Главное управление образования Гродненского облисполкома

Учреждение образования

«Гродненский государственный политехнический колледж»

Специальность: 2 – 40 01 01 «Программное обеспечение информационных

технологий»

Специализация: 2 – 40 01 01 35 «Программное обеспечение обработки

экономической и деловой информации»

Дисциплина: «Технология разработки программного обеспечения»

Группа: ПЗТ – 34

**ТЕХНО-РАБОЧИЙ ПРОЕКТ**

Техническое задание на разработку игры «Политех»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Разработала П.С. Кузьменок

Руководитель проекта Е.В. Заяц

2022

**Содержание**

Изм.

Кол

Лист №док

Подпись

Дата

Лист

2

КП 2-40 01 01.35.34.08.22 ПЗ

Разраб.

Кузьменок

Пров.

Заяц

За

Н. контр.

Утв.

Техническое задание на разработку игры «Политех»

Стадия

Листов

УО ГГПК

1. Постановка задачи 3
2. Техническое задание 6
3. Руководство пользователя 10
4. Приложение А 10

**1 Постановка задачи**  
 **1.1 Организационно-экономическая сущность задачи**

**Наименование задачи:** Компьютерная игра "Политех"

**Цель разработки:** Привлечение новых учащихся в политехнический колледж; мотивирование учащихся более творчески подходить к учебному процессу посредством отображения их будней в креативной форме; получение удовольствия и поднятие настроения; расширить имеющиеся знания в области языков программирования, познакомиться с новыми языками программирования и средами разработки; получить новый навык создания компьютерных игр;

**Назначение:** данный программный продукт разрабатывается для  
людей любого возраста, желающим погрузиться в повседневность среднестатистических учащихся колледжей, высших учебных заведений и т. п.

**Периодичность использования:** при желании ознакомиться с содержанием компьютерной игры, по мере скачивания.

**Источники и способы получения данных:** сеть интернет.

**1.2 Функциональные требования**

**Описание перечня функций и задач, которые должен выполнять будущий ПП:**  
Игрок:  
1. Просмотр глав игры  
2. Создание персонажа : выбор имени, пола персонажа, внешности персонажа, одежды  персонажа  
3.Прохождение мини-игр  
4. Взаимодействие с неуправляемыми персонажами посредством диалогов и выбора ответов, влияющих на исход игры  
5. Сохранение игрового прогресса  
6. Просмотр досье каждого из неуправляемых персонажей  
7. Просмотр небольшой части содержания следующих глав игры  
8. Возможность вернуться в одну из прошлых глав при неудовлетворении текущим исходом игры  
9. Редактирование персонажа при необходимости: изменение одежды, внешности.

**Сценарий**

Главный персонаж, вчерашний школьник, приезжает на остановку «Белкард», которая расположена возле Политехнического колледжа, дабы подать документы на профессию техник-программист.

Зайдя в колледж, главный персонаж сталкивается с вахтёршей-охранницей, которая спрашивает у него, где студенческий билет. При его отсутствии она не может пропустить игрока. Далее следует выбор «правильной реплики». В случае успеха охранница пропускает героя, а в случае поражения за него заступается новый NPS — Юрий Иванович Манкевич.

При любом результате ситуации, главному персонажу указывают дорогу к кабинету, где заседает приёмная комиссия. Чтобы скрасить время ожидания в коридоре, главный герой разговаривает с людьми, поступающими на эту же специальность, и узнает краткую информацию о преподавателях, которые будут обучать различным дисциплинам.

После разговора с абитуриентами игрока приглашают в кабинет приёмной комиссии. В помещении комиссия просит главного героя заполнить анкету с контактной информацией и сдать необходимые документы (аттестат, справку о здоровье, конверт, фотографии). (спросить средний балл, ввести имя и фамилию и т.д.)

**Мини-игры:**

Игра «Пинг-понг». В этой игре нужно перебрасывать с помощью ракетки мячик своему сопернику. Основной целью является забить как можно больше очков на поле соперника.

Игра «2048» в этой игре нужно соединить плитки с значением 2 и набрать максимальное значение. Игра завершиться при наборе числа 2048 или, когда не будет новых ходов и места для новых плиток.

Игра «Змейка». В данной игре нужно собрать яблочки чтобы накормить змейку. Цель игры вырастить большую змейку попутно избегая границ игрового поля и хвоста. В случае попадания на границы игрового поля или задевания хвоста змейки игра закончится и выведется окно с поражением.

**Требования к применению:**

Помогает адаптироваться студентам к темпу жизни колледжа. Программа разрабатывается для персональных компьютеров под управлением операционной системы Windows.

Минимальные требования: Windows 7 64-bit, ОЗУ 4 ГБ, Процессор Intel Pentium G450.

Требования к реализации: Для реализации игры будет использоваться язык программирования C#.

**Требования к интерфейсу:**

При разработке игры будет использована технология пиксель-графики. Оттенки в основном светлые, цвета приближены к настоящим аналогам. При первом запуске игры начнётся процесс создания персонажа.

**2 Техническое задание**

**2.1 Введение**

**2.1.1 Наименование программы**

Наименование программного продукта —«Политех»

**2.1.2 Краткая характеристика области применения**

Программа предназначена для людей, которые желают погрузиться в повседневность среднестатистических учащихся колледжей, высших учебных заведений и т.п.

