 21 – 35% – 2 балла;
* 36 – 55% – 3 балла;
* 56 – 80% – 4 балла;
* 81 – 100% – 5 баллов.

Третий этап измерение и расчет показателей качества для программного продукта.

Показатель качества оценивают суммой произведений взвешенных оценок относительных показателей критериев, относящихся к оцениваемому фактору (рисунок 12).

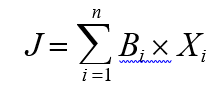


Рис 12. Показатель качества

где N – число рассматриваемых показателей.

Четвертый этап расчет коэффициента технического уровня отношением итогового качества программного продукта и его аналога (или предыдущей версии) (рисунок 13).

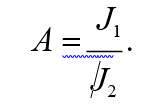


Рис 13. Формула расчета коэффициента технического уровня

Разработка проекта с технической точки зрения оправдана, если полученный коэффициент больше единицы.

В качестве программы-аналога при разработке проекта принята подобная система на рынке ПП «Цеховая база».

Показатели качества разрабатываемой ПС и аналога, оцененные и приведены в таблице 1.

Таблица 1. Показатели качества.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показатели качества | Весовой коэффициент  Bi | Проект | | Аналог | |
| X1 | B x X1 | X2 | B x X2 |
| 1. Удобство работы (пользовательский интерфейс) | 0,1 | 5 | 0,5 | 3 | 0,3 |
| 2. Стабильность | 0,06 | 4 | 0,24 | 4 | 0,24 |
| 3. Соответствие профилю деятельности  заказчика | 0,15 | 5 | 0,75 | 2 | 0,3 |
| 4. Функциональность | 0,05 | 4 | 0,2 | 3 | 0,15 |
| 5. Надежность (защита данных) | 0,13 | 3 | 0,39 | 4 | 0,52 |
| 6. Скорость доступа к данным | 0,09 | 5 | 0,45 | 5 | 0,45 |
| 7. Гибкость | 0,05 | 5 | 0,25 | 4 | 0,2 |
| 8. Функции обработки информации | 0,13 | 3 | 0,39 | 2 | 0,26 |
| 9. Возможности | 0,09 | 5 | 0,45 | 3 | 0,27 |
| 10 Время обучения персонала | 0,15 | 5 | 0,75 | 3 | 0,45 |
| Обобщенный показатель качества J | | J1=4,37 | | J2=3,14 | |

Коэффициент технического уровня составляет (рисунок 14).

**A = J1/J2 = 4,37 / 3,14 = 1,39**

Рис 14. Вычисление коэффициента технического уровня

Так как коэффициент больше 1, то разработка проекта с технической точки зрения оправдана.

* 1. **Расчет затрат на разработку продукта**

Оценка данного курсового проекта произведена методом PERT.

Инженерный метод оценки трудоемкости проекта PERT (Program/Project Evaluation and Review Technique) был разработан в 1958 году в ходе проекта по созданию баллистических ракет морского базирования «Поларис». Входом для данного метода оценки служит список элементарных пакетов работ.

Перечень элементарных пакетов работ данного приложения для проведения оценки:

* создание базы данных (количество 0 шт.);
* разработка функций (количество 7 шт.);
* реализация защиты данных (количество 1 шт.);
* разработка элементов интерфейса (количество 5 шт.);
* разработка кнопок (количество 4 шт.).

Диапазон неопределенности достаточно охарактеризовать тремя оценками:

* Mi – Наиболее вероятная оценка трудозатрат;
* Oi - Минимально возможные трудозатраты на реализацию пакета работ;
* Pi – Максимально возможные трудозатраты.

Проведем оценку для лучших, худших и наиболее вероятных случаев.

Оценку средней трудоемкости по каждому элементарному пакету можно определить по формуле (рисунок 15).



Рис 15. Расчет оценки средней трудоемкости

где Pi – Максимально возможные трудозатраты;

Mi – Наиболее вероятная оценка трудозатрат;

Oi – Минимально возможные трудозатраты на реализацию пакета работ.

1) Создание базы данных.

Oi = 0

Pi = 1

Mi = 0

Средняя трудоемкость рассчитана по формуле:

Еi = (Pi+4\*Mi + Oi)/6 = (5+4\*0+1) /6= 1 чел. час

2) Разработка хранимых процедур и представлений.

Oi = 5

Pi = 20

Mi = 15

Средняя трудоемкость рассчитана по формуле:

Еi = (Pi+4\*Mi + Oi)/6 = (20+4\*15+5) /6= 14 чел. час

3) Реализация защиты данных.

