

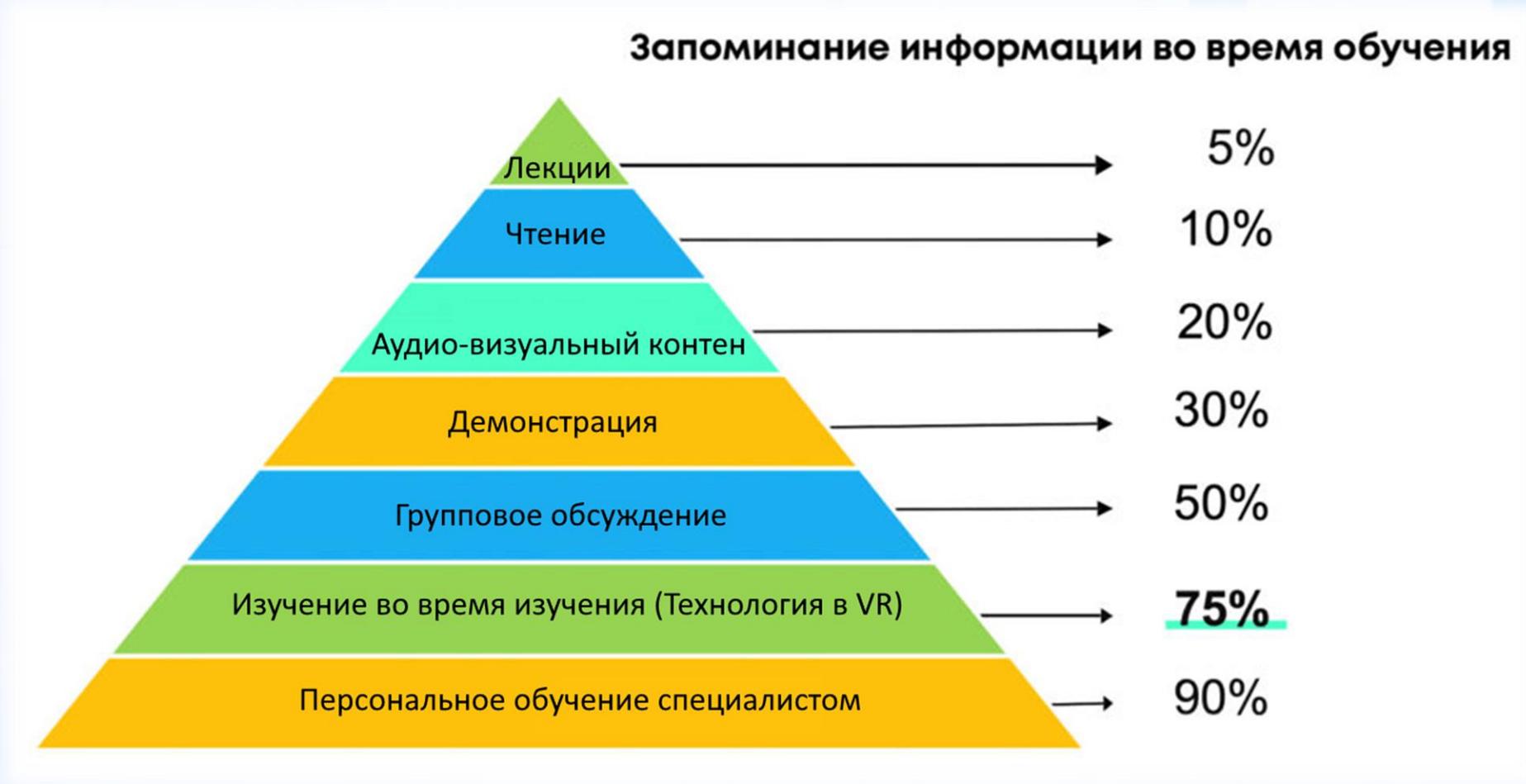
Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

Разработка симулятора для изучения теории графов

Научный руководитель:
к.т.н., доцент кафедры ЭВМ
К. А. Домбровский

Автор работы:
студент группы КЭ-406
П. Е. Некрасов

Актуальность темы



Цели и задачи

Целью работы является создание симулятора для изучения теории графов.