**2.2 Основания для разработки**

Основанием для разработки является задание по предмету «Технология разработки программного обеспечения» на тему «Разработка программного обеспечения, предоставляющего досуг для пользователя» Тема согласована с преподавателем.

**2.3 Назначение разработки**

Основным назначением программы является изучение принципов создания программного обеспечения на языке программирования C#, создание программной документации.

**2.4 Требования к программе или программному изделию**

**2.4.1 Требования к применению**

Помогает адаптироваться студентам к темпу жизни колледжа. Программа разрабатывается для персональных компьютеров под управлением операционной системы Windows.

**2.4.2 Требования к производительности**

Программа должна запускаться на слабо конфигурированных системах.

Минимальные требования:

* Процессор Intel Pentium G450.
* ОЗУ 4 ГБ,
* 1 ГБ свободного места на жестком диске;
* Windows 7 64-bit.

Максимальное время на запуск программы на запуск программы при минимальных требованиях должно составлять не более 15 секунд.

**2.4.3 Требования к функциональным характеристикам**

Чтобы обеспечить возможность полного взаимодействия всех частей разрабатываемого приложения, необходимо автоматизировать следующие задачи:

— Просмотр глав;

— Создание персонажа: выбор имени, пола персонажа, внешности персонажа, одежды персонажа;

— Прохождение мини-игр;

— Сохранение игрового прогресса;

— Просмотр досье каждого из неуправляемых персонажей;

— Редактирование персонажа при необходимости: изменение одежды, внешности;

— Возможность вернуться в одну из прошлых глав при неудовлетворении, текущим исходом игры;

— Взаимодействие с неуправляемыми персонажами посредством диалога и выбора ответов, влияющих на исход игры;

— Просмотр небольшой части содержания следующих глав игры.

**2.4.4 Требования к реализации**

Для достижения максимальной производительности и оптимизации программа должна быть разработана на языке C# в среде Visual Studio 2022

**2.4.5 Требование к надёжности**

Приложение должно производить самовосстановление после сбоя работы.

**2.4.6 Требования к интерфейсу**

Интерфейс должен быть с приятной цветовой гаммой и понятной для пользователя.

Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информация, отображённой на экране. Объекты (рисунки и символы) не должны быть слишком мелкие.

При разработке игры будет использована технология пиксель-графики. Оттенки в основном светлые, цвета приближены к настоящим аналогам. При первом запуске игры начнётся процесс создания персонажа.

**2.5 Требования к программной документации**

Для удобства использования программного продукта пользователем, необходимо добавить файл справки, в котором будет находиться вся необходимая информация о работе с программой.

**2.6 Технико-экономические показатели**

Ориентировочная экономическая эффективность не рассчитываются. Предполагаемое число использования программой в год – 365 сеансов работы на одном рабочем месте.

**2.7 Стадии и этапы разработки**

**2.7.1 Стадии разработки**

Разработка должна быть проведена в три стадии: разработка технического задания, рабочее проектирование, внедрение.

**2.7.2 Этапы разработки**

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

— разработка программы;

— разработка программной документации;

— испытания программы.

На стадии внедрения должен быть выполнен этап разработки – подготовка и передача программы.

**2.7.3 Содержание работ по этапам**

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. Постановка задачи;

2. Определение и уточнение требований к техническим средствам;

3. Определение требований к программе;

4. Определение стадий, этапов и сроков разработки программы и

документации на неё;

5. Выбор языков программирования;

6. Согласование и утверждение технического задания.

На этапе разработки программной документации должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 с требованием п. Предварительный состав программной документации настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

— Разработка, согласование и утверждение программы и методики испытаний;

— Проведение приемо-сдаточных испытаний;

— Корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

**2.8 Порядок контроля и приёмки**

**2.8.1 Виды испытаний**

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

**2.8.2 Общие требования к приёмке работы**

На основе Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывают Акт приемки-сдачи программы в эксплуатацию.

**3 Руководство пользователя**

**1. Общие сведения о программном продукте**

Разработанная программа является игровым приложением.

Данный программный продукт предназначен для любителей играть в различного рода «ретро-игр».

Быстродействие любой программы во многом зависит от характеристик

выбранного персонального компьютера: рабочей частоты процессора, объёма оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, программа легко запускается и функционирует на любых компьютерах.

Тестирование проводилось на разных классах ЭВМ, но даже на системе

следующего типа работать было комфортно:

– ОС (операционная система): Windows XP/Vista/7/8;

– Процессор: Intel Core 2 Duo @ 3.0 Ghz / AMD Athlon 64 X2 6000+;

– Оперативная память: 1 Gb;

– Жесткий диск: 10 Gb свободно;

– Видео память: 512 Mb;

– Видео карта: nVidia GeForce 9800 / AMD Radeon HD 4870;

– Звуковая карта: Совместимая с DirectX;

– DirectX 9.0c;

**2. Выполнение программы**

Для того чтобы открыть игру нужно открыть проект в Unity.

После открытия проекта открывается главное окно (рисунок 1).



Рисунок 1 – Главное окно

**3.2 Инструкции по работе с программой**

Далее мы можем выбрать кнопку «На пары» и мы перейдем в окно, где на выбор будет предложено две опции – подать документы и пойти на учебу. В случае, если персонаж создан, опция подать документы будет недоступна, в противном случае – наоборот.

