

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Восточно-Сибирский государственный университет технологий и управления»

Электротехнический факультет
Кафедра систем информатики

Курсовой проект
по дисциплине «Программирование»
Тема:
«Реализация редактора космической ракеты»

Выполнил:	студент гр. Б661
_____	Коковихин А.В.
Руководитель:	стар.преп. каф. СИ
_____	Мердыгеев Б.Д.
Оценка:	_____
Дата защиты:	_____

Улан-Удэ
2022

ВОСТОЧНО-СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ
ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра систем информатики

З А Д А Н И Е

на курсовой проект

Дисциплина: Программирование

Тема: Реализация редактора космической ракеты

Исполнитель: Коковихин А.В.

Руководитель: Мердыгеев Б.Д.

Краткое содержание проекта: данная курсовая работа посвящена написанию редактора космической ракеты, прототипом для которой является редактор из игры: «Kerbal Space Programm» (KSP)

1. Теоретическая часть: словесная постановка задачи, описание оригинальной игры, Описание функций редактора, отличия от оригинала, специфика консольных приложений

2. Практическая часть: формальная постановка задачи, UML модель, алгоритм решения задачи, реализация управления интерфейсом «горячими» клавишами, разработка приложения и тестирование полученного программного продукта

Сроки выполнения работы по календарному плану*:

Этап 1. Теоретический раздел – 15% к 5 неделе.

Этап 2. Проектный раздел – 40% к 8 неделе.

Этап 3. Программный раздел – 70% к 12 неделе.

Этап 4. Экспериментальный раздел – 90% к 14 неделе.

Этап 5. Защита – 100 % к 16 неделе.

Требования к оформлению:

1. Отчет по курсовой работе должен быть представлен в электронной и твердой копиях.
2. Объем отчета должен быть не менее 20 машинописных страниц без учета приложений.
3. Отчет оформляется по ГОСТ 7.32-2001.

Руководитель работы _____

Исполнитель _____

Дата выдачи " ____ " _____ 2020 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Теоретический раздел.....	6
1.1 Словесная постановка задачи	6
1.2 Описание оригинальной игры (KSP)	6
1.3 Описание функций редактора (продукта)	7
1.4 Отличия от оригинала	7
1.5 Специфика консольных приложений	8
2 Проектный раздел.....	12
2.1 Формальная постановка задачи	12
2.2 UML модель	13
2.3 Алгоритм решения задачи	15
2.4 Реализация управления интерфейсом «горячими» клавишами.....	16
3 Программный раздел	20
3.1 Описание программы	20
3.2 Описание структуры данных	20
3.3 Описание основных функций.....	21
4 Экспериментальный раздел	24
4.1 Тестирование в нормальных условиях	24
4.2 Тестирование в исключительных условиях	28
4.3 Тестирование в экстремальных условиях.....	31
4.4 Итоги тестирования	31
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30
Приложение А.....	30
Define.h	30
Point2.h.....	30
Point2.cpp.....	30
Main.cpp.....	31
Приложение В.....	38
Part.h	38
PartConnector.h.....	38
Part.cpp	39
PartConnector.cpp.....	43
Приложение С.....	45
Window_Args.h	45

					Д.661.1.4.22.31.025.22.ПЗ						
Изм.			Подпись	Дата							
Разраб.	Коковихин А.В.				Редактор космической ракеты			Лит.	Лист	Листов	
Провер.	Мердыгеев Б.Д.									4	1
Консультант								ВСГУТУ			
Н. Контр.											
Утв.											

ВВЕДЕНИЕ

Недавно по сети прокатился тренд на создание ASCII графики. И 3-х мерный рендеринг вращающегося бублика, Doom в консоли и т.д. Вдохновившись этой идеей было принято решение создать полноценный редактор космической ракеты используя ASCII графику.

Результатом работы является игра, работающая на компьютере с ОС Windows. Данная реализация является наиболее приближенной к оригиналу, но при этом обладает собственной уникальной графикой.

Целью курсовой работы является закрепление теоретических знаний и получение практических знаний по объектно-ориентированному программированию. Для достижения поставленной цели были решены следующие задачи:

- 1) анализ предметной области;
- 2) разработка объектной модели;
- 3) создание спрайтов деталей
- 4) разработка интерфейса и управления;
- 5) разработка программы;
- 6) тестирование работоспособности программы.

Расчетно-пояснительная записка состоит из:

- теоретической части, содержащей словесную постановку задачи, описание оригинальной игры, описание функций редактора, отличия реализации от оригинала, специфика консольных приложений;
- практической части, содержащей формальную постановку задачи, UML модель и алгоритм работы программы.