Задачи:

1. Обзор и анализ программного обеспечения для создания графов.
2. Анализ и выбор средств реализации проекта.
3. Архитектурное проектирование приложения.
4. Программная реализация и тестирование проекта.

Анализ и сравнение аналогов

Проект	Недостатки
Graphviz	<ul style="list-style-type: none">• отсутствие пользовательского интерфейса;• необходимость знания языка DOT.
yEd Graph Editor	<ul style="list-style-type: none">• перегруженность пользовательского интерфейса.
Graph Builder	<ul style="list-style-type: none">• небольшой список предоставленных алгоритмов;• возможность возникновения ошибок при длительном использовании приложения;• возникновение проблем с работоспособностью на последних версиях операционной системы.
Граф Онлайн	<ul style="list-style-type: none">• невозможность осуществлять работу в офлайн режиме;• ограниченный набор функций.
Редактор графов от ООО «Новый семестр»	
Редактор графов веб-ресурса Programforyou	
Онлайн-версия Graphviz	

Функциональные требования

Симулятор должен предоставлять возможность:

- получения теоретической информации по теории графов;
- самостоятельного построения графов;
- изменения построенного графа;
- применения алгоритмов на построенном графе;
- переключения режима отображения графа;
- применения алгоритмов раскраски графов и карт;
- самостоятельного закрашивания графов и карт с применением теории четырёх красок.

Нефункциональные требования

- симулятор должен соответствовать минимальным системным требованиям:
 - ОС: Windows 8.1, 10;
 - процессор: Intel Core 2-ядерный, аналогичный AMD или лучше;
 - оперативная память: 4 ГБ;
 - видеокарта: Nvidia GeForce GT 710, Intel HD Graphics 630 или лучше;
 - место на диске: не более 2 ГБ.
- интерфейс необходимо реализовать на русском языке.