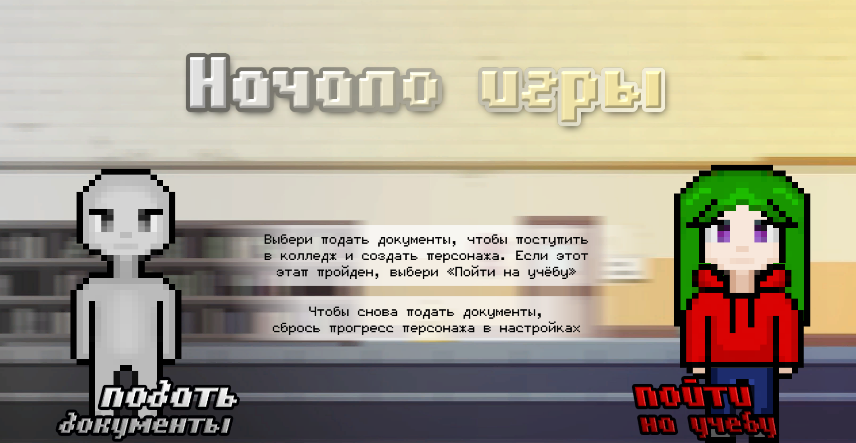


Рисунок 2 - Начало игры

Если выбрать опцию «Подать документы», то вы перейдете в окно для кастомизации персонажа.

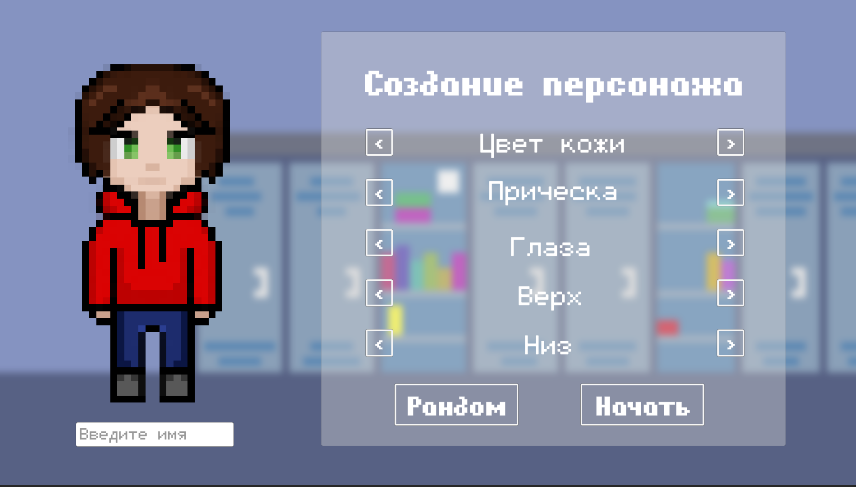


Рисунок 3 – Создание персонажа

С помощью кнопок Цвет кожи, Прическа, Глаза, Верх, Низ можно вполне удобно перемещать назад и вперед по доступным элементам. Кнопки отвечают за соответствующие элементы одежды/внешности персонажа.

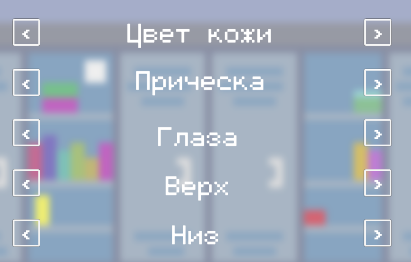


Рисунок 4 – Кнопки управления

Например, если игрока не устраивает прическа, цвет глаз и кожа персонажа, он может их сменить:

Рисунок 5, 6 – Пример изменений некоторых функций

Также на сцене присутствует поле для ввода имени для будущего персонажа. Оно вводится с клавиатуры и потом будет сохранено для игрового прогресса и диалогов.



Рисунок 7, 8 – Пример ввода имени

При нажатии на кнопку «Рандом» персонаж приобретает случайные черты внешности и гардероба.



Рисунок 9 – Пример рандомной кастомизации

**4 Тестирование и отладка ПО**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения этапа написания программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования представлен в таблице 1.