					Д.661.1.4.22.31.025.22.ПЗ		
Изм.			Подпись	Дата	Редактор космической ракеты		
Разраб.	Коковихин А.В.						
Провер.	Мердыгеев Б.Д.						
Консультант							
Н. Контр.							
Утв.							
					Лит.	Лист	Листов
						5	1
					ВСГУТУ		

1 Теоретический раздел

1.1 Словесная постановка задачи

Необходимо написать программу, повторяющую функционал «Цеха вертикальной сборки» из игры (Kerbal Space Program) KSP, используя ASCII графику. Редактор должен быть приспособлен к деталям различных форм и размеров, так же в нём должны быть реализованы механики присоединения, удаления и добавления деталей. Задачу необходимо решить, используя объектно-ориентированное программирование и язык C++ без использования сторонних фреймворков и движков. Дополнительно требуется создать дружелюбный пользовательский интерфейс используя для отрисовки только консоль.

1.2 Описание оригинальной игры (KSP)

Kerbal Space Program – Игра песочница, симулятор космической компании: «Почувствуйте себя Илоном Маском». Игроку предстоит:

- проектировать ракеты, самолёты, спутники, роверы и прочие «машины» для того, чтобы проводить исследования в Солнечной системе и выполнять контракты;
- пилотировать собственные миссии;
- исследовать новые технологии и закупать новое оборудование/улучшать здания.

В цехе вертикальной сборки корабли центрируются по вертикальной оси, в нём игрок может размещать детали на проектируемый корабль перетаскивая их из списка деталей на ту позицию, где он её хочет закрепить. У деталей есть свои «точки крепления» для ровной установки одной детали на другую, чаще всего их 2: сверху и снизу. Помимо этого, игроку доступна функция симметричного копирования детали вокруг оси, для этого он выбирает число копий и устанавливает первую деталь, а уже редактор ставит остальные копии на свои места. Следующее «удобство» проецирование детали и её копий: как она будет расположена если подтвердить её размещение сейчас.

Правила редактора:

- первая деталь – главная, все остальные крепятся к ней;
- если отсоединить деталь - то отсоединятся все прикреплённые к ней;
- детали не прикреплённые к главной будут «неактивными», но их можно перемещать, и они не будут учувствовать при запуске;
- детали не должны пересекаться.

					Д.661.1.4.22.31.025.22.ПЗ			
Изм.			Подпись	Дата				
Разраб.	Коковихин А.В.				Редактор космической ракеты	Лит.	Лист	Листов
Провер.	Мердыгеев Б.Д.						6	3
Консультант						ВСГУТУ		
Н. Контр.								
Утв.								

2 Проектный раздел

2.1 Формальная постановка задачи

Входные данные: горячие клавиши, файлы сохранённых проектов:

- ←/a/ф/4 (Влево), ↑w/ц/8 (Вверх), →/d/в/6 (Вправо), ↓/s/ы/2 (Вниз) – для навигации по интерфейсу;
- 0, Space, Enter (OK) – для подтверждения действия;
- Esc (Cancel) – для закрытия списка деталей/перехода к главной детали;
- Ctrl + C (Copy), Ctrl + S (Save), Ctrl + O (Open) – для копирования в буфер, сохранения, открытия файла;
- F2, Ctrl + H, Ctrl + I (Help) – выводит сообщение со справочной информацией;
- Ctrl + Q (Quit) – закрывает программу;
- Файлы содержащие необходимую информацию для загрузки проекта (*.asdat).

Выходные данные: графическое отображение интерфейса, а также информационные сообщения (диалоговые окна).

Метод решения: для решения задачи должны быть созданы классы для следующих объектов: Window_Args – хранящий информацию необходимую для отрисовки интерфейса (движок), Window_Parts_Panel_Args – подкласс Window_Args отвечающий за отрисовку списка деталей, Part – класс детали ракеты, PartConnector – точка для соединения деталей между собой, Point2 – двумерный вектор (используется как позиция или размер). А также перечислений: Color – 8-ми битный цвет и Part_Type – тип детали. Основными методами классов должны быть вывод графического изображения объекта и изменение его данных. В свою очередь «спрайты» деталей будут «вшиты» в соответствующие Get методы, например Get_Cockpit() представлен на рисунке 3.

```
Part* Get_Cockpit()
{
    const Point2 center = Point2(6, 3);
    const Point2 size = Point2(12, 5);
    char** sprite = new char* [5]
    {
        new char[12] {R"( /| | | \ )"},
        new char[12] {R"( /| | / \ | \ )"},
        new char[12] {R"([| | \ /| | |)"),
        new char[12] {R"([| | | <=>| | |)"),
        new char[12] {R"([| | | <=>| | |)"),
    };
    int con_count = 4;
    PartConnector* connectors = new PartConnector[4]
    {
        PartConnector(Point2(0, -2), Point2(0, -1)),
        PartConnector(Point2(5, 1), Point2(1, 0)),
        PartConnector(Point2(0, 2), Point2(0, 1)),
        PartConnector(Point2(-5, 1), Point2(-1, 0)),
    };
}
```