Выбор среды разработки

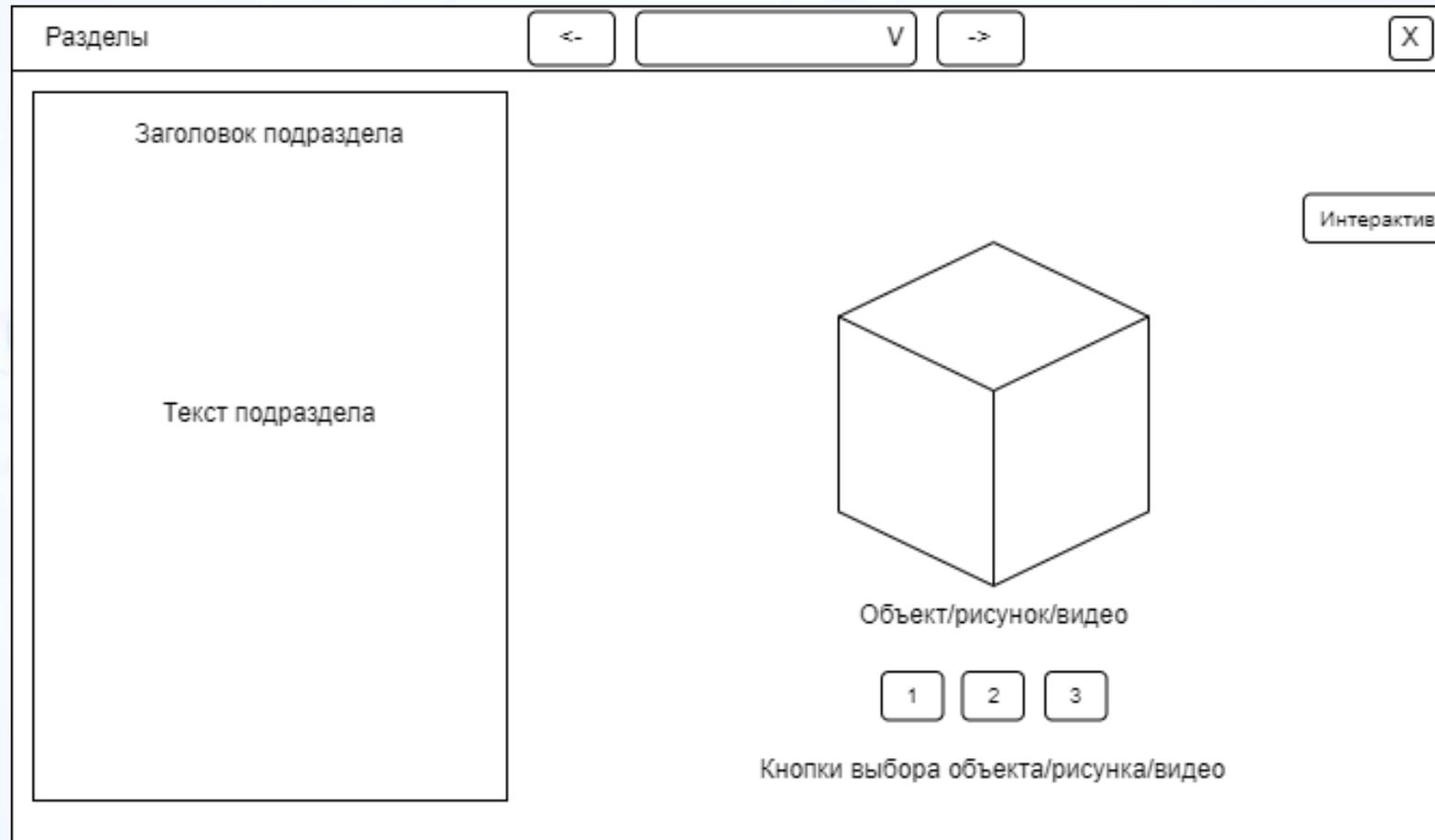
7

Среда разработки	Проработанная документация	Порог вхождения	Поддерживаемый язык	Широкий инструментарий
Unreal Engine 5	+	высокий	C++	+
Unity	+	низкий	C#	+
CryEngine	-	высокий	C++	+
Unigine	-	низкий	C++, C#	+ -



