

Второй отборочный этап

Задания второго отборочного этапа разделяются на два тура: индивидуальный и командный

Индивидуальный этап:

- Блок Геймдизайн
- Блок теории программирования
- Блок программирования
- Блок моделирования

Командный этап:

- Блок дизайн-документ
- Блок моделирование жителей
- Блок программирование генерации
- Блок демонстрация геймплея

Блок Геймдизайн

Задание 1. Секрет успеха

Что нужно для создания одной из первых компьютерных игр, получившей широкое распространение по всему миру?

Новый мини-компьютер PDP-1, ученые Массачусетского технологического института и тригонометрическая функция, необходимая для расчёта траектории полёта. Собрав всё нужное воедино, образовавшаяся группа представила свою первую компьютерную игру:

Ответ: spacewar!

Баллы: 5

Задание 2. Сбор ресурсов

У игрока есть 3 ресурса: 40 - металл, 240 - камень и 200 - дерево. На эти ресурсы он строит домики в городе. Игрок может построить домики трёх типов: А, В, С (жилой коттедж, торговый центр, офисное здание).

Постройка домика приносит разное количество опыта игрока (EXP). Но разный тип домика требует разное кол-во ресурсов на постройку (см. таблицу). При этом под каждый тип дома есть ограниченное количество мест на карте. (см. таблицу, например: нельзя построить больше 8 домов типа А). Сколько максимум опыта может получить игрок, построив домики за те ресурсы, которые у него есть?

тип задания	кол-во мест	EXP	требуемые ресурсы (металл, камень, дерево)
А	8	50	5, 25, 20
В	6	100	15, 20, 45
С	4	75	10, 20, 20

Ответ: так как количество ресурсов у нас разное и ограниченное, то мы можем задать им ценность обратно пропорциональную их количеству для максимально эффективного использования.

Нужно найти наименьший общий знаменатель - $240 \cdot 5 = 1200$, и этот знаменатель разделить на количество ресурсов, чтобы определить их ценность, таким образом у нас получаются следующие ценности ресурсов металл = $1200/40=30$, камень = $1200/240=5$, дерево = $1200/200=6$

Теперь мы можем определить, какой домик будет построить эффективнее всего:

домик А = $5 \cdot 30 + 25 \cdot 5 + 20 \cdot 6 = 395$ на 5 exp, 1 exp = 79 ед. ценности

домик В = $15 \cdot 30 + 20 \cdot 5 + 45 \cdot 6 = 820$ на 15 exp, 1 exp = 54,(6) ед. ценности

домик С = $10 \cdot 30 + 20 \cdot 5 + 20 \cdot 6 = 520$ на 10 exp, 1 exp = 52 ед. ценности

Строим дом С - итого ресурсов: 30, 220, 180 и 75 exp;

И так далее продолжаем, пока есть возможность строить дома, учитывая лимит по их количеству.

В итоге максимальное возможное количество $EXP = 400$;

Баллы: 10

Задание 3. Теория вероятностей

В городе есть роддом. Каждый час в роддоме рождается ребёнок. Какова вероятность того, что следующие пять детей родятся одного пола при условии, что вероятность рождения мальчиков — 50%?

Ответ: $50\% = 1/2$

$(1/2)^5 = 1/32$ - мальчики

$(1/2)^5 = 1/32$ - девочки

$1/32 + 1/32 = 1/16$, получаем 0.0625;

Баллы: 5

Задание 4. Холодный расчёт

В городе живёт 10000 человек. На данный момент в городе один потенциальный преступник на 50 человек. В городе 100 полицейских. Зарплата полицейского 20000 рублей. При увеличении зарплаты на 1000 рублей работать в полицию приходит ещё один человек. Какую минимальную зарплату нужно предложить полиции, чтобы количество полицейских уравнило количество потенциальных преступников при условии, что потенциальный преступник тоже может пойти работать в полицию и преступником быть перестанет?

Ответ: 2% - преступники, 1% - полицейские;

$(2\% + 1\%)/2 = 1,5\%$ - необходимо ещё 0,5% человек в полицию - 50 человек не хватает, значит надбавка в $50 \cdot 1000$, $20000 + 50000 = 70000$.

Баллы: 10

Задание 5. Курьерская служба

В городе существует курьерская служба. Она доставляет заказы из ресторанов. Стоимость доставки 100 рублей, уже включена в стоимость заказа. Зарплата курьера составляет 63 рубля за доставку одного заказа. При потере заказа компания компенсирует клиенту полную стоимость заказа, но при этом не штрафует курьера. Сколько заработает компания в день, если курьеры выполнили 18300 заказов со средней стоимостью в 580 и потеряли 2% из них?

Ответ: 100% - 18300 заказов, 2% - утераны, 100 – доставки, 63 – оплата курьеру. $100 - 63 = 37$ – выгода с заказов.

Выгода: $18300 * 37 = 677100$

Убытки – 212280 (2% утераны – $18300 * 0.02 = 366$ заказов по 580 рублей).

Общая выгода: $677100 - 212280 = 464820$;

Баллы: 10

Задание 6. Два здания

У игрока есть два здания, непрерывно приносящие и копящие золото. На момент старта золота в зданиях нет. Здание 1 приносит 100 золота в секунду, здание 2 — 500 золота в секунду. Как только здание накапливает сумму кратную 1000, из здания изымается 1000 золота, и оно автоматически улучшается в течение 20 секунд, после чего начинает приносить на 100 золота в секунду больше. В процессе улучшения здание не копит золото вообще. Сколько золота будет суммарно в двух зданиях по прошествии трёх минут?

Ответ: составляем алгоритм, где счётчиком является одна секунда, а также имеем пару переменных количества текущего золота, сколько золота приносит данное здание, количество секунд улучшения. Если у здания 0 кол-во секунд улучшения, и кол-во золота не кратно 1000, то добавляем к кол-ву текущего золота количество, сколько золота приносит данное здание, если кратно 1000, то изымаем 1000 золота, а также секунды улучшения становятся равны 20, и, пока секунды улучшения здания не равны нулю, каждый шаг цикла отнимаем единицу у переменной секунд улучшения. Количество шагов в цикле равно 180 (3 минуты = 180 секундам).

Баллы: 10

Задание 7. UI

В какой из данных игр в основном используется диегетический интерфейс?

Ответ: Half-Life: Alyx.

Баллы: 10

Задание 8. Разнообразие жанров

В градостроительном игровом сегменте с момента его возникновения и до последнего десятилетия было сделано не так много игр, что изменилось с приходом инди. Это повлияло на то, что сейчас градостроительные симуляторы разбиваются на более большое количество направлений — поджанров, каждое из которых часто имеет достаточно отличную друг от друга аудиторию.

Всего можно выделить 5 основных поджанра:

1. Симулятор поселения

Вся экономика держится на жителях, которые являются одним из самых ценных ресурсов. Так, для дальнейшего прогресса и новых возможностей игроку требуется все больше жителей, что чаще всего появляются в зависимости от времени.

2. Реализм и современность

Отличительная особенность — реалистичная графика и упор на детальную экономику города, что включает в себя механики зонирования, работу с логистикой и удовлетворение потребностей населения города.

3. Районный поджанр

Основная механика подразумевает районирование, получающееся за счет правильного соединения дорог, относительно которых развивается каждый из районов.

4. Военные

Цель выжить и победить в данном поджанре стоит превыше механики строительства и управления городом, которая в то же время часто упрощена.

5. Сюжетные

В играх данного поджанра присутствует кампания с определенными миссиями и выдаваемым порционно сюжетом, что ставится в приоритете, из-за чего часть градостроительных механик нередко упрощается.

Распределите каждую из последующих игр в таблице по одному из поджанров.

Ответ:

Игры\поджанры	Реалистичная современность	Военный поджанр	Районный поджанр	Симулятор поселения	Сюжетный поджанр
They Are Billions	☐	☒	☐	☐	☐
RimWorld	☐	☐	☐	☒	☐
Frostpunk	☐	☐	☐	☐	☒
Banished	☐	☐	☐	☒	☐
Endzone - A World Apart	☐	☐	☐	☒	☐
SimCity	☒	☐	☐	☐	☐
Surviving Mars	☐	☐	☐	☒	☐
Emperor: Rise of the Middle Kingdom	☐	☐	☒	☐	☐
Northgard	☐	☒	☐	☐	☐

Баллы: 10

Задание 9. Анализ рынка

Молодая команда разработчиков энтузиастов планирует создать лучшую игру с военной техникой, поэтому хочет проанализировать рынок данной тематики: на какие платформы выпускаются игры-конкуренты?

Соотнесите игру с платформами, на которых она представлена.

Ответ:

Игры\Платформа	Windows	Android	IOS	Xbox	PlayStation	Web
World of Tanks	☒	☐	☐	☒	☒	☐
Tanki Online	☒	☒	☐	☐	☐	☒
War Robots	☒	☒	☒	☐	☐	☐
War Thunder	☒	☐	☐	☒	☒	☐
Armored Warfare	☒	☐	☐	☒	☒	☐
Crossout	☒	☐	☐	☒	☒	☐
World of Tank: Blitz	☒	☒	☒	☐	☐	☐

Баллы: 15**Задание 10. Нужно ли перезаряжаться?**

Игры вызывают различные эмоции и отклик у игроков, используя разнообразные механики, которые углубляют геймплей.

Например, одна из таких механик — перезарядка оружия. Однако, целесообразно ли использовать перезарядку оружия в любой экшн игре? Проанализируйте игру по описанию и выясните, какой вид перезарядки оружия использовали бы в игре.

Шутер от третьего лица, ключевой особенностью которого является способность героя замедлять время, что позволяет ему крайне быстро ориентироваться в пространстве во время боя, а также реагировать на всевозможные угрозы.

Шутер, события которого разворачиваются на Диком Западе. Игроков ждет невероятно проработанный огромный и живой мир, в котором NPC живут своей жизнью и реагируют на все действия игрока.

Это седьмая часть приключенческого экшена про ассасинов и тамплиеров, действие которого разворачивается во времена Великой Французской Революции. Главным героем этой части становится молодой человек, чья миссия — отомстить за убитого отца.

Игра в жанре Survival horror. Игровой процесс основан на исследовании территории, решении головоломок и схватках с врагами.

Необходимо указать, какой вид перезарядки (быстрый или медленный) подходит к описанию игры.