Рисунок 3 – Пример спрайта

					Д.661.1.4.22.31.025.22.ПЗ						
Изм.			Подпись	Дата							
Разраб.		Коковихин А.В.				Лит.		Лист		Листов	
Провер.		Мердыгеев Б.Д.						12		6	
Консультант						ВСГУТУ					
Н. Контр.											
Утв.											
					Редактор космической ракеты						

3 Программный раздел

3.1 Описание программы

Данная программа реализована на языке C/C++ и выполнена в среде разработки Visual Studio 2019. В ней используются библиотеки *Windows.h*, *cstdio*, *chrono*, *thread*, *stdlib.h*, *conio.h*, *stdio.h*, *iostream*, *shobjidl.h*, подключаемые в файле *Define.h*.

Так же подключаются в *header.h* перечислены все классы(структуры) и перечисления: *struct Point2*, *enum class Color*, *struct Part*, *struct PartConnector*, *struct Window_Args*, *enum class Part_Type*. Определения самих классов, находятся в соответствующих файлах с расширением «.h», а реализация некоторых методов в файлах с расширением «.cpp».

Данная программа является много файловым проектом, функция *main* представлена в файле «main.cpp».

3.2 Описание структуры данных

Данная программы была написана, используя объектно-ориентированный подход, с использованием классов, инкапсуляции и полиморфизма, механизмов управления памятью и сборкой мусора. А что касается «сохранения» данных то для это цели есть отдельные скрипты:

- В файл сохраняется только количество деталей; типы, позиции, с какой деталью и через какой коннектор соединён коннектор каждой детали (рис. 7);
- В буфер формируется картинка только спрайтов всех деталей, без лишних пробелов (рис. 8).

```
FILE* f = NULL;
fopen_s(&f, convertWstringToAstringWinapi(wstring(file_path)).c_str(), "w");
fwrite(&args.parts_count, sizeof(int), 1, f);
for (int i = 0; i < args.parts_count; i++)
{
    Part* p = &args.parts[i];
    fwrite(&p->Type, sizeof(Part_Type), 1, f);
    fwrite(&p->Position, sizeof(Point2), 1, f);
}
int NULL_key = -1;
for (int i = 0; i < args.parts_count; i++)
{
    Part* p = &args.parts[i];
    for (int ic = 0; ic < p->Connectors_Count; ic++)
    {
        PartConnector* c = &p->Connectors[ic];
        if (c->Connected == NULL)
            fwrite(&NULL_key, sizeof(int), 1, f);
        else
        {
            fwrite(&c->Connected->Main->Index, sizeof(int), 1, f);
            fwrite(&c->Connected->Index, sizeof(int), 1, f);
        }
    }
}
```

Рисунок 7 – Скрипт записи в файл.

					Д.661.1.4.22.31.025.22.ПЗ		
Изм.			Подпись	Дата	Редактор космической ракеты		
Разраб.	Коковихин А.В.						
Провер.	Мердыгеев Б.Д.						
Консультант							
Н. Контр.							
Утв.							
					Лит.	Лист	Листов
						20	3
					ВСГУТУ		

4 Экспериментальный раздел

4.1 Тестирование в нормальных условиях

Было проведено тестирование игры на наличие критичных ошибок и багов. И все команды в обычных условиях выполняются корректно.

Так на рисунке 7 видно, что сверху выведен текст с горячими клавишами для вызова справки, открылся режим выбора детали, и «курсор» – выделение зелёным цветом – расположен на первой детали первой страницы.