Проектирование интерфейса

8



Концепция теоретического раздела

Проектирование интерфейса

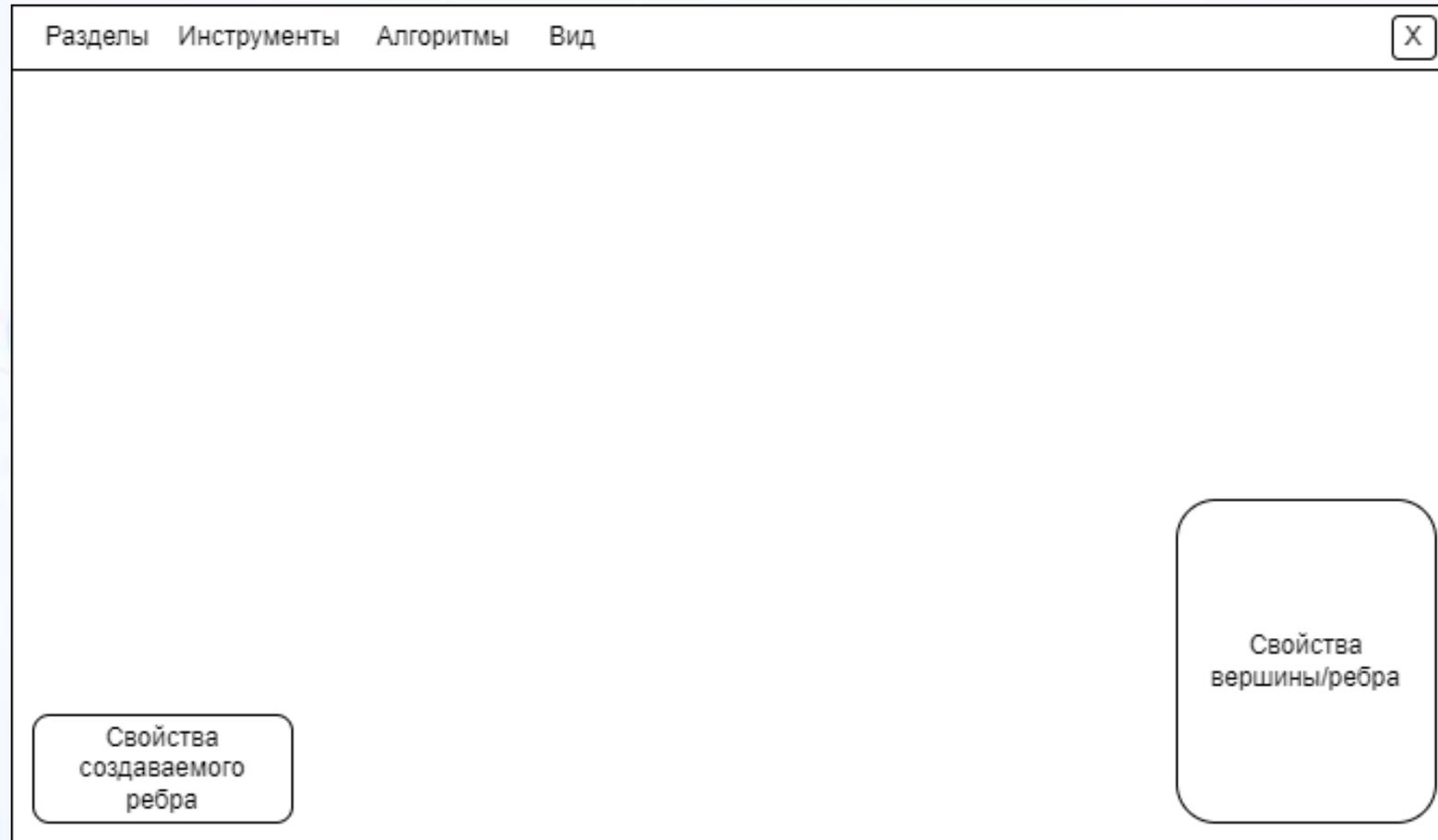
9



Концепция раздела «Построение пути»