Таблица 1 – Отчёт о результатах тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Фактический результат | Результат тестирования |
| 1 | Проверка анимации фона на главной форме | После открытия сцены, фон не должен тормозить, исчезать после того как картинка дойдет до краев сцены, а дублироваться | Фон плавно движется, не исчезает, а незаметно перемещается | Выполнено |
| 2 | Проверка кнопок перемещения по элементу «Цвет кожи» | При нажатии по клавишам «>» и «<» одноименные элементы соответственно перемещаются назад и вперед | Элементы меняются, при нажатии на кнопку «>» перемещаются на следующий элемент в списке, на кнопку «<» на предыдущий | Выполнено |
| 3 | Проверка кнопок перемещения по элементу «Цвет глаз» | При нажатии по клавишам «>» и «<» одноименные элементы соответственно перемещаются назад и вперед | Элементы меняются, при нажатии на кнопку «>» перемещаются на следующий элемент в списке, на кнопку «<» на предыдущий | Выполнено |
| 4 | Проверка кнопок перемещения по элементу «Верх» | При нажатии по клавишам «>» и «<» одноименные элементы соответственно перемещаются назад и вперед | Элементы меняются, при нажатии на кнопку «>» перемещаются на следующий элемент в списке, на кнопку «<» на предыдущий | Выполнено |
| 5 | Проверка кнопок перемещения по элементу «Низ» | При нажатии по клавишам «>» и «<» одноименные элементы соответственно перемещаются назад и вперед | Элементы меняются, при нажатии на кнопку «>» перемещаются на следующий элемент в списке, на кнопку «<» на предыдущий | Выполнено |
| 6 | Проверка кнопок перемещения по элементу «Прическа» | При нажатии по клавишам «>» и «<» одноименные элементы соответственно перемещаются назад и вперед | Элементы меняются, при нажатии на кнопку «>» перемещаются на следующий элемент в списке, на кнопку «<» на предыдущий | Выполнено |
| 7 | Проверка работы кнопки «Рандом» | При нажатии на кнопку персонаж приобретает случайные элементы внешности и гардероба | При нажатии на кнопку персонаж приобрел случайные элементы внешности и гардероба | Выполнено |
| 8 | Проверка работы кнопки «Начать» | При нажатии на кнопку персонаж сохраняется в отдельный префаб, который потом может быть использован в любой другой сцене | При нажатии на кнопку персонаж сохранен в отдельный префаб, который потом может быть использован в любой другой сцене | Выполнено |
| 9 | Проверка поля ввода имени | Поле ввода позволяет ввести имя; После нажатия кнопки «Начать» имя сохраняется и потом может быть использовано в других сценах | Поле ввода позволяет ввести имя; После нажатия кнопки «Начать» имя сохранилось и потом может быть использовано в других сценах | Выполнено |
| 10 | Проверка изменения анимации персонажа при изменении элементов внешности и гардероба | При выборе любых элементов внешности или гардероба, анимация, связанная с ними, заменяет текущую и используется для движения влево-вправо | При выборе любых элементов внешности или гардероба, анимация, связанная с ними, заменилась текущей и используется для движения влево-вправо | Выполнено |

Вывод: все тесты были успешно выполнены, все обнаруженные ошибки исправлены. Весь изначально составленный функционал успешно реализован