Ответ:

Номер игры	Долгая перезарядка	Быстрая перезарядка
1	☐	☒
2	☒	☐
3	☒	☐
4	☒	☐

Баллы: 15

Блок моделирования

Задача 1: Моделирование модулей зданий

Первый этап, подразумевает моделирование перечня модулей. Сами модули, представляют собой набор элементов, из которых можно собрать большое число разнообразных, составных объектов, например здания, строения, мосты, дороги, магистрали, и т. д. Такой подход к моделированию обусловлен также тем, что такие (составные, модульные) объекты не подгружаются целиком, а соответственно все, что выходит за пределы поля зрения игровой камеры, можно освободить от необходимости визуализации. А количество моделей в корневом каталоге игры будет более структурировано и гибко в доступе к отдельным файлам.

Многосоставная модель, состоящая из множества элементов. Выполнить можно в любом программном пакете для 3д моделирования (рекомендуется использовать программу Blender). Используя набор модулей, можно собрать большое разнообразие зданий.

Задание: смоделировать модули здания, представленные ниже в стилистике «LOW POLY MODELS» и в формате obj, а также предоставить

разные текстуры (различные текстуры для одного того же модуля дают вариативность составления зданий) в формате png на каждый модуль.

Критерии проверки: количество полигонов должно соответствовать сложности и типу модуля (оптимизировать и минимизировать количество полигонов, но без ущерба детализации), визуальное качество модели и соответствие стилистике «LOW POLY MODELS», полезное пространство текстуры, её разрешение и визуальное качество, а также различие текстур у разных моделей.

Перечень моделей, которые необходимо смоделировать:

1. **Стена сплошная - 3 текстуры** (название файла **wall_simple**)
2. **Стена роскошная сплошная - 2 текстуры** (название файла **wall_lux**)
3. **Стена с оконным проемом - 3 текстуры** (название файла **wall_simple_window**)
4. **Стена с оконным проемом для роскошных окон – 2 текстуры** (название файла **wall_lux_window**)
5. **Модуль крыши – 3 текстуры** (название файла **roof_simple**)
6. **Модуль роскошной крыши - 2 текстуры** (название файла **roof_lux**)
7. **Будка выхода на крышу – 3 текстуры** (название файла **roof_booth**)
8. **Окно бюджетного ценового сегмента – 1 текстура** (название файла **window_simple**)
9. **Окно среднего ценового сегмента – 1 текстура** (название файла **window_mid**)
10. **Окно роскошного ценового сегмента – 1 текстура** (Название файла **window_lux**)

Задача 2: Моделирование модулей дороги

1. **Однополосная** асфальтированная **прямая** дорожная единица длиной **5 метров, 3 текстуры** (название файла **road1**)
2. **Двухполосная** асфальтированная **прямая** дорожная единица длиной **5 метров 3 текстуры** (название файла **road2**)
3. **Трехполосная** асфальтированная **прямая** дорожная единица длиной **5 метров 3 текстуры** (название файла **road3**)
4. **Однополосный** асфальтированный **перекресток** четырех дорог **3 текстуры** (название файла **crossroad1**)
5. **Двухполосный** асфальтированный **перекресток** четырех дорог **3 текстуры** (название файла **crossroad2**)
6. **Трехполосный** асфальтированный **перекресток** четырех дорог **3 текстуры** (название файла **crossroad3**)
7. **Однополосный** асфальтированный **перекресток** трех дорог **3 текстуры** (название файла **threeroad1**)
8. **Двухполосный** асфальтированный **перекресток** трех дорог **3 текстуры** (название файла **threeroad2**)
9. **Трехполосный** асфальтированный **перекресток** трех дорог **3 текстуры** (название файла **threeroad3**)
10. Модель дорожного знака **1 текстура** (название файла **sign**)
11. Набор пропс-моделей фонарные столбы, линии коммуникаций, трубы, виадуки и т. д. **1 текстура** (название файла **props**)

моделька	соответствует	количеству	треугольников	<u>+2 балла</u>
если	не	прошла	первую, то	<u>+1 балл</u>
моделька	прошла	соответствует	количеству	треугольников
погрешность на 75%				
текстура	проверена на	соотношение	сторон 1к1, разрешение не	<u>+1 балл</u>
более 2048 пикселей, степень разрешения равна 2				

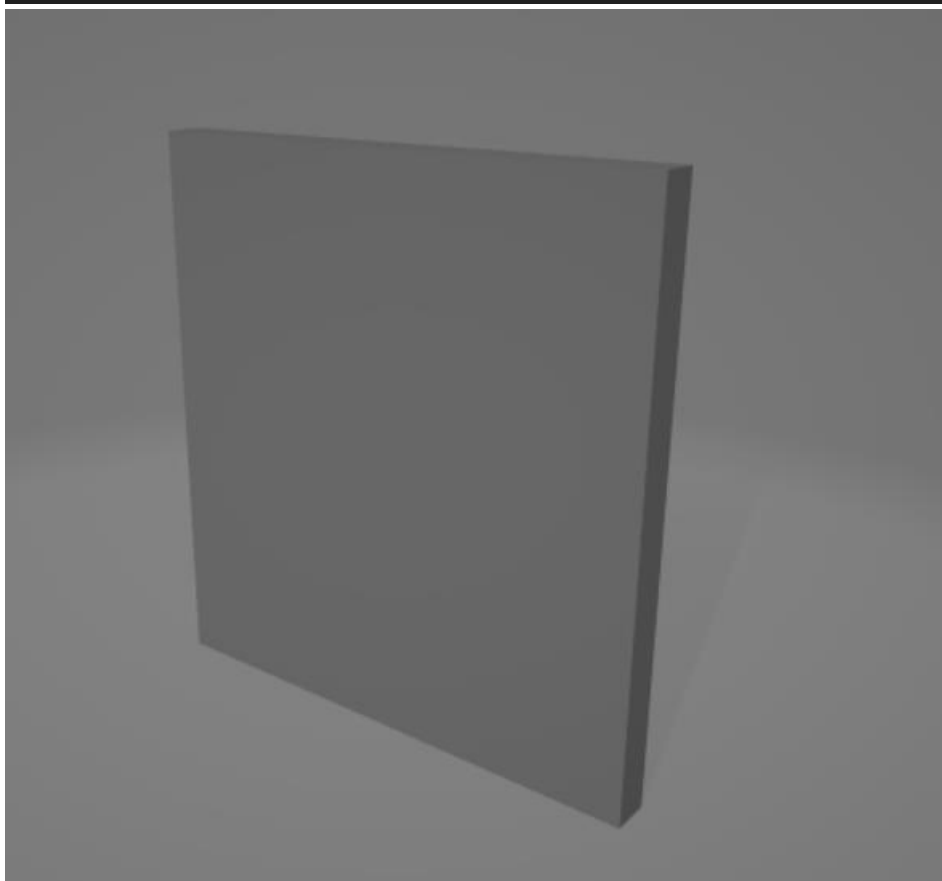
если прошла проверку выше, то	<u>+1 балл</u>
текстура проверена на наполненность более 65%	

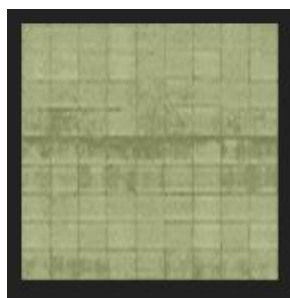
Если текстур несколько, например три, берётся наименьший балл из всех текстур, то есть если у двух текстур всё ОК и по 2 балла, но у третьей текстуры 0 баллов, то за текстуры в этом под-задании выходит 0 баллов

Решение задачи 1:

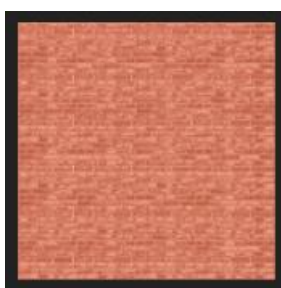
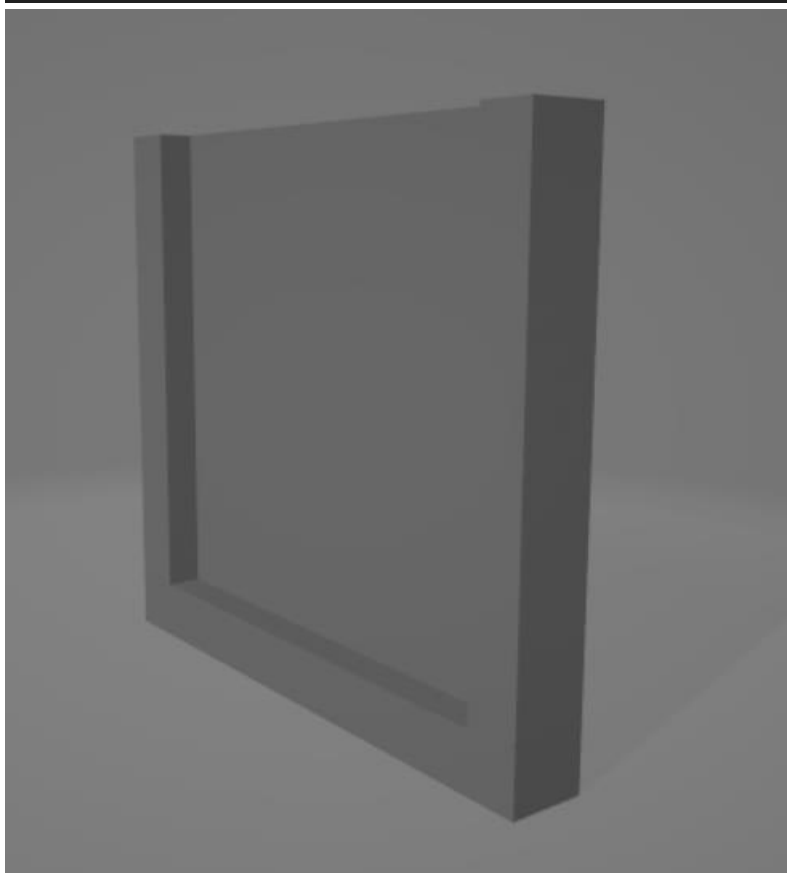
1	Папка с файлами		
2	Папка с файлами		
3	Папка с файлами		
4	Папка с файлами		
5	Папка с файлами		
6	Папка с файлами		
7	Папка с файлами		
8	Папка с файлами		
9	Папка с файлами		
10	Папка с файлами		

wall_simple_tex1.png	Файл "PNG"	27 КБ	Нет
wall_simple_tex2.png	Файл "PNG"	22 КБ	Нет
wall_simple_tex3.png	Файл "PNG"	27 КБ	Нет
wall_simplej.obj	3D Object	1 КБ	Нет