Oi = 3

Pi = 13

Mi = 7

Средняя трудоемкость рассчитана по формуле:

Еi = (Pi+4\*Mi + Oi)/6 = (13+4\*7+3)/6= 7,3 чел. час

4) Разработка элементов интерфейса.

Oi = 10

Pi = 30

Mi = 17

Средняя трудоемкость рассчитана по формуле:

Еi = (Pi+4\*Mi + Oi)/6 = (30+4\*17+10)/6= 18 чел. час

5) Разработка кнопок.

Oi = 3

Pi = 15

Mi = 10

Средняя трудоемкость рассчитана по формуле:

Еi = (Pi+4\*Mi + Oi)/6 = (15+4\*10+3)/6= 9,6 чел. Час

В таблице 2 показаны оценки наиболее вероятного и ожидаемого случаев.

Таблица 2. Оценка наиболее вероятного и ожидаемого случаев

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Компоненты | Лучший случай | Наиболее вероятный | Худший случай | Ожидаемый случай |
| Создание базы данных | 1 | 1 | 0 | 1 |
| Разработка хранимых процедур и представлений | 25 | 15 | 5 | 14 |
| Реализация защиты данных | 13 | 7 | 3 | 7,3 |
| Разработка элементов интерфейса | 30 | 17 | 10 | 18 |
| Разработка кнопок | 15 | 10 | 3 | 9,6 |
| Итого: | 84 | 50 | 21 | 49,9 |

Для расчета среднеквадратичного отклонения используется формула (рисунок 16).



Рис 16. Формула среднеквадратичного отклонения

Где Pi – Максимально возможные трудозатраты

Oi - Минимально возможные трудозатраты на реализацию пакета работ.

Для создания базы данных, среднеквадратичное отклонение равно:

CKOi = (Pi - Oi)/6 = (1-0)/6= 0 чел. Час

Для разработки хранимых процедур и представлений, среднеквадратичное отклонение равно:

CKOi = (Pi - Oi)/6 = (25-5)/6= 3,3 чел. час

Для реализация защиты данных, среднеквадратичное отклонение равно:

CKOi = (Pi - Oi)/6 = (13-3)/6= 1,6 чел. час

Для разработки элементов интерфейса, среднеквадратичное отклонение равно:

CKOi = (Pi - Oi)/6 = (30-10)/6= 3,3 чел. час

Для разработки кнопок, среднеквадратичное отклонение равно:

CKOi = (Pi - Oi)/6 = (15-3)/6= 2 чел. Час

Суммарная трудоемкость проекта может быть рассчитана по формуле (рисунок 17).



Рис 17. Формула суммарной трудоемкости проекта

Е = Σ Ei = 1\*1+8\*14+1\*7,3+5\*18+4\*9,6=248,7

Среднеквадратичное отклонение для оценки суммарной трудоемкости вычисляется по формуле (рисунок 18).



Рис 18. Формула среднеквадратичного отклонения

где CKO – среднеквадратичное отклонение

Среднеквадратичное отклонение для оценки суммарной трудоемкости равно:

CKO = √ (∑CKOi)^2 = 12,7 чел. /час

Оценка суммарной трудоемкости проекта, которая с вероятностью 95% точности расчета, можно применить формулу (рисунок 19).



Рис 19. Формула оценки суммарной трудоемкости

где CKO = Среднеквадратичное отклонение, а E = Ei

∑(95%=E+2\*CKO) = (248,7+2\*12,7) = 274,1 чел. Чаc

Полученную оценку трудоемкости кодирования необходимо умножить на четыре, поскольку помним, что кодирование составляет только 25% общих трудозатрат проекта. Поэтому суммарная трудоемкость нашего проекта составит, приблизительно = 1096,4 чел.\*час.

В месяц сотрудник будет работать по проекту, примерно 219,28 чел.\*час/мес.

Следовательно, трудоемкость проекта в человеко-месяцах составит, приблизительно 986,4 /176 ~ 6,3

Тогда согласно формуле Б.Боэма оптимальная продолжительность проекта составит:

T = 2,5 \* (6,3) ^ 1/3=5 месяцев.

Средняя численность команды – 6,3/5 ~ 1,2 человека.

# **Заключение**

В ходе работы мы создалимультимедийную игру, по мотивам исторических событий из реальной жизни. Получили качественный и интересный программный продукт, который будет аналогом исторической битвы.

Научились создавать приложение и код к нему, тестировать его , находить ошибки и исправлять их.

Сами нарисовали загрузочный экран, декорации, персонажей, оружие и заставки перед началом игры.

# **Использованные источники информации**

1. Приложение “Куликово поле”
2. Google.ru
3. Документ описания предметной области
4. Yandex.ru