***Приложение А***

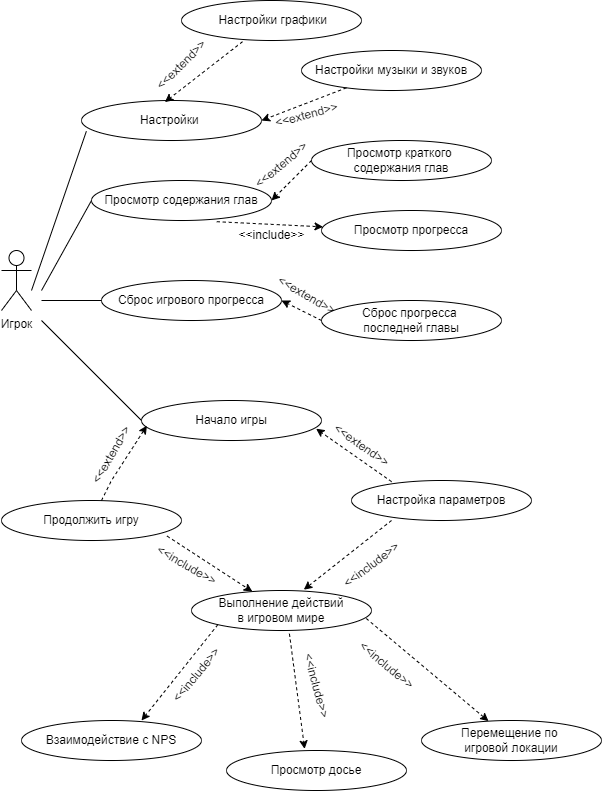


Рисунок 1 — Диаграмма вариантов использования

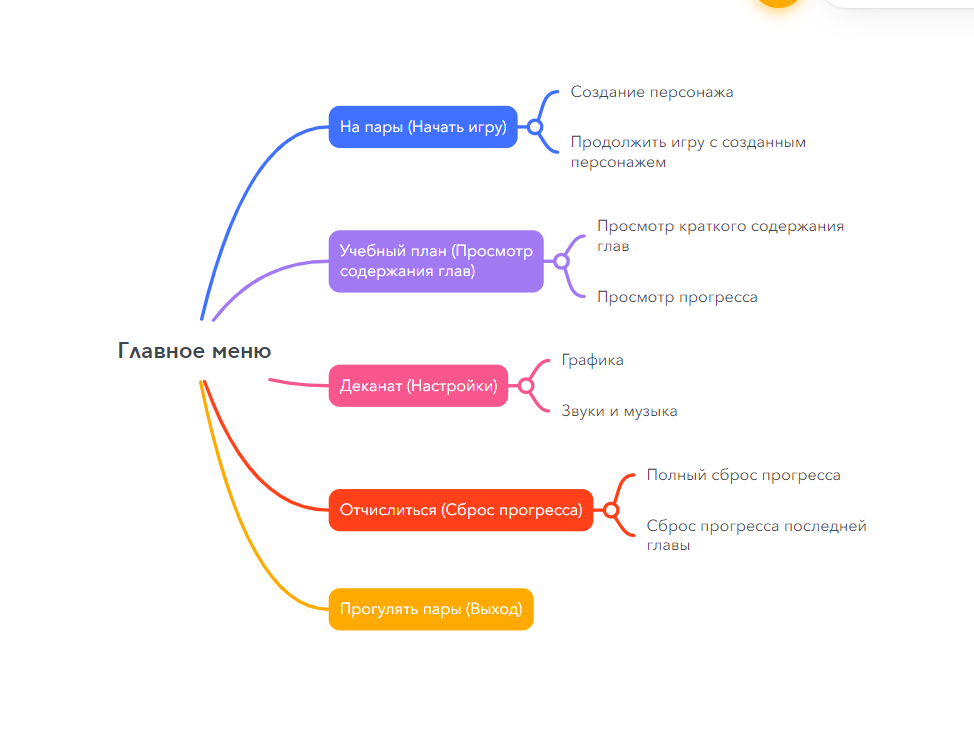