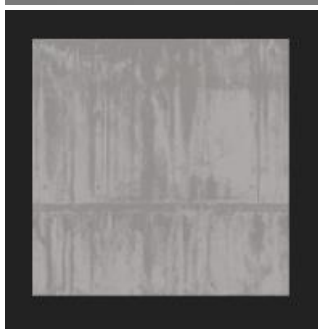
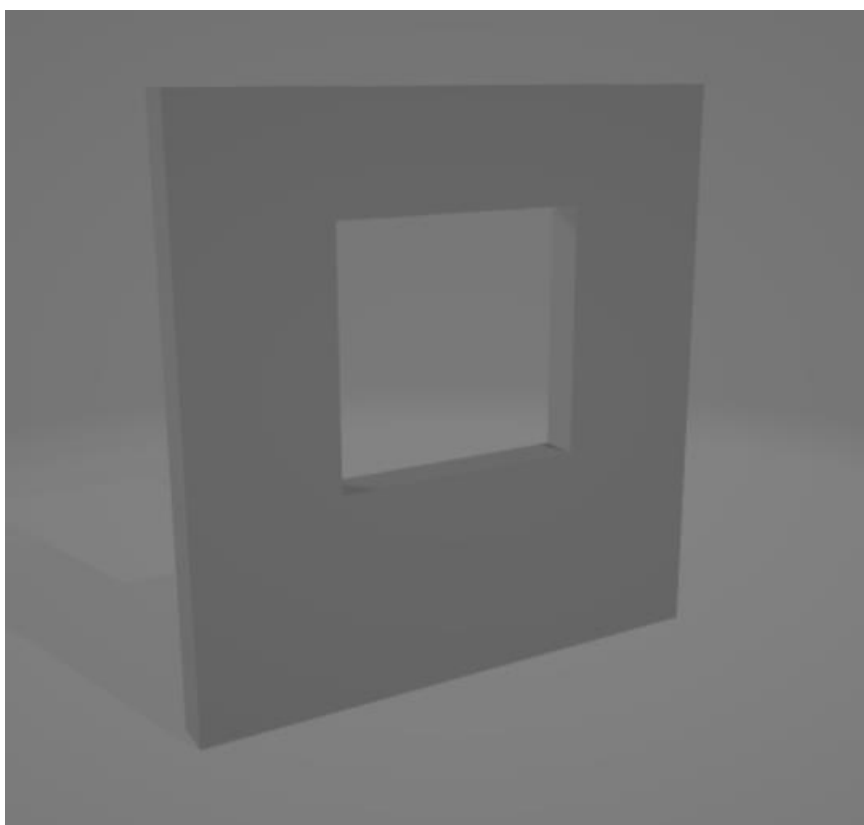




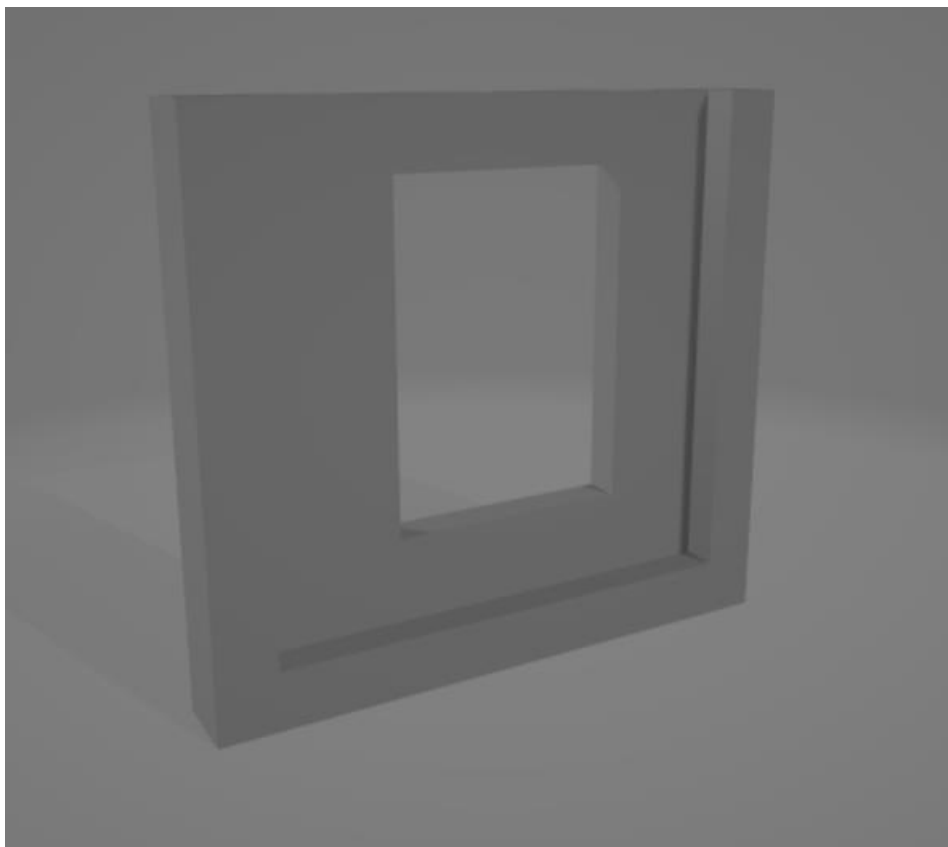
 wall_lux.obj	3D Object	1 КБ	Нет
 wall_lux_tex1.png	Файл "PNG"	30 КБ	Нет
 wall_lux_tex2.png	Файл "PNG"	27 КБ	Нет



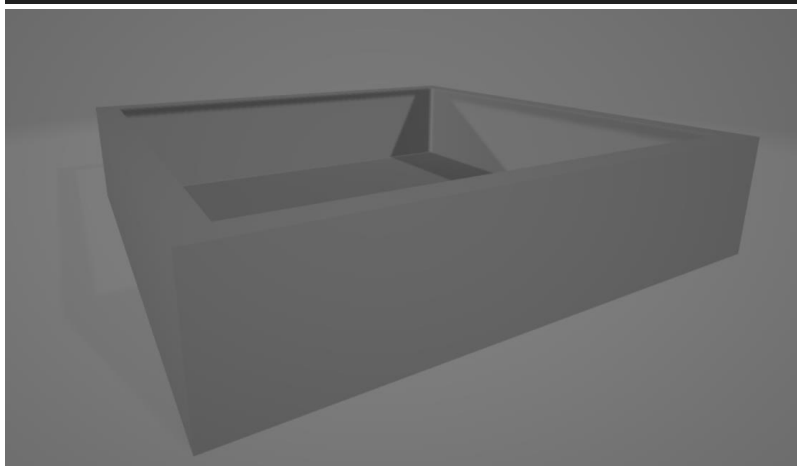
 wall_simple_wind_tex1.png	Файл "PNG"	24 КБ	Нет
 wall_simple_wind_tex2.png	Файл "PNG"	26 КБ	Нет
 wall_simple_wind_tex3.png	Файл "PNG"	26 КБ	Нет
 wall_simple_window.obj	3D Object	2 КБ	Нет

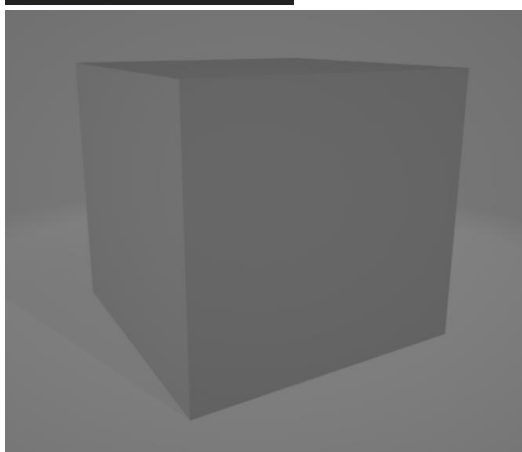
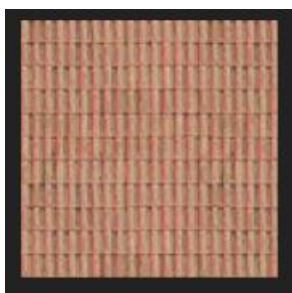
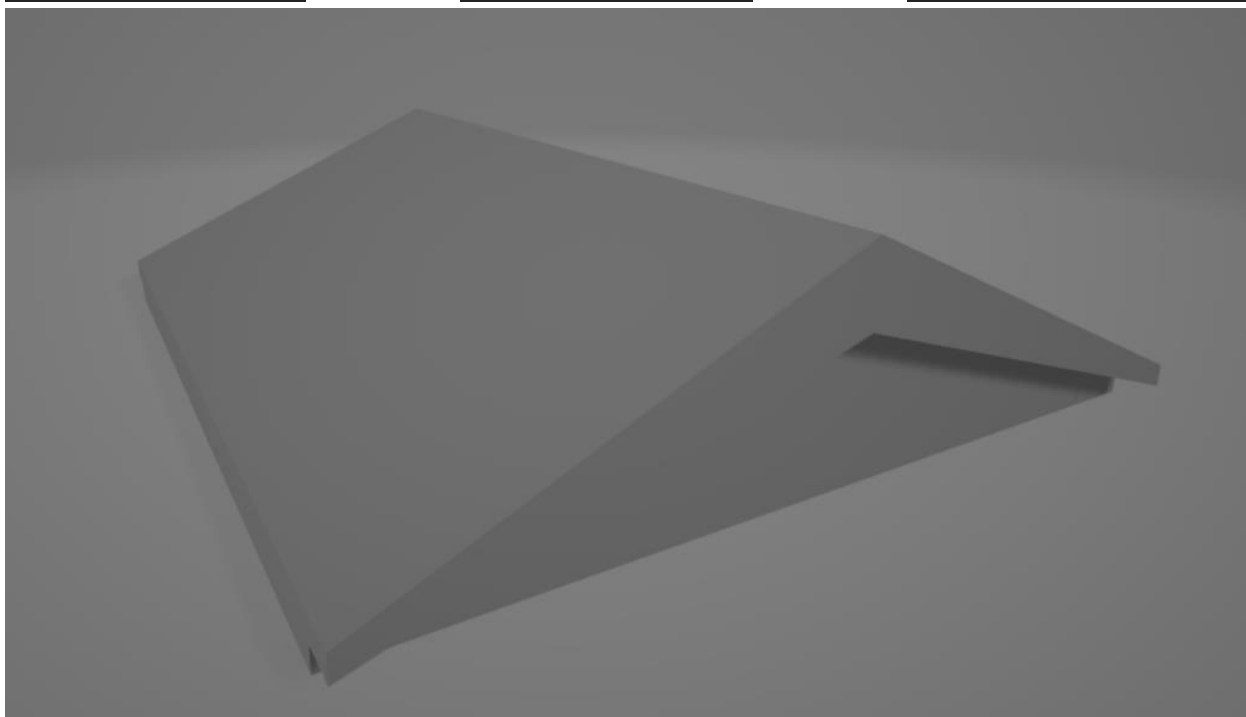
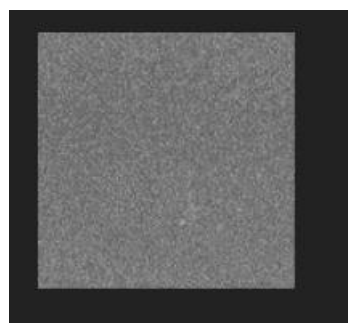
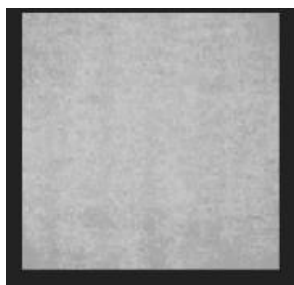
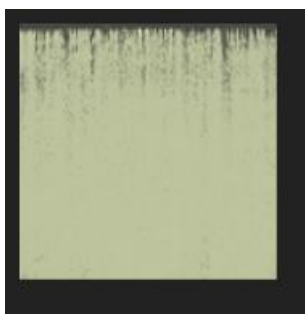