Проектирование интерфейса

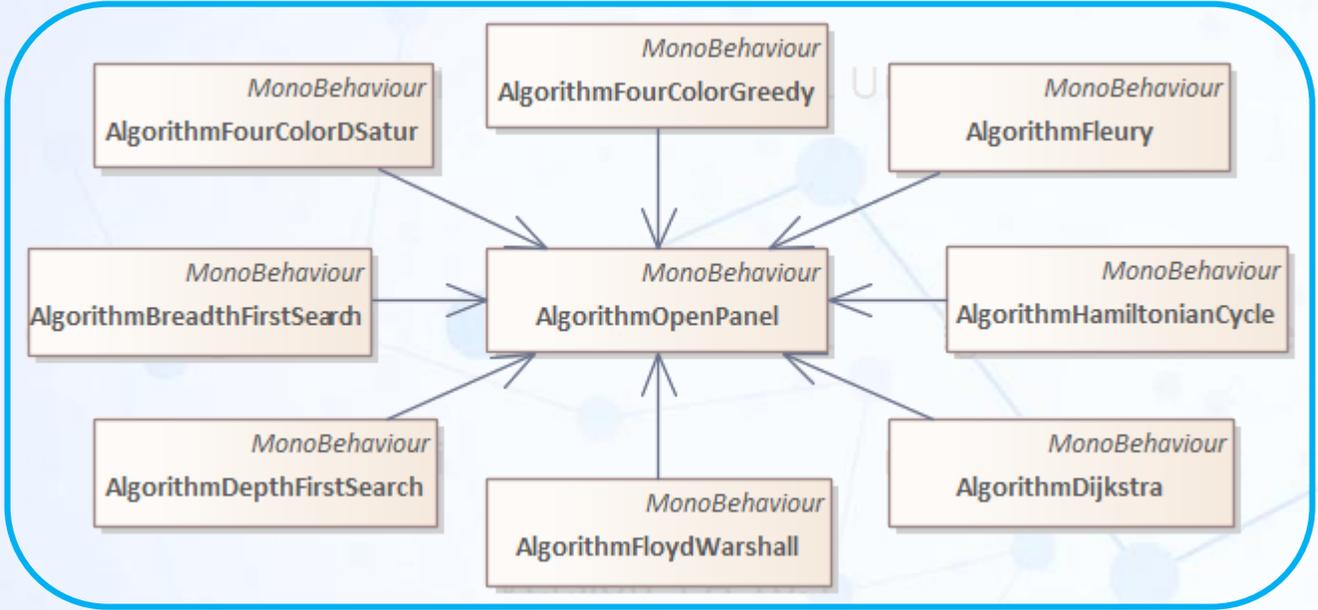
10



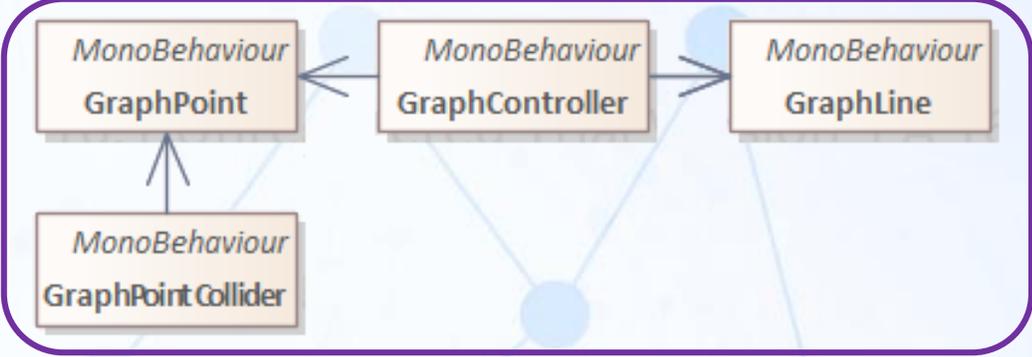
Концепция раздела «Редактор»