Рисунок 2 — Структура главного меню

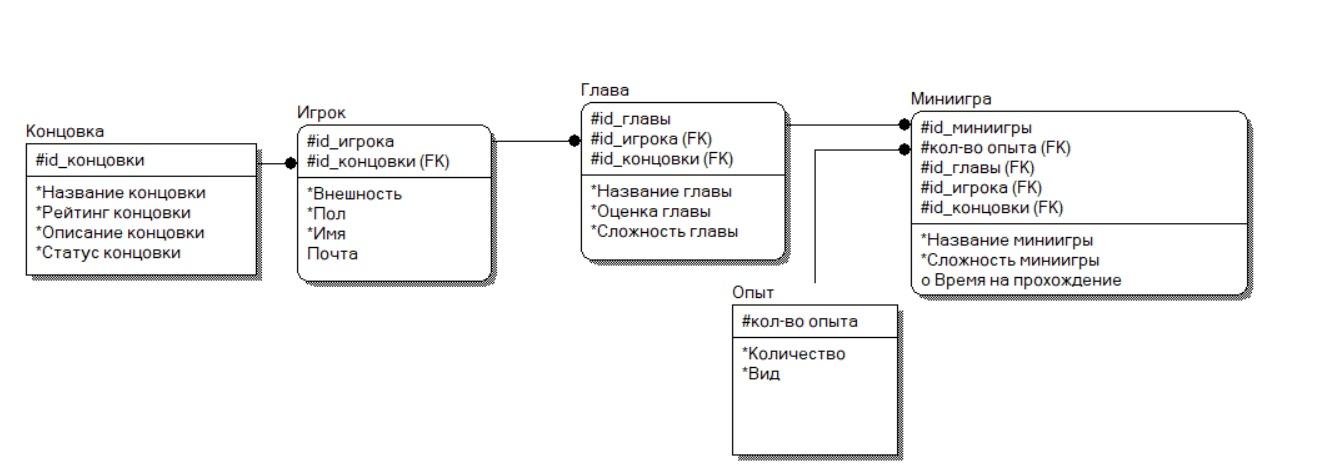


Рисунок 3 — Модель данных

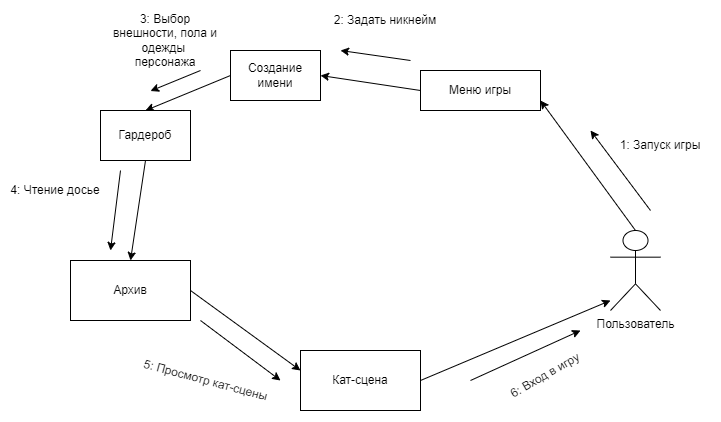


Рисунок 4 — Диаграмма объектов