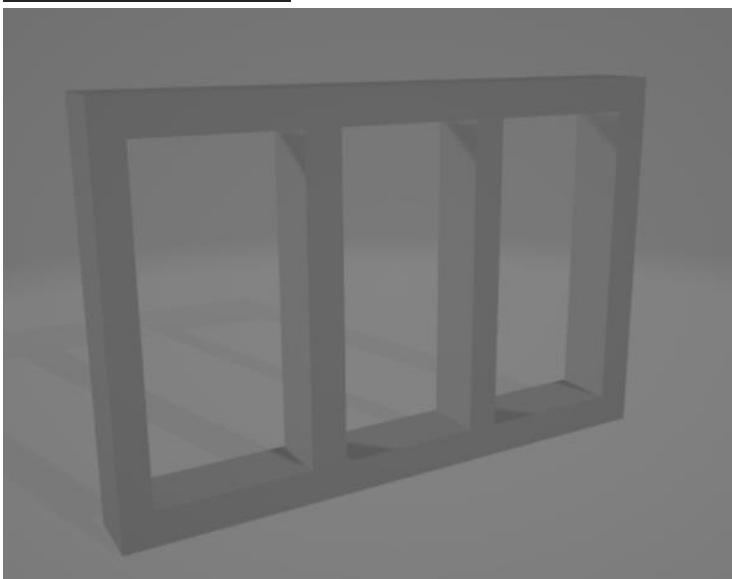
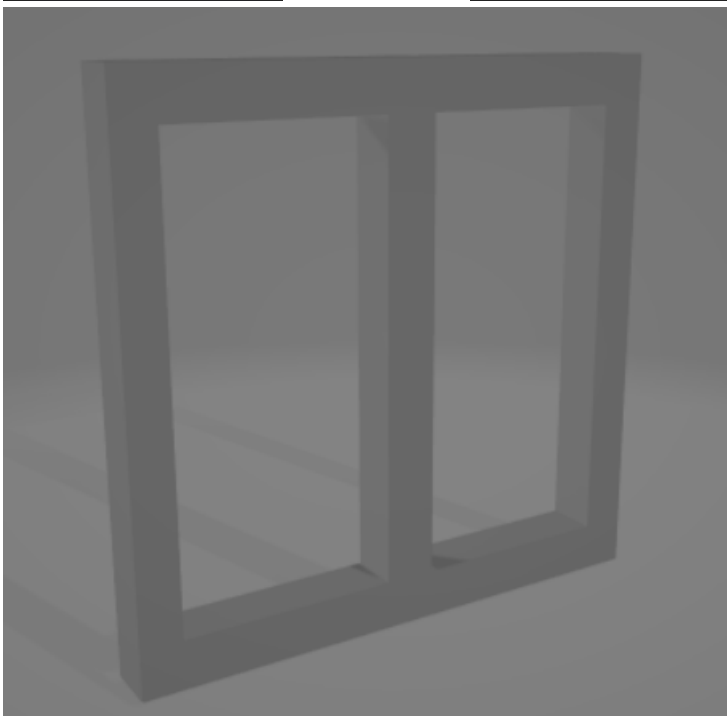
 wall_lux_window.obj	3D Object	2 КБ	Нет
 wall_lux_window_tex1-min.png	Файл "PNG"	26 КБ	Нет
 wall_lux_window_tex2.png	Файл "PNG"	27 КБ	Нет

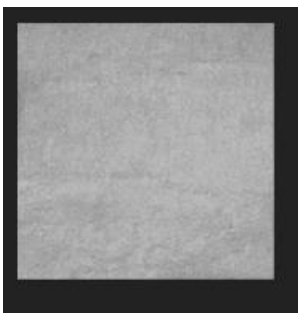


 roof_simple.obj	3D Object	1 КБ
 roof_simple_tex1.png	Файл "PNG"	22 КБ
 roof_simple_tex2.png	Файл "PNG"	15 КБ
 roof_simple_tex3.png	Файл "PNG"	22 КБ

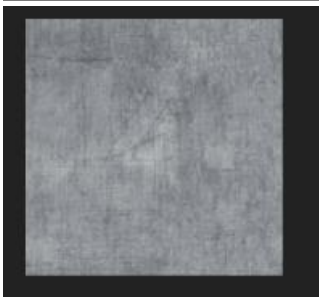
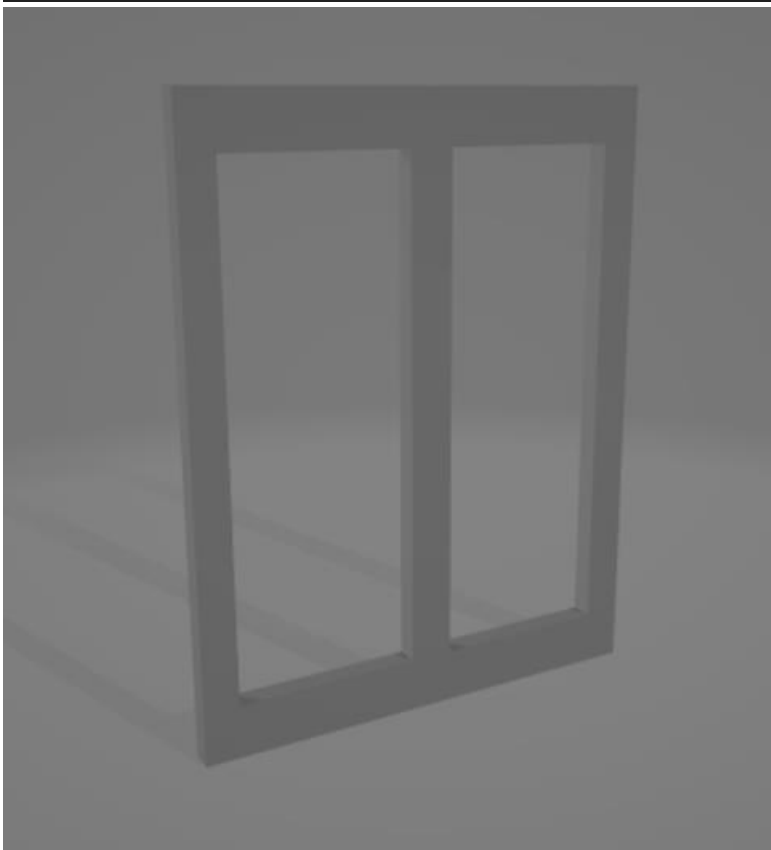




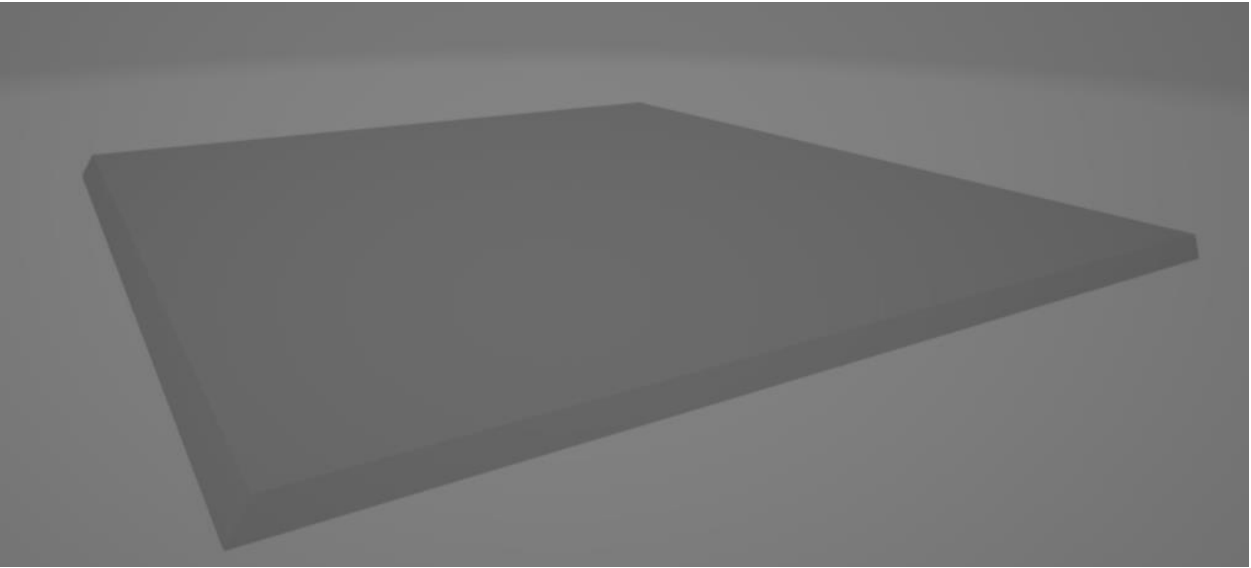




 window_lux.obj	3D Object	2 КБ
 window_lux_tex1.png	Файл "PNG"	36 КБ



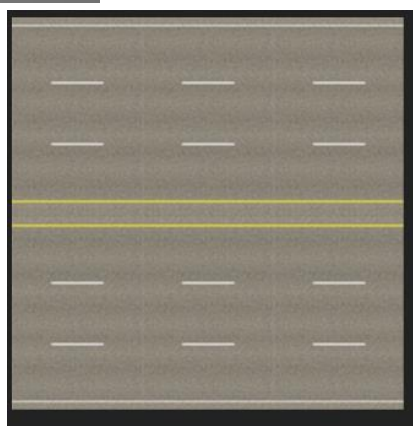
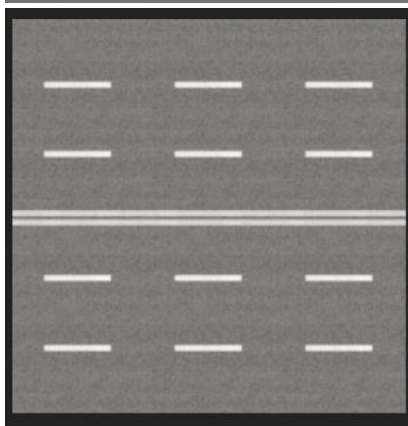
1	Папка с файлами
2	Папка с файлами
3	Папка с файлами
4	Папка с файлами
5	Папка с файлами
6	Папка с файлами
7	Папка с файлами
8	Папка с файлами
9	Папка с файлами
10	Папка с файлами
11	Папка с файлами