Диаграммы классов

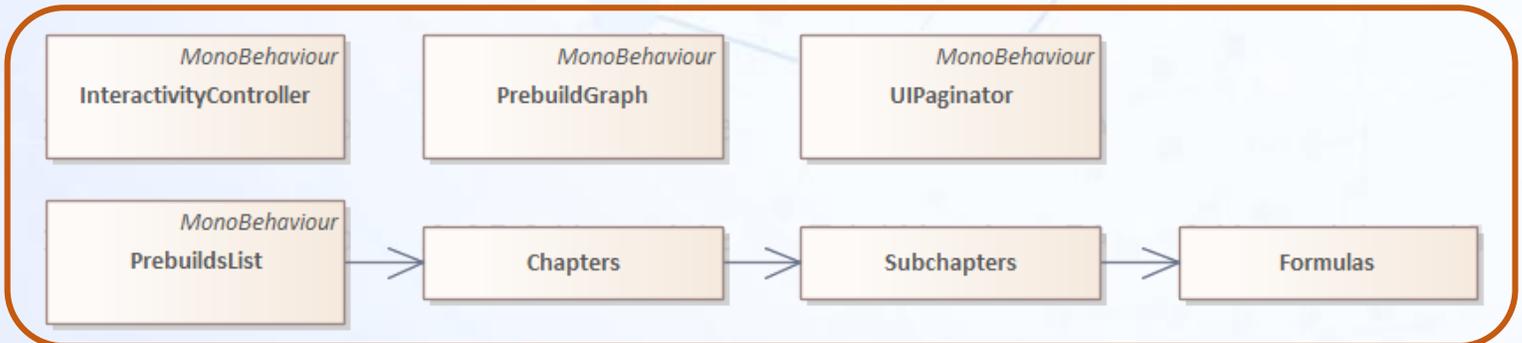
Алгоритмы



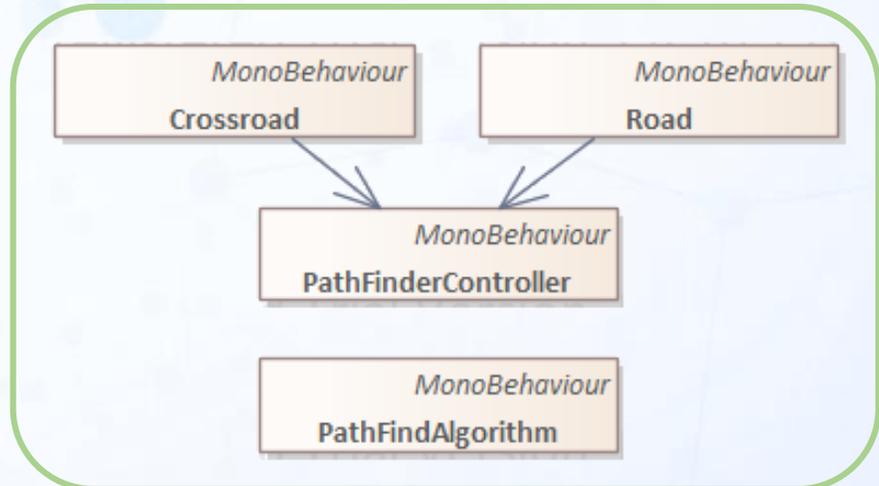
Создание и редактирование графа



Теоретический раздел

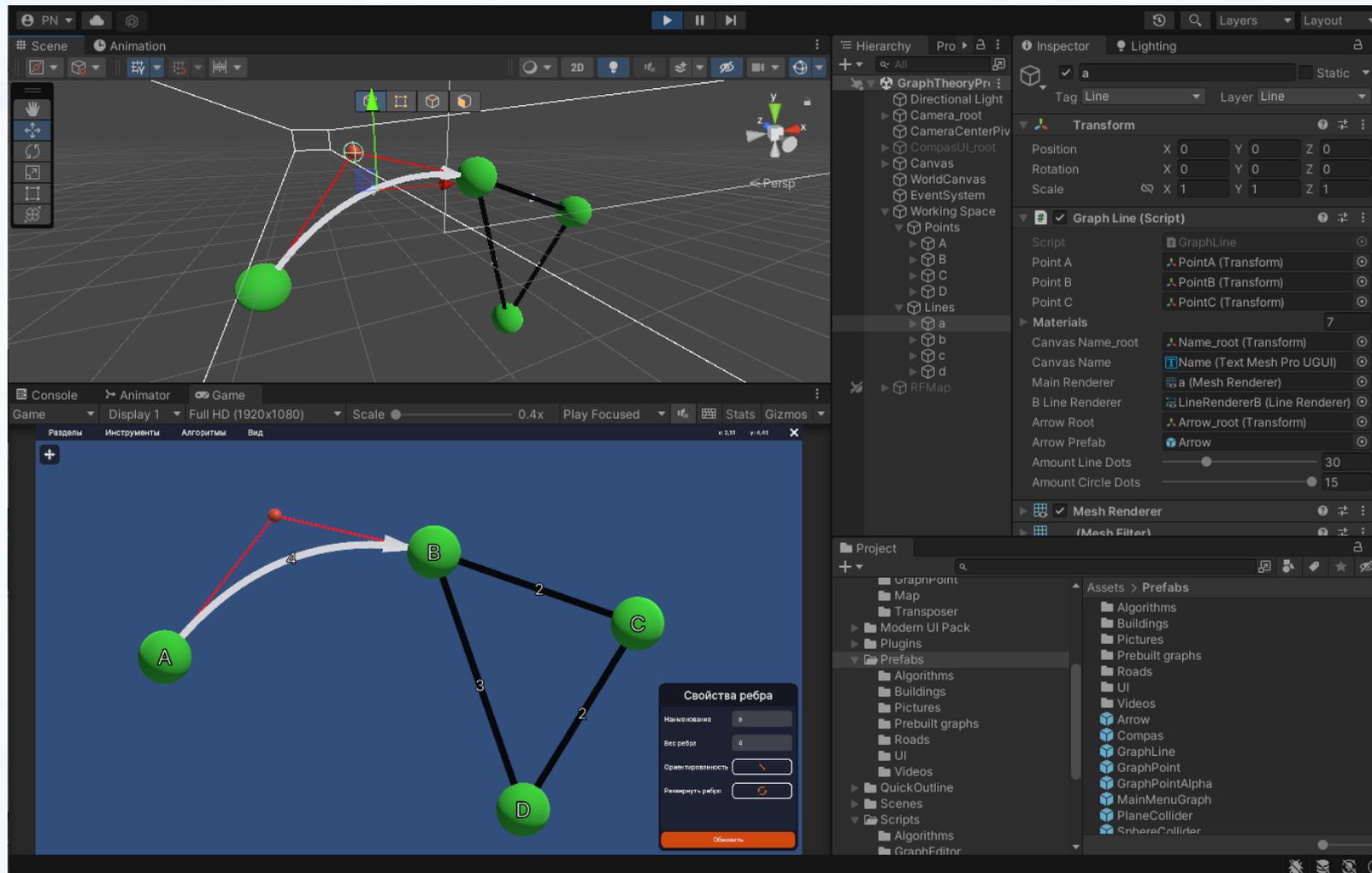


Построение пути



Работа с редактором Unity

12



Класс GraphLine

Реализация проекта

Разделы Инструменты Алгоритмы Вид x: -3,47 y: -1,07

А Алгоритм Флойда-Уоршелла

Панель действий алгоритма

Матрица расстояний

	A	B	C	D	E	F	G
A	0	7	Inf	8	11	12	Inf
B	7	0	6	15	12	19	Inf
C	Inf	6	0	19	Inf	Inf	Inf
D	8	15	19	0	19	4	Inf
E	11	12	Inf	19	0	42	Inf
F	12	Inf	Inf	4	42	0	Inf
G	3	11	3	16	Inf	Inf	0

Проверка условия: $11 + 8 \leq 19$

Матрица путей

	A	B	C	D	E	F	G
A	-1	A	-1	A	A	A	-1
B	-1	-1	B	A	B	A	-1
C	-1	C	-1	C	-1	-1	-1
D	-1	A	D	-1	A	D	-1
E	-1	E	-1	A	-1	E	-1
F	-1	-1	-1	F	F	-1	-1
G	G	G	G	G	-1	-1	-1

Журнал действий алгоритма

до общей точки A

Шаг 17: Заносим значение расстояния между точками D и A

Шаг 18: Проверяем выполнение условия. Точки D и C. *Условие не выполнено*

Шаг 19: Проверяем выполнение условия. Точки D и E. *Условие выполнено*

Шаг 20: Вычисляем расстояние между точками D и E, складывая расстояния до общей точки A

Шаг 21: Заносим значение расстояния между точками D и A