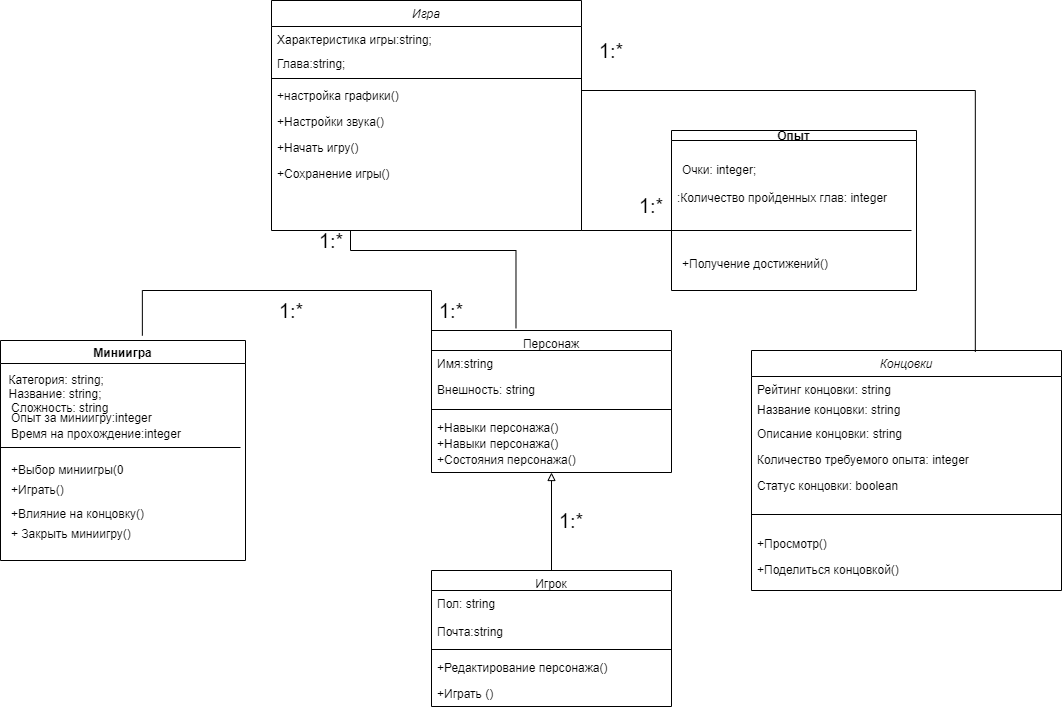


Рисунок 5 — Диаграмма классов

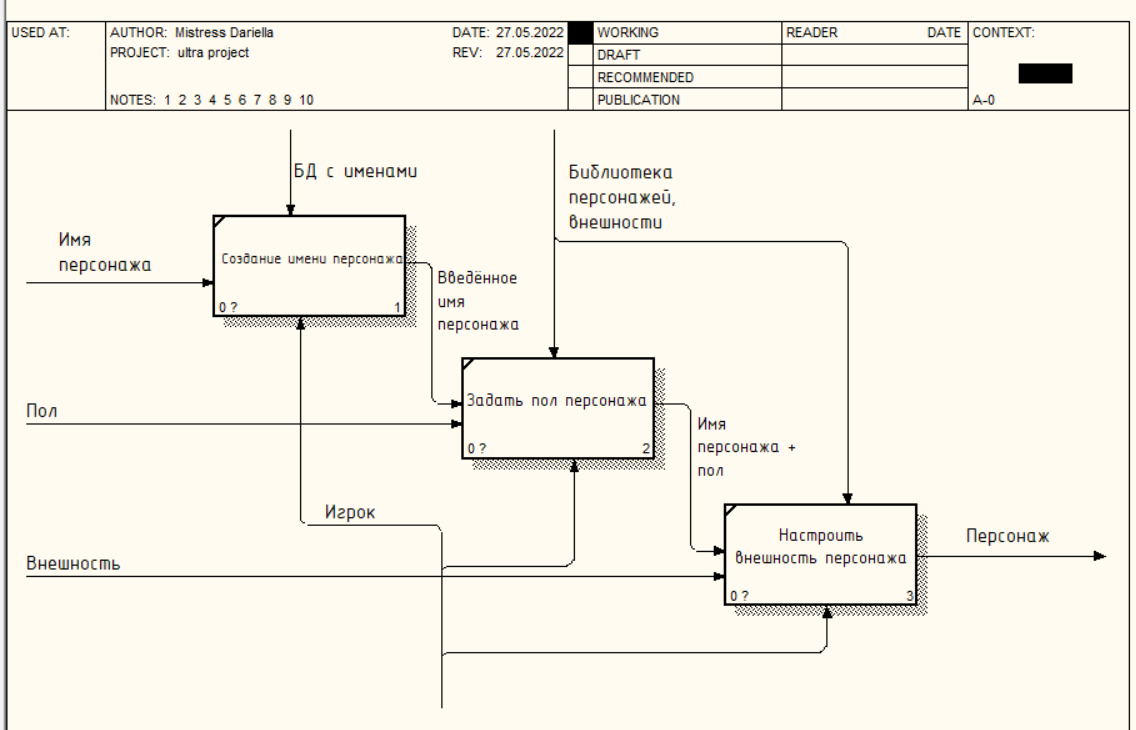
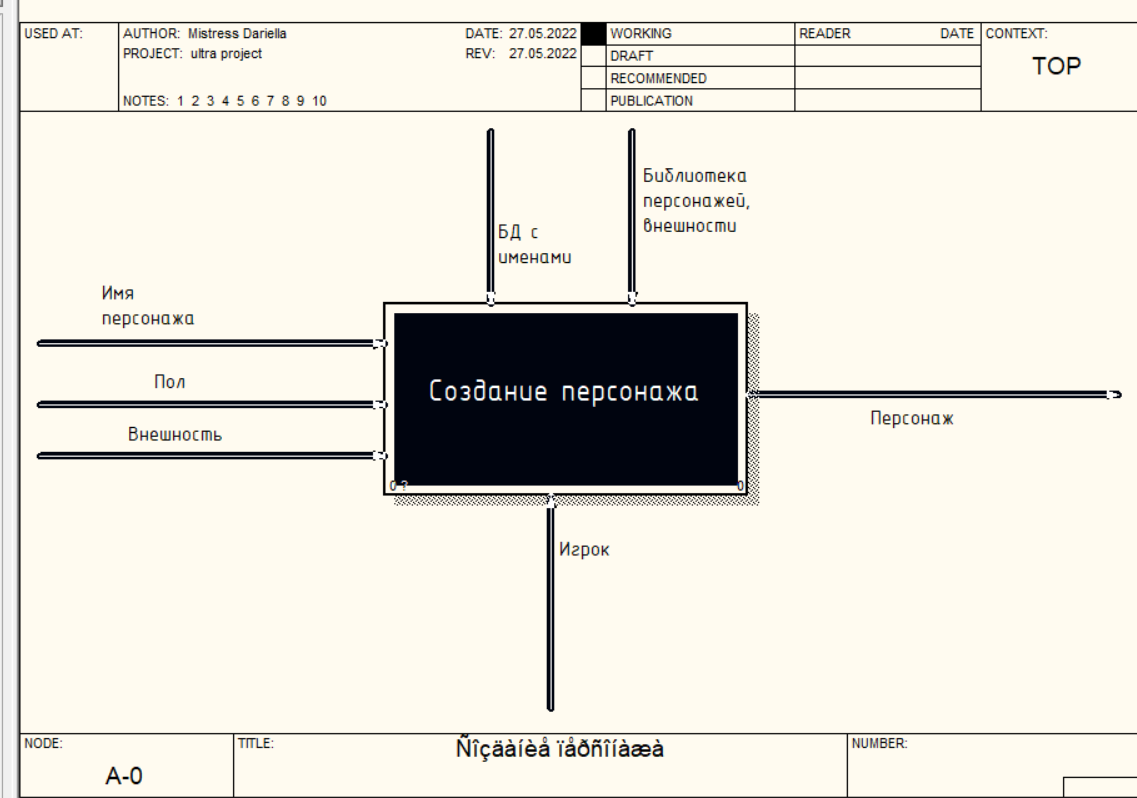