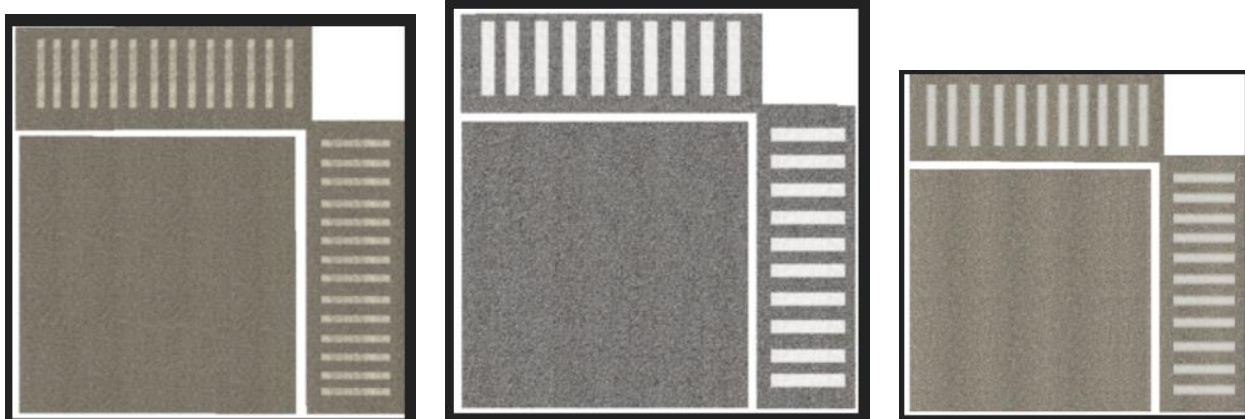
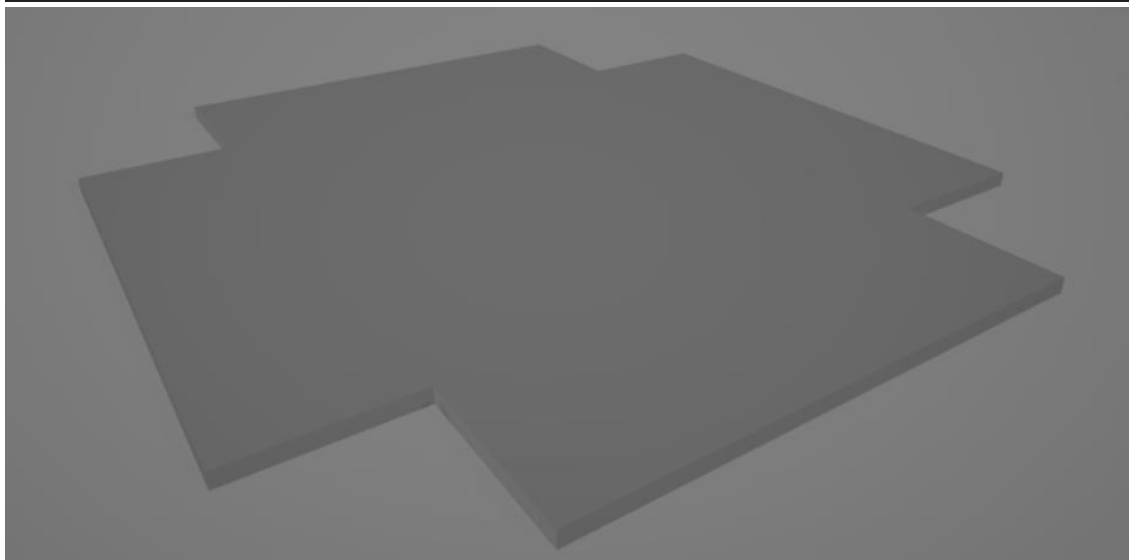
 road2.obj	3D Object	1 КБ
 road2_tex1.png	Файл "PNG"	24 КБ
 road2_tex2.png	Файл "PNG"	29 КБ
 road2_tex3.png	Файл "PNG"	24 КБ



 road3.obj	3D Object	1 КБ	Нет
 road3_tex1.png	Файл "PNG"	59 КБ	Нет
 road3_tex2.png	Файл "PNG"	62 КБ	Нет
 road3_tex3.png	Файл "PNG"	82 КБ	Нет



	crossroad1.obj	3D Object	1 КБ	Нет
	crossroad1_tex1.png	Файл "PNG"	116 КБ	Нет
	crossroad1_tex2.png	Файл "PNG"	154 КБ	Нет
	crossroad1_tex3.png	Файл "PNG"	123 КБ	Нет



	1	Папка с файлами	
	2	Папка с файлами	
	3	Папка с файлами	
	4	Папка с файлами	
	build_simple.obj	3D Object	91 КБ

. Баллы: 84 баллов.

Задача 3: Сборка зданий из модулей

1. Собрать здание бизнес-класса **бюджетного** ценового сегмента. Малоэтажные строения, с непрезентабельной, рутинной и не дорогой

архитектурой. Под такое описание, лучше всего подходят так называемые трущобы и серые «многоквартирники» (название файла **build_simple**)

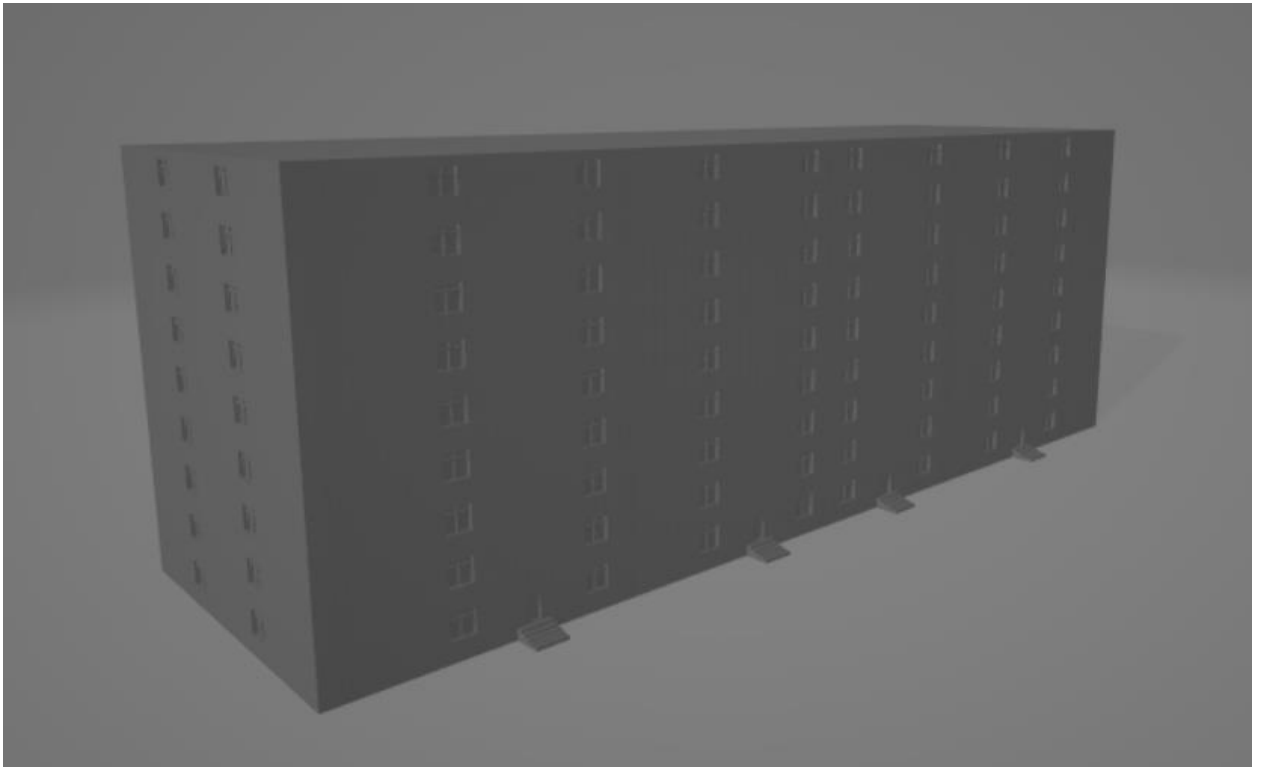
2. Собрать здание бизнес-класса **среднего** ценового сегмента. В большинстве своем, это здания каменные 5-8 этажей. Такие здания обычно имеют историческое значение, то есть должны выглядеть соответственно. (название файла **build_mid**)

3. Собрать жилое здание **роскошного** ценового сегмента. Красивые многоквартирные дома. (название файла **build_lux**)

4. Собрать **уникальный** по своему строению дом (можно из одного модуля). Например, «Сталинский высотки», «Пентагон», «Лувр», «Останкинская башня» и т. д. (название файла **build_unique**)

