**Введение**

На практику была поставлена задача, разработать программу на тему: «Игровое приложение о криптографии».

Цель практики заключается в знакомстве массового пользователя с возможными способами шифрования сообщений, их стойкостью и преимуществами.

Создаваемое приложение будет рассчитано для любого рода пользователей. Применить данную программу смогут не только люди, имеющее какое-то представление о криптографии, но и просто люди, заинтересовавшиеся в данной теме.

Далее приведём краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение её организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой создаётся данный проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет чётко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

«Реализация задачи» – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут чётко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов, используемых в данном проекте.

Четвёртый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе «Применение» будет описано назначение, область применения, среда функционирования программы. Также в нем будет описано использование справочной системы.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств.

В «Литературе» будет приведён список используемых при разработке источников.

В приложениях к пояснительной записке будет приведён листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

**1 Анализ задачи**

**1.1 Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

**Наименование задачи:** видеоигра “Crypto Dungeon”

**Цель разработки:** Создание видеоигры для обучения основам криптографии.

**Назначение:** Данный продукт разрабатывается для людей любого возраста, желающих изучить основы криптографии

**Периодичность использования ПП:** по желанию пользователя.

**Источники и способы получения данных:**

учебные пособия:

* Книга шифров Саймон Сингх

интернет статьи:

* <https://ru.wikipedia.org/wiki/Шифр_Виженера>
* <https://ru.wikipedia.org/wiki/Шифр_Цезаря>
* https://ru.wikipedia.org/wiki/AES\_(стандарт\_шифрования)

**Информационная связь с другими задачами:** отсутствуют.

**Обзор существующих аналогичных ПП:**

В ходе анализа был обнаружен продукт из данной предметной области. Decipherism является хорошей игрой из данной области, однако от нашего продукта очень сильно отличается, так как выполнен в 2д, а также имеет гораздо меньший функционал, являясь только иллюстратором работы шифров, а также маленьким задачником.

Геймплей:

Жанр данной игры – инди-игра, приключения.

Игрок посещает различные планеты, на которых находятся заброшенные строения. В этих местах он может отыскать шифры и сокровища. Шифры раскрывают историю всего мира или места, где шифр был найден. Так же раскрытые шифры могут вести на новые локации и содержать в себе координаты новых мест с сокровищами.

В игре имеется игровая валюта, которая начисляется за продажу сокровищ на базе через компьютер. После продажи предмет исчезает из инвентаря, а игрок получает игровую валюту.

Каждое строения является процедурно генерируемым.

Чтобы разгадать шифр требуется купить специальное оборудование для ручной дешифровки или брутфорса ключа. Оборудование можно улучшать, повышая его скорость. Оборудования размещается на базе игрока.

Покупка предметов и оборудования реализуется через компьютер на базе игрока. Купленное оборудование доставляется в коробках, после нажатия на которые размещается автоматически на базе или попадает в инвентарь.

Игрок может купить некоторые предметы, которые обладают свойствами временно используемых, например, одноразовый взлом дверей.

Предметы и сокровища, найденные игроком, размещаются в инвентаре, который ограничен определённом количеством ячеек. Шифры неограниченно размещаются в папке пользователя.

Пользователь имеет папку, в которой в любой момент может просмотреть найденные шифры и перечитать их.

У игрока есть корабль, который он используется для путешествия между разными локациями и базой.

В строениях находятся монстры, которые мешают игроку забрать сокровища и шифры, нанося ему урон. Если игрок теряет все свои жизни и товарищей, то игра начинается заново, сбрасывая весь прогресс игроков. В случае, если хотя бы один игрок вернулся на базу живым, то остальные игроки возродятся, потеряв своё оборудование и игровую валюту. Игрок стремится к богатству и знаниям, постепенно открывая всё новые и новые загадки.

Игрок улучшает свою базу и закупает новое оборудование. Исследует заброшенные строения, побеждает монстров и забирает сокровища.

**1.1.2 Функциональные требования**

**Описание перечня функций и задач, которые должен выполнять будущий ПП:**

**Пользователь:**

1. Изучение различных шифров;
2. Дешифровка сообщений;
3. Получение обновлений игры;
4. Отслеживание прогресса и достижений в игре.

**1.1.3 Входная информация**

* Уровень сложности, выбранный пользователем;
* Настройки игры.

**1.1.4 Выходная информация**

* Статистика игры, прогресс пользователя, достижения.

**1.1.5 Условно-постоянная информация**

* Правила игры, описание уровней, параметры приборов.
* Текстовые документы, таблицы.

**1.1.6 Эксплуатационные требования**

Требования к применению: игра предоставляет возможность быстро начать игру, сохранить прогресс и получить обратную связь от системы, а также возможность узнать о статусе текущих заданий и достижений.

Требования к реализации: игра будет разработана с использованием современных технологий, таких как Unity для обеспечения высокого качества графики и производительности. Продукт поддерживает многопользовательский режим.

Требования к надежности: игра будет стабильной и надежной, с минимальными перебоями в работе.

Требования к интерфейсу: игра будет обладать интуитивно понятным и удобным интерфейсом. Будут предоставлены различные настройки для адаптации игры под предпочтения пользователя. Игра будет поддерживать различные языки интерфейса. Продукт будет иметь возможность получения помощи или подсказок внутри игры.