Рисунок 6 — Функциональная модель

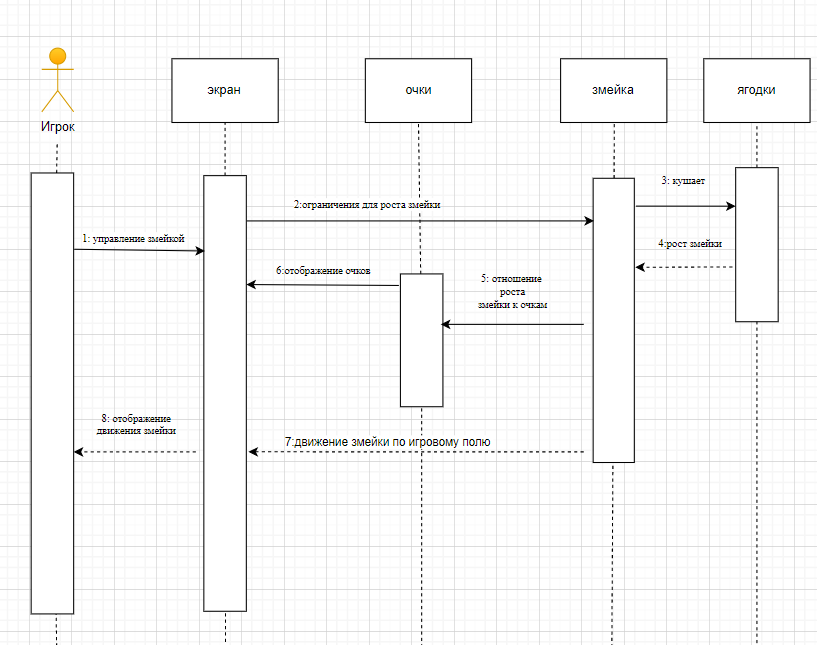


Рисунок 7 — Диаграмма последовательности

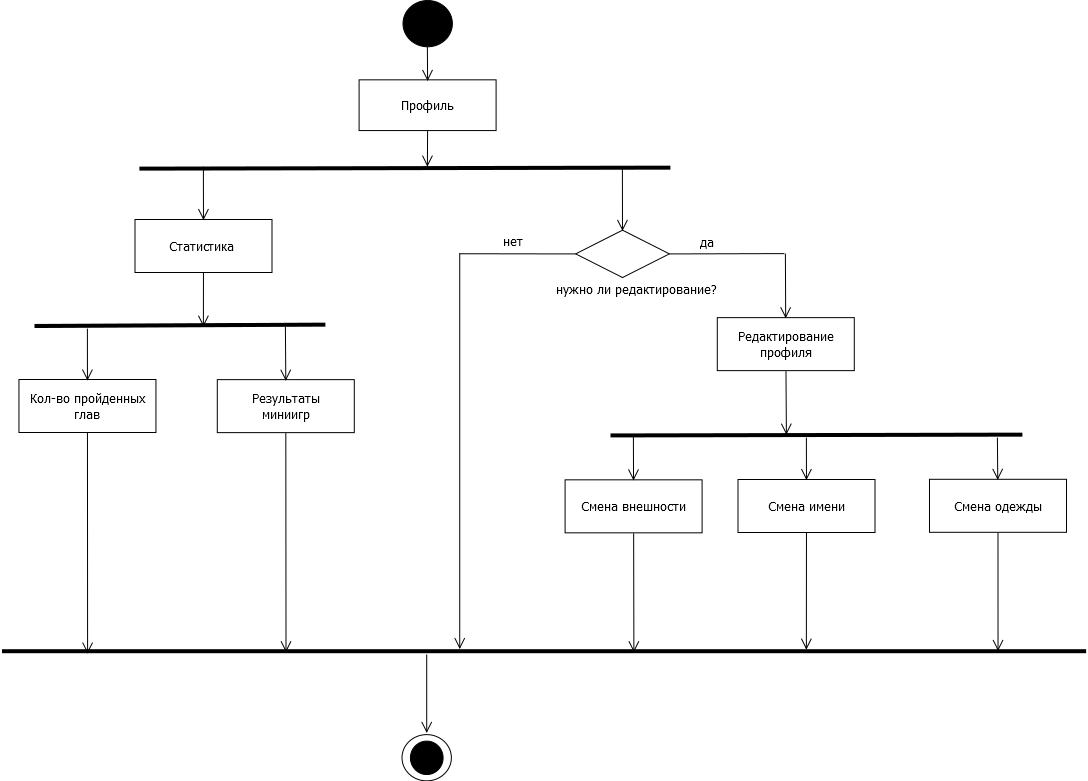
****

Рисунок 8 — Диаграмма деятельности



Рисунок 9 – UX прототип главного меню



Рисунок 10 - UX прототип начала игры

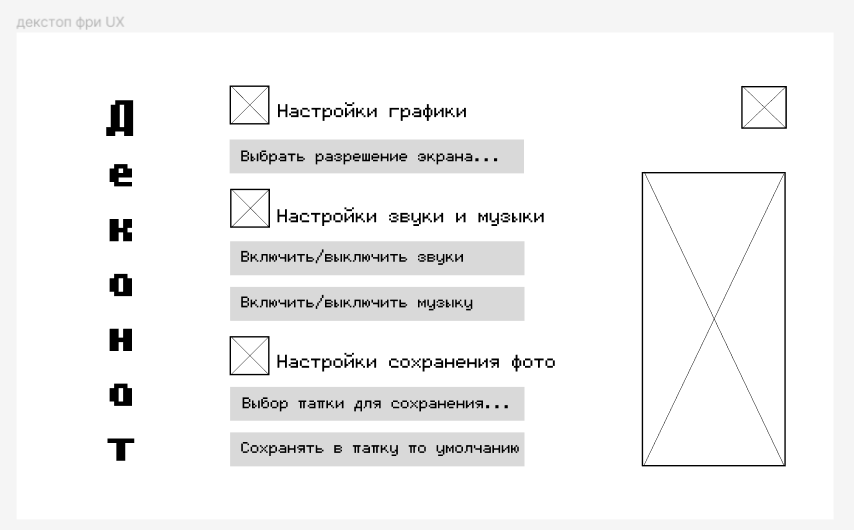


Рисунок 11 - UX прототип настроек

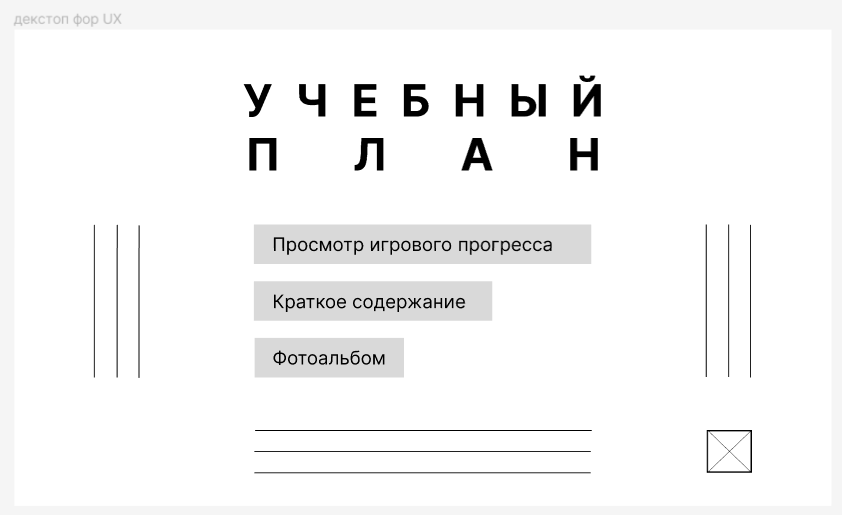


Рисунок 12 - UX прототип просмотра игрового процесса

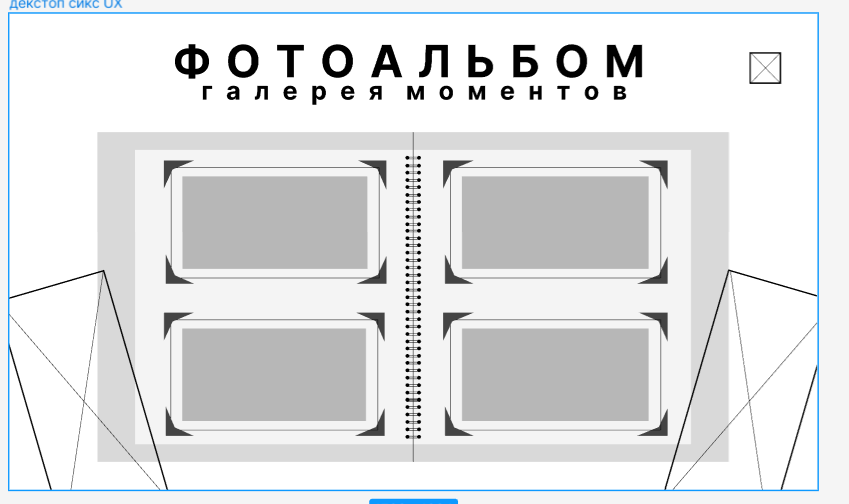


Рисунок 13 - UX прототип фотоальбома