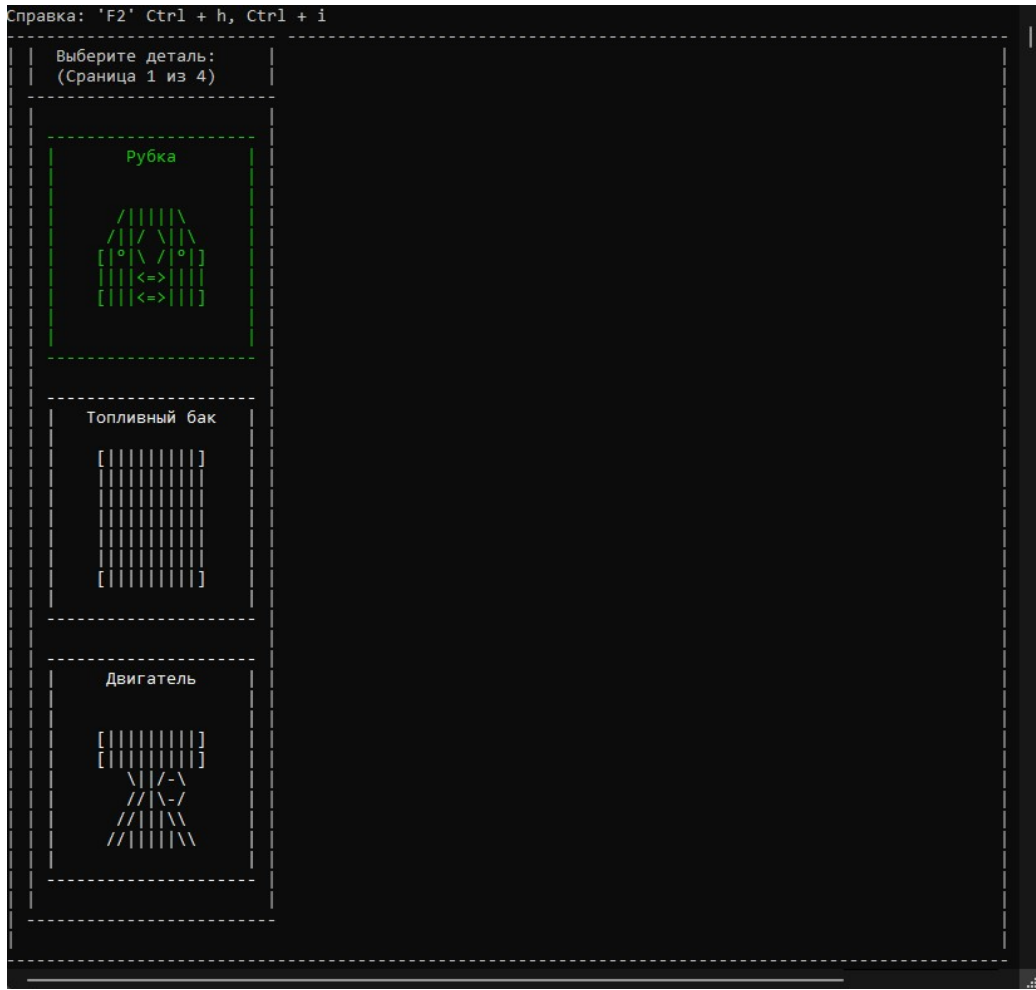


Рисунок 7 – Запуск

На рисунках 8, 9 продемонстрирован отклик интерфейса на команду (Вниз) и последующую команду (Вверх), а на рисунках 10, 11 – отклик на (Вправо), (Влево).

					Д.661.1.4.22.31.025.22.ПЗ			
Изм.			Подпись	Дата				
Разраб.	Коковихин А.В.				Редактор космической ракеты	Лит.	Лист	Листов
Провер.	Мердыгеев Б.Д.						24	8
Консультант						ВСГУТУ		
Н. Контр.								
Утв.								

ЗАКЛЮЧЕНИЕ

В ходе разработки была создана удобная база и возможности для будущих версий программы. В неё с лёгкостью можно добавить другие типы деталей, стилизовать список деталей чтобы он выводил миниатюры, создать другие классы деталей, например контейнеры, добавить цветов, поработать над интерфейсом и т.д.

Так же в ходе разработки была идея сделать управление мышью, т.к. в консоли есть так называемое «выделение», можно попытаться её реализовать в будущем.

Вывод: проект доведён до финальной стадии: ключевые механики реализованы, все неисправности были устранены на этапе тестирования, что свидетельствует о том, что все поставленные цели достигнуты, а задачи выполнены.

					Д.661.1.4.22.31.025.22.ПЗ						
Изм.			Подпись	Дата							
Разраб.	Коковихин А.В.				Редактор космической ракеты			Лит.	Лист	Листов	
Провер.	Мердыгеев Б.Д.									29	1
Консультант								ВСГУТУ			
Н. Контр.											
Утв.											

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Эккель, Б. Философия C++ [Текст]: учебник / Б. Эккель, Чак Эллисон. – М.: Питер, 2004. – 577 с.
2. Лафоре, Р. Объектно-ориентированное программирование в C++ [Текст]: учебник / Р. Лафоре. – М.: Питер, 2004. – 992 с.
3. Kerbal Space Program [Электронный ресурс]: – URL: <https://www.kerbalspaceprogram.com/game/kerbal-space-program/>
4. Чистый код. Создание, анализ и рефакторинг [Текст]: Книга / Мартин Роберт К: Питер 2010. – 464 с.
5. Кондитерская программиста [Электронный ресурс]: Habr.com – URL: <https://habr.com/ru/post/650011/>

					Д.661.1.4.22.31.025.22.ПЗ					
Изм.			Подпись	Дата						
Разраб.		Коковихин А.В.				Редактор космической ракеты	Лит.	Лист	Листов	
Провер.		Мердыгеев Б.Д.						30	1	
Консультант							ВСГУТУ			
Н. Контр.										
Утв.										