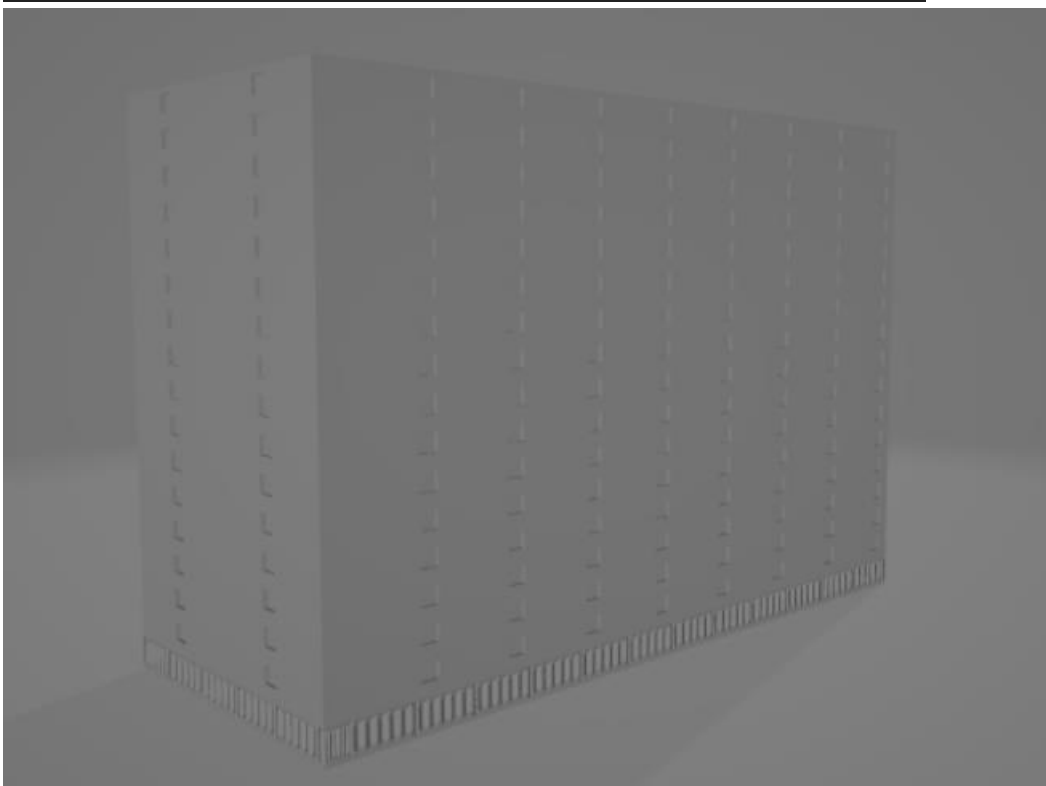
Решение задачи 3:

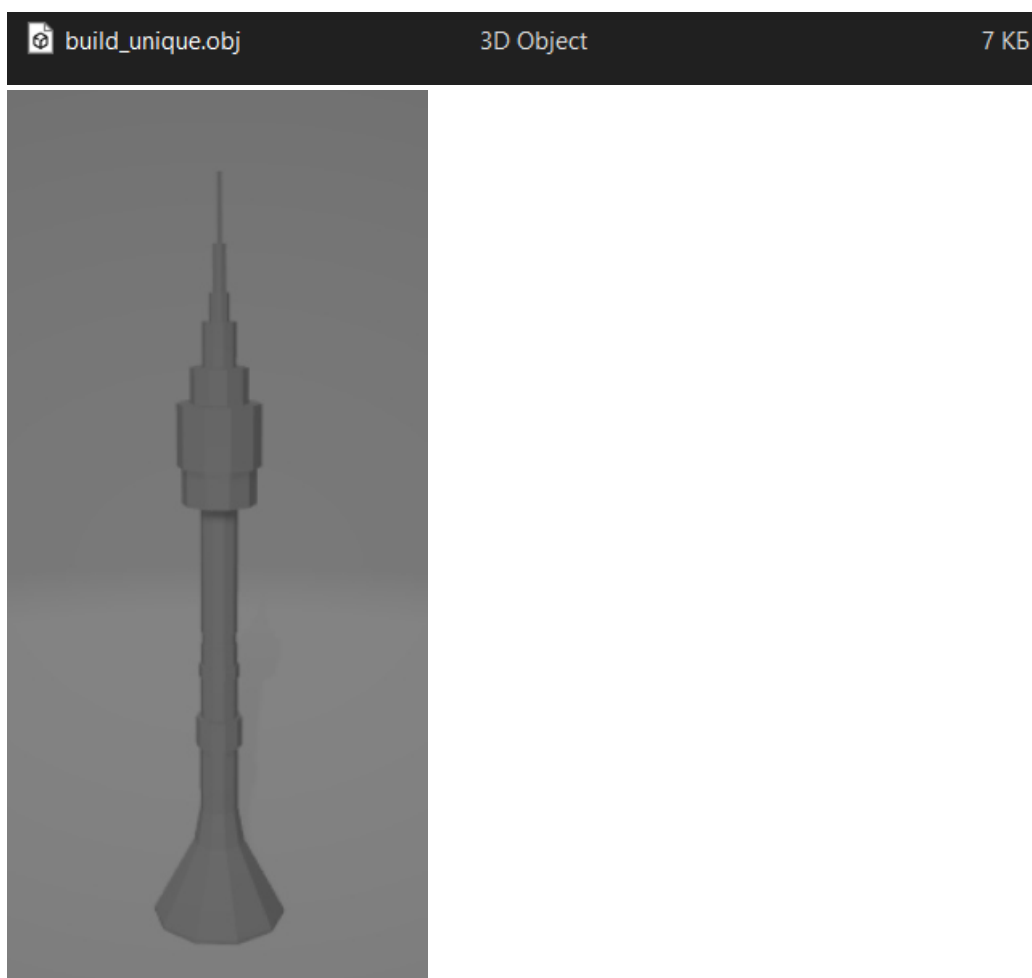




 build_lux.obj

3D Object





Каждый подпункт задачи проверяется:

моделька соответствует количеству треугольников	<u>+4 балла</u>
если не прошла первую, то моделька соответствует количеству треугольников с погрешностью на 75%	<u>+2 балл</u>

Баллы: 16 баллов.

Итого баллов за два задания: 100 баллов.

Блок моделирования жителей

Задание 1: Моделирование людей

Необходимо создать низкополигональные модели людей:

Модель-болванка взрослого человека мужского пола

Модель-болванка взрослого человека женского пола в юбке

Модель-болванка взрослого человека женского пола в брюках

Модель-болванка ребенка мужского пола

Модель-болванка ребенка женского пола в юбке

Модель-болванка ребенка женского пола в брюках

Модель-болванка ребенка мужского пола с рюкзаком

Модель-болванка ребенка женского пола в юбке с рюкзаком

Модель-болванка ребенка женского пола в брюках с рюкзаком

Вы можете сделать какого-либо другого жителя, который спишется в ваш визуальный стиль и сеттинг игры.

Задание 2: Разработка скелета

Необходимо разработать универсальный скелет и именованными костями, при этом к именам костей необходимо в отдельном текстовом файле составить пояснительную записку, что любой человек смог в течение некоторого, непродолжительного изучения смог разобраться в системе костей.

Задание 3: Рисование веса

Под рисованием веса всегда подразумевается определение мягких и жестких участков сетки геометрии на 3д объектах. Выполненной задачей считается, когда веса на всех моделях определены.

Задание 4: Библиотека анимации

Для скелета необходимо выполнить библиотеку анимации, состоящую из следующих пунктов:

Idle – анимация ожидания. Обычно под этой анимацией подразумевается дыхание стоя на месте

Walk – анимация ходьбы

Running – анимация бега

Также можно расширить библиотеку собственными вариантами

Задание 5: Библиотека материалов

Для каждой сетки геометрии необходимо разработать по меньшей мере 3-5 различных текстур для канала Defuse (Color или Albedo), при этом текстуры для остальных каналов (Normal, AO, Roughness, Specular, Metallic) могут быть одинаковыми.

Необходимо загрузить модель в формате fbx со всеми текстурными картами и материалами, скелетом (в том числе и записку) и анимациями для одного любого жителя и скриншот получившейся модели.

Критерии оценки

Сетка, геометрия	баллы
В моделях отсутствуют, или сведены к минимуму N-гоны	5
В моделях отсутствуют бессмысленные скопления вершин	5
В моделях отсутствуют, или сведены к минимуму вершины из которых исходят не четыре ребра	5
Модели внутри максимально пустые	5
Текстуры, развертка	
В развертке отсутствуют наложения, либо они контролируются автором	5
Развертка имеет островную топологию	5
Разрешение текстур соответствует правилу кратности двойки в степени	5
Соотношения сторон в текстурах одинаковые	5
Разрешение текстур не превышает показателя в 2048 пикселей	5
Скелет, веса геометрии	
Карта весов соответствует скелету	5

Реализована инверсивная кинематика	5
Каждая кость осмысленно именована	5
Реализован шейный позвонок, кисти рук	5
Уточнение весов	5
Анимация	
Реализована одна анимация ходьбы	5
Реализованы две-пять анимация ходьбы	5
Реализована как минимум одна анимация Idle	5
Реализована библиотека анимаций Idle	5
Реализованы анимации бега	5
Общая библиотека анимаций состоит более чем из 3 различных группированных дорожек	5

Решение

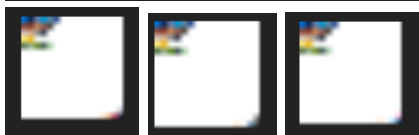
:

 Анимации	Папка с файлами
 Материалы	Папка с файлами
 Модель со скелетом	Папка с файлами
 Скриншот модели	Папка с файлами
 Man.blend	Blender File

 man@Idle.fbx	3D Object
 man@Running.fbx	3D Object
 man@Walking.fbx	3D Object
 man@Waving.fbx	3D Object



 man_texture.blend	Blender File
 texture1.png	Файл "PNG"
 texture2.png	Файл "PNG"
 texture3.png	Файл "PNG"





Head – голова

Neck – шея

Collar – ключица

Upper arm – плечо

Forearm – предплечье

Palm – ладонь

F1a – фаланга, где 1 – номер пальца, а – первая фаланга (b – вторая фаланга, c – третья)

Chest – грудная клетка

Stomach – область живота

Pelvis – таз

Thigh – бедро

Lower leg – голень

Foot – стопа

Toe – пальцы ног

Knee – управление поворотом ног

Base – управление корпусом

Root – перемещение скелета в пространстве

Ik-leg – управление ногами

Баллы: 100 баллов.

Блок программирования

Задание 1: Бюджет

Спустя 20 лет правления, Грудин Сергей решил захватить своего самого могущественного врага — Альгеру. После войны большая часть транспортной инфраструктуры страны была разрушена. Встал вопрос об её восстановлении.

Из-за сложностей в строительстве дорог и ограниченности бюджета, Грудин поручил в течение двух недель разработать проект соединения оставшихся целыми региональных дорожных сетей. НИИ Дорог и Конструкций начали разрабатывать проект, но столкнулись с тем, что региональных сетей автодорог очень много, поэтому вручную создать оптимальный план по строительству дорог практически невозможно. В НИИ создали рабочую группу, куда наняли вас в качестве программиста.

В НИИ уже подготовили список городов и список дорог между ними. Вам предстоит создать программу, которая рассчитает оптимальный по стоимости проект строительства дорог.

Входные данные

Вам дается 5 файлов, в каждом из которых находится информация о городах и дорогах в следующем формате:

В начале файла идет описание каждого города на отдельной строке в формате ШИРОТА ДОЛГОТА НАЗВАНИЕ ГОРОДА.

Дальше идет разделитель в виде трех минусов (---).

После этого на каждой новой строке в формате ГОРОД1 --- ГОРОД2 следуют города, между которыми уже существуют дороги.

Ссылка на скачивание — https://yadi.sk/d/0IQO_XpXCHBqZg?w=1

Выходные данные

Необходимо найти минимальную сумму длин дорог, которые нужно построить. Сумма длин дорог является целым числом, округляемым в большую сторону.

В качестве ответа необходимо через пробел ввести все длины дорог.

Учтите, что Венис имеет те же размеры, что и Земля ($R=6372,8\text{km}$). Для поиска расстояния используется гаверсинус. Считается, что дороги не пересекаются и движение по ним может осуществляться в обе стороны.

Ответ:

Задача подразумевает обработку набора данных в оффлайн-режиме, то есть временных ограничений для решения нет. Задача может решаться несколькими способами, но рассмотрим авторский вариант.

- В качестве основы для решения используется алгоритм минимального оставного дерева. Полный вариант решения:
 - Считываем данные
 - Строим исходный граф (в качестве веса ребра используется расстояние между
 - Обнуляем веса у существующих ребер
 - Достаиваем граф до полносвязного, в качестве весов ребер используем расстояние между городами, считаемое по формуле гаверсинуса (haversine)
 - Запускаем алгоритм минимального оставного дерева
 - Для получившегося минимального оставного дерева смотрим ненулевые ребра и выписываем их в результирующий файл

Баллы: 25

Авторский генератор и решатель

ВСТАВИТЬ

КОД

<https://gist.github.com/Mikhail57/e28fb521aeada07d500d567faa8073f8>

Задание 2. Навигатор

Инновационный стартап Ситирайдер решил представить рынку новую, доселе невиданную систему оптимизации доставки грузов. Презентации показаны, питчинг-сессии проведены, деньги инвесторов собраны, но встал вопрос: как найти оптимальный маршрут хотя бы для одного грузовика? Вам предстоит разработать алгоритм по определению оптимального маршрута грузовика, который должен объехать несколько точек.

Входные данные

В первой строке

N — количество точек, $1 \leq N \leq 1000$

Во второй строке через пробелы перечислены номера точек, которые необходимо объехать в любом порядке

В третьей строке

M — количество ребер между точками, $1 \leq M \leq 2000$

В пятой и последующих строках на каждой расположены данные по расстоянию между точками в формате **A B C**, где **C** — расстояние между точками **A** и **B**.

Граф направленный, т.е. путь из **A** в **B** может занимать больше, чем путь из **B** в **A**

Выходные данные

В первой строке число — сколько километров проедет грузовик.

Во второй и последующих строках — маршрут грузовика от точки **0** до конца маршрута.

Считается, что грузовик стартует из точки **0** и должен вернуться в неё же. Если выполнить маршрут невозможно, необходимо вывести **NO WAY**.

Авторское решение считает субоптимальный маршрут. Засчитываются маршруты короче или равные авторскому решению.

Ответ

Как было сказано в конце задания, авторское решение считает субоптимальный маршрут. Важные замечания: грузовик должен выехать и приехать обратно в точку, ребра в графе ориентированные. Авторский алгоритм решения:

- Считываем данные из потока ввода
- Строим граф
- Строим N случайных путей
- Выбираем самый короткий
- Выводим найденный путь

Баллы: 25

Авторский генератор и решатель

ВСТАВИТЬ

КОД

<https://gist.github.com/Mikhail57/39beb6f7406416a6d83826051e20ec1d>

Задание 3. Сбор сталкеров

В Кировлевске после войны образовалось множество заброшенных мест, откуда уехали все люди. Всекоролевское сообщество сталкеров решило собраться и обследовать 3 самых новых заброшенных города: Икутск, Петсбург и Санизападск. Они решили устроить своеобразный «марафон» по обследованию: каждый город должны обследовать I , P и V людей соответственно. Всего планирует принять участие в исследований $S = I + P + V$ человек, $30 \leq S \leq 300$. Все они расположены в разных городах, в одном городе может быть несколько участников. Организаторы оплачивают дорогу до места исследования, и они хотят сократить свои расходы.

Вам предстоит посчитать, сколько всего средств понадобится на организацию этого мероприятия (на дорогу туда и обратно для всех участников).

Входные данные

В первой строке через пробел дается необходимое количество сталкеров для каждого города (для Икутска **I**, Петсбурга **P** и Санизападск **V**)

Во второй строчке общее количество городов (**T**) в Кировлевске,
 $10 \leq T \leq 2000$

В третьей строчке через пробелы ID Икутска, Петсбурга и Санизападска

В четвертой строке написаны ID городов всех участников марафона (через пробел)

После этого строка ---

После этого в последующих ($50 \leq R \leq 80000$) строчках варианты добраться из города **A** в **B** в формате **ID_ГОРОДА_A ID_ГОРОДА_B СТОИМОСТЬ_ПРОЕЗДА**

Выходные данные

В качестве выхода одно число — минимальная стоимость организации мероприятия

Граф направленный. Возможен транзит, так как не всегда существуют прямые пути. Выбирается самый дешевый маршрут.

Ответ

Авторский алгоритм решения:

- Считываем входные данные
- Считаем для каждого города количество людей, которые поедут оттуда
- Для каждого исходного города считаем самый дешевый путь (Алгоритм Дейкстры) до каждого принимающего города
- Создаем новый граф для алгоритма Min Cost Flow:
 - Создаем исток, сток, исходные города и города назначения
 - Создаем ребра между истоком и городами, откуда отправляются люди, задавая им нулевой вес

- Создаем таблицу, обозначающую стоимость маршрута из точки А в точку В для каждого ребра новосозданного графа. «Побочные» ребра (из/в стока/истока) получают стоимость 0
- Находим алгоритмом минимальную стоимость перевозки
- Выводим результат

Баллы: 25

Авторский генератор и решатель

ВСТАВИТЬ	КОД	—
https://gist.github.com/Mikhail57/e88c3b6b2b4280f08b3f315b9c954a4d		

Задание 4. Парад

После победы в войне, Сергей Викторович решил провести в Храбровске парад. Но так как хотелось разнообразить уже сложившуюся традицию парадов, решили провести автомобильный парад — любой житель города может приехать в точку старта и пройтись в автоколонне. Мольква не резиновая, но об этом вспомнили только за день до парада, когда уже информация о точке старта и конца уже была объявлена. Встала проблема организации такого маршрута, по которому смогут пройти все (или почти все) зарегистрированные участники. Также необходимо учесть максимальное время проведения парада — 4 часа. Маршруты приоритезируются сначала по количеству участников (должно быть максимально), а затем по времени (должно быть минимально). Гарантируется, что как минимум один маршрут существует.

Входные данные

В первой строке N — количество зарегистрированных участников, $10 \leq N \leq 20000$

Во второй строке M — количество перекрестков в городе, $10 \leq M \leq 10000$

В третьей строке K — количество улиц в городе, $40 \leq K \leq 50000$

В четвертой и последующих K строках — описание улиц в формате $A \ B \ C \ D$, где C — максимальная пропускная способность улицы от перекрестка A до B за 1 минуту, D — время преодоления этого участка в минутах.

Парад начинается в точке 0 и заканчивается в точке $M-1$. Граф ориентированный.

Выходные данные

Вывести 2 числа через пробел: сколько участников сможет участвовать, и время в минутах, за которое пройдет парад. Выбирается маршрут, который занимает меньше всего времени.

Ответ

В задаче подразумевается, что люди — материальные точки, которые движутся по улице равномерно «порциями» по L человек раз в минуту, где L — минимальная пропускная способность маршрута. Также участники не могут отступать от обозначенного ранее маршрута. В задаче приведено избыточное условие про максимальное количество участников маршрута, которое не используется в тестах. Оптимизация происходит сначала по максимальному потоку, потом по времени.

Авторский алгоритм решения:

- Считываем входные данные
- Строим граф
- Находим кратчайшие маршруты
- Для каждого маршрута проверяем поток
 - Если поток больше текущего максимального, то считаем для него время прохождения
- Иначе пропускаем

- Выводим время получившегося маршрута и количество участников

Баллы: 25

Авторский генератор и решатель

ВСТАВИТЬ

КОД

<https://gist.github.com/Mikhail57/69bb5665b025f9db09e4dc1274192c82>

Блок теории программирования

Задание 1. Включить поворотники

Вопрос на знание программирования в игровом движке Unreal Engine 4

В нашем умном городе функционирует умная доставка еды, использующая беспилотные автомобили.

Строится оптимальный маршрут и машина выезжает по построенному пути.

Все алгоритмы работают отлично, однако тестировщики негодуют: почему автомобиль не включает поворотники?

Вам, как программисту, дали задание: написать небольшой модуль программы, который по трём точкам на плоскости — А, В, С будет определять, произойдет поворот по часовой стрелке или против часовой при смене направления из АВ в ВС, чтобы можно было понять какой поворотник включать (наш беспилотник следует по маршруту А -> В -> С).

Какую функцию вы будете использовать для определения поворота, ориентируясь на знак ("+" или "-") выходного значения функции?

Ответ:

Lerp

Length

GetForwardVector

GetControlRotation

SetActorRotation

Distance(Vector)

CrossProduct

Rotator

RotateVector

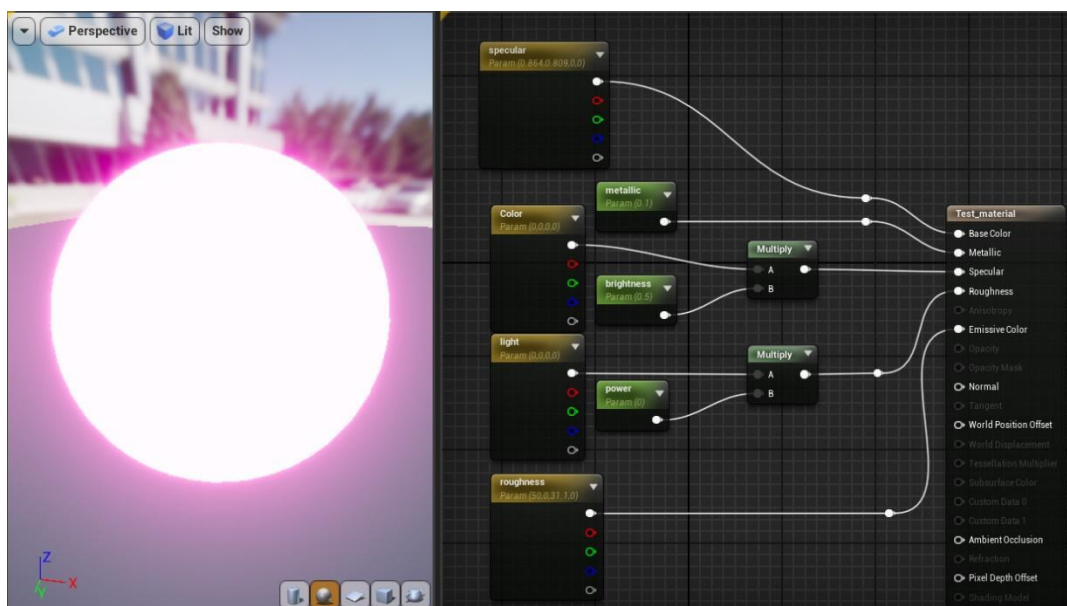
Баллы: 15

Задание 2. Материал света

Вопрос на знание программирования материалов в игровом движке Unreal Engine 4

Взяв на работу юного программиста в проект умного города, мы попросили его создать неоновый светящийся материал для вывесок магазинов, салонов и т.д. А также попросили использовать параметры и подписать их, чтобы удобно регулировать свойства материала прямо на сцене. Однако наш программист неграмотно составил материал, и при этом неправильно подписал все параметры. Придётся всё переделывать, поэтому не могли бы вы помочь нам?

Определите название ноды, которая в данном случае регулирует яркость цвета (света) материала?



Ответ:

color

Metallic

light

specular

roughness

brightness

power

Баллы: 15

Задание 3. Жёсткая конкуренция

Вопрос на знание программирования в игровом движке Unreal Engine 4

Наш умный город развивается не по дням, а по часам до дедлайна, но конкуренты не спят: они подкинули нашим программистам различные высказывания о классах в UE4, при этом какие-то высказывания правдивые, а другие – ложные!

Очень надеемся, что вы справитесь с тем, чтобы определить какие из высказываний правда, а какие ложь, **указав правдивые высказывания**, и поможете нам!

- ☒ Character - часть сцены, имеет положение в мире.
- ☒ Pawn - часть сцены, имеет положение в мире. Может реплицироваться и тикать.
- ☒ Uobject - основной класс для объектов unreal engine'a
- ☐ Класс pawn не наследуется от actor
- ☒ Actor - часть сцены, имеет положение в мире. Может реплицироваться и тикать.
- ☐ Класс uobject наследуется от actor

Баллы: 15

Задание 4. Потоки.

Вопрос на знание программирования в игровом движке Unreal Engine 4

Для быстрого функционирования умного города и чтобы игра шла в 4к 60 fps, необходимо провести оптимизацию кода и поработать над мультипоточностью, только вот наши программисты забыли названия инструментов для работы с ней.

Помогите программистам, указав основные способы работы с мультипоточностью в Unreal'e?

Ответ:

- ☐ Network Role
- ☐ Authority
- ☒ TaskGraph
- ☒ AsyncTask
- ☐ Substepping
- ☐ Replication
- ☐ Network Role
- ☒ Runnable

Баллы: 15

Задание 5. Беспорядок методов

Вопрос на знание программирования в игровом движке Unity

При постройке нового здания в нашем умном городе мы хотим показывать сообщение игроку о процессе постройки объекта. Однако скрипт, прикреплённый к зданию, не выводит никаких сообщений. Опытный программист, изучив наш код, посоветовал ознакомиться с методами жизненного цикла `MonoBehaviour` в Unity и определить их последовательность, чтобы мы смогли разобраться в ошибке.

Помогите определить в каком порядке вызываются методы жизненного цикла `MonoBehaviour`?

Awake
OnEnable
Start
FixedUpdate
Update
LateUpdate
OnDisable
OnDestroy

Ответ:

Баллы: 15

Задание 6. Застройка комплекса

Вопрос на знание программирования в игровом движке Unity

Нам необходимо собрать и настроить 3д модели зданий, чтобы потом внедрить их в умный город.

Где мы будем собирать и настраивать здания в Unity?

Ответ:

- ☐ MainScene
- ☐ GameObject
- ☐ Script
- ☒ Prefab

Баллы: 10

Задание 7. Каллиграфия.

Вопрос на знание программирования в игровом движке Unity

В программировании важны не только работоспособность алгоритмов и отсутствие багов, также важно понимание своего кода и его читаемость! Мы хотим создать класс дорогого здания, определите как грамотно дать название данному классу?

Ответ:

- ☐ Buildclasslux
- ☐ Build_Class_LUX
- ☐ Build_Class_Lux
- ☐ BuildClassLUX
- ☐ buildclasslux
- ☐ BuildclassLUX
- ☒ BuildClassLux

Баллы: 10

Задание 8. Процесс строительства

Вопрос на знание программирования в игровом движке Unity

Строительство здания - ответственный и длительный процесс. Как хорошо, что в умном городе, строчки кода строят все здания. Но есть проблема, построить здание по щелчку кнопки мы можем, однако также необходимо подождать какое-то время, чтобы здание построилось. Что же из перечисленного нам подойдёт, чтобы симулировать процесс строительства здания?

Ответ:

- ☐ WaitForSeconds
- ☒ Coroutine
- ☐ RequirereComponent
- ☐ Start
- ☐ Trigger

Баллы: 10

Блок Дизайн документ

ШАБЛОН ДИЗАЙН ДОКУМЕНТА

Раздел 1.

Сформулируйте общую информацию об игре:

- *Название игры*
- *Платформы*
- *Возраст целевой аудитории*
- *Титульная картинка, концепт арт*

Раздел 2.

Опишите начало игры, что происходит по ходу игрового сюжета, придумайте финал игры:

Например, Мэр города НТИ-Лэнд, решил расширить свой город и сделать город самым продвинутым и счастливым на всей земле.

Раздел 3.

Детально опишите главного героя игры:

Какими способностями он обладает, как он выглядит.

Раздел 4.

Опишите геймплей*, какие у игры “фишки” и особенные супер крутые уровни и задания:

Все наиболее прикольные фишки игры должны как можно сильнее заинтересовать.

Раздел 5.

Составьте описание игрового мира, как он выглядит, где происходят события, из каких локаций состоит:

Описания и изображения мира, какие здания можно будет строить, какие дороги, как выглядят жители города.

Раздел 6.

Опишите какие эмоции и настроение должна создавать игра, к чему побуждать или мотивировать, какая музыка используется для этого:

Какие эмоции и настроение должна создавать игра? Как музыка и звуки способствуют нужной атмосфере?

Раздел 7.

Придумайте и опишите игровые механики, которые используются в игре:

Различные механики игры.

Раздел 8.

Опишите, с какими трудностями будет встречаться игрок:

Какие вызовы ждут игрока?

Раздел 9.

Опишите сюжетные ролики, которые будут использоваться в игре:

Они есть в игре? Как они будут показываться? Видеоролики, флэш анимация, внутриигровые сцены на движке игры, слайдами с картинками. Когда они будут показываться? (между уровнями, в процессе игры). Для отсылок можно упомянуть примеры из фильмов.

Раздел 10.

Опишите бонусные материалы, которые получит игрок после победы:

Что убедит игрока продолжить игру после победы? Открытие нового режима или Игра+. Или игра в мультиплеере. Или скачиваемый контент. Или эпизодический контент и спец-события.

Критерии оценки	Баллы
Раздел 1	
Название игры	5
Платформы	5
Возраст целевой аудитории обусловлен геймплеем, сеттингом и сюжетом	5

Титульная картинка, концепт арт (приемлемое качество картинки)	5
Раздел 2	
Наличие описания сюжета	5
Раздел 3	
Наличие описания главного героя	5
Раздел 4	
Имеется чёткое краткое описания геймплея	55
Описаны необычные фишки, расписан геймплей	
Раздел 5	
Общее описание мира	5
Как выглядит мир:1) окружение,2) жители, 3) какие локации (если все три пункта имеются, то зачёт)	5
Раздел 6	
Описание эмоций, которые пытается вызвать игра у игрока	5
Имеется ссылка на музыку, или название треков, или точное описание саунд-дизайна (больше двух прилагательных)	5
Раздел 7	
Перечисление основных, главных механик, позволяющих понять, о чём будет игра	5
Имеется подробное описание механи-к/-ки (для чего нужна механика)	5
Раздел 8	

Перечисление челленджей и трудностей без подробностей	5
Имеется подробное описание челлендж-ей/а	5
Раздел 9	
Описание видеороликов (картинкой, или отсылкой к другой игре, к фильму, или исчерпывающим текстовым описанием)	5
Где и когда будут воспроизводиться ролики?	5
или	
Имеется исчерпывающее обоснование, что ролик не нужен (например, игра в жанре гипер-казуал, а значит ролики будут только мешать геймплею)	10
Раздел 10	
Описание бонусов после прохождения игры	5
Развёрнутое объяснение, почему игрок захочет перепройти игру	5
или	
Объяснение почему в данной игре нет необходимости перепроходить её	10

Решение:

Раздел 1.

Сформулируйте общую информацию об игре:

- *Название игры: Villagital Way*
- *Платформы: PC*
- *Возраст целевой аудитории: 12+*

- Титульная картинка, концепт арт



Раздел 2.

Опишите начало игры, что происходит по ходу игрового сюжета, придумайте финал игры:

Под управлением нового городничего (главного героя), глухой деревне предстоит дорасти до новой вехи в своем развитии. Попадает на эту должность герой по настоянию своего дедушки, по совместительству бывшего управляющего деревни – мудрого, но обаятельного старичка, который будет отвечать за обучение игрока в начале уровня. С каждым пройденным испытанием, деревня будет переходить на новый этап своей эволюции – сначала город, а позже и футуристичный мегаполис. Особенность в том, что город остается прежним и никто не удивляется резкому скачку в развитии, лишь меняющийся вместе с антуражем окружения старичок дает подсказки по новым условиям игры. Игра заканчивается в момент достижения городом определённого уровня развития и выполнения глобальной цели последнего этапа – озеленения экологии, путем замены всех работающих на угле производств и заводов, полезных в прошлом, на экологичные аналоги.

Раздел 3.

Детально опишите главного героя игры:

Главный герой – обычный молодой парень в начале игры, обязательно похожий на дедушку, которого мы встречаем в начале. Вместе с развитием города происходит и его преобразование: он становится взрослее и солиднее, а стиль его одежды соответствует эпохе в данный момент игры, при этом персонаж остается узнаваемым (ключевые черты внешности сохраняются). Дедушка также соответствует временной моде.

Раздел 4.

Опишите геймплей*, какие у игры “фишки” и особенные супер крутые уровни и задания:

Геймплей представляет собой смесь геймплея симулятора поселения (важность каждого отдельного жителя, а именно их индивидуальных потребностей) и симулятора современности (традиционные механики в виде постройки и развития города, распределения ресурсов), но также с вплетенным сюжетом (в виде исторического сеттинга) и событиями/заданиями, обеспечивающими продвижение по сюжету и переход на новый уровень.

На каждом этапе соответственно эпохе меняется антураж местности и ее функционал (например, смена зданий, назначением которых является