Требования к обратной связи: игра будет предоставлять обратную связь игроку о его прогрессе и достижениях, будет предусмотрена возможность обратной связи от игроков для улучшения игры.

Приложения будет ориентировано на десктоп, 64-битную систему под windows от 7 версии.

Минимальные требования:

* ОС: Windows 7/8/10
* Процессор: Dual-core от 2.0 ГГц
* Оперативная память: 8 ГБ
* Видеокарта: NVIDIA GeForce GTX 1050
* Место на диске: 2 ГБ свободного пространства
* DirectX: Версия 11

**1.2 Инструменты разработки**

Для разработки проекта выбран игровой движок Unity версия 2022.3.25f1 с использованием языка C#, так как он является объектно-ориентированным и включает в себя большое количество удобного функционала для разработки 3д, 2д, а также игр с дополненной реальностью и виртуальной реальностью.

Выгоды от проектирования в среде Windows с помощью Unity:

* устраняется необходимость в повторном вводе данных;
* обеспечивается согласованность проекта и его реализации;
* увеличивается производительность разработки и переносимость программ.

Также для разработки программы необходимы:

* Help & Manual – редактор для создания справок;
* Visual Studio 2022 – редактор для написания кода;
* Word 2021 – необходим для написания пояснительной записки;
* Power Point 2021 – необходим для создания отчётной презентации;
* Drawio – сайт для создания диаграмм и схем;
* MagicaVoxel – программа для создания воксельных 3д моделей;
* Блокнот – для создания сохранений.

При разработке данного программного продукта был использован компьютер со следующими характеристиками:

* процессор Intel Core i5;
* объем ОЗУ 16 гб;
* объем места на HDD – 1 тб ;
* видеоподсистема 1920х1080 точек с глубиной цвета 32 Bit;
* ОС – Windows 10.

**1.3 Выбор модели ЖЦ**

Таблица 1 - Выбор модели жизненного цикла на основе характеристик требований.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементна я | Быстрого прототипиро вания | Эволюционн ая |
| 1. | Являются ли требования к  проекту легко определимыми и реализуемыми? | Да | Да | Да |  |  |  |
| 2. | Могут ли требования быть  сформулированы в начале ЖЦ? | Да | Да | Да | Да |  |  |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? |  |  |  | Да | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? |  |  | Да | Да | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или  системы? |  |  | Да | Да | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? |  |  | Да | Да | Да | Да |

Вычисления: 2 за каскадную, 2 за V-образную, 5 за RAD, 6 за инкрементную, 5 за быстрого прототипирования и 5 за эволюционную.

Итог: на основе результатов заполнения табл. 1 подходящей является инкрементная модель.

Таблица 2 - Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементн ая | Быстрого прототипирова ния | Эволюционн ая |
| 1. | Являются ли проблемы  предметной области проекта новыми для большинства разработчиков? |  |  |  | Да | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для  большинства разработчиков? | Да | Да |  |  |  | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? |  |  |  | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да |  | Да |  |  |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да |  |  |
| 6. | Приемлет ли команда  разработчиков оценки, проверки, стадии разработки? | Да | Да |  | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V-образную, 1 за RAD, 5 за инкрементную, 3 за быстрого прототипирования и 4 за эволюционную.

Итог: на основе результатов заполнения табл. 2 подходящей является инкрементная модель.

Таблица 3 - Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементн ая | Быстрого прототипирова ния | Эволюционн ая |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да |  | Да | Да | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного  продукта (системы) в процессе разработки? |  |  |  | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? |  |  | Да | Да | Да | Да |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? |  |  |  | Да | Да | Да |

Вычисления: 1 за каскадную, 1 за V-образную, 1 за RAD, 4 за инкрементную, 4 за быстрого прототипирования и 4 за эволюционную.

Итог: на основе результатов заполнения табл. 3 подходящими являются инкрементная, быстрого прототипирования и эволюционная модели.

Таблица 4 - Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-  образная | RAD | Инкрементн ая | Быстрого прототипиров ания | Эволюцион ная |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? |  |  |  | Да | Да | Да |
| 2. | Будет ли проект являться  расширением существующей системы? | Да | Да | Да | Да |  |  |
| 3. | Будет ли проект крупно- или среднемасштабным? |  |  |  | Да | Да | Да |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да |  | Да |  | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? |  | Да |  | Да | Да | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? |  |  |  | Да | Да | Да |
| 8. | Является ли график сжатым? |  |  | Да | Да | Да | Да |
| 9. | Предполагается ли повторное использование компонентов? |  |  | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? |  |  |  |  | Да | Да |

Вычисления: 2 за каскадную, 3 за V-образную, 4 за RAD, 9 за инкрементную, 8 за быстрого прототипирования и 9 за эволюционную.

Итог: на основе результатов заполнения табл. 4 подходящей является инкрементная модели.

Общий итог: в итоге заполнения табл. 1 – 4 наиболее подходящей является инкрементная модель, однако так как не все требования известны, по итогу выбрана эволюционная модель.

Общая статистика по всем таблицам:

Каскадная: 9

V-образная: 10

RAD: 11

Инкрементная: 24

Быстрого прототипирования: 20

Эволюционная: 22