Шаг 22: Проверяем выполнение условия. Точки D и F. *Условие не выполнено*

Шаг 23: Проверяем выполнение условия. Точки D и G. *Условие не выполнено*

Шаг 24: Проверяем выполнение условия. Точки E и B. *Условие не выполнено*

Шаг 25: Проверяем выполнение условия. Точки E и C. *Условие не выполнено*

Шаг 26: Проверяем выполнение условия. Точки E и D. *Условие выполнено*

Шаг 27: Вычисляем расстояние между точками E и D, складывая расстояния до общей точки A

Шаг 28: Заносим значение расстояния между точками E и A

ПАНЕЛЬ УПРАВЛЕНИЯ АЛГОРИТМОМ

← ▶ ▷ ■

Пример работы раздела «Редактор»

Реализованные алгоритмы

1. Алгоритм поиска в ширину;
2. Алгоритм поиска в глубину;
3. Алгоритм Флойда-Уоршелла;
4. Алгоритм Дейкстры;
5. Алгоритм Флёрри;
6. Алгоритм поиска гамильтонова цикла;
7. Жадный алгоритм;
8. Алгоритм DSatur;

Реализация проекта

15



Пример работы раздела «Построение пути»

Заключение

1. Проведены обзор и анализ программного обеспечения для создания графов.
2. Проведён анализ и выбор средств реализации проекта.
3. Проведено архитектурное проектирование приложения.
4. Проведена программная реализация и тестирование проекта.

Предполагается использование проекта на кафедре ЭВМ.





Спасибо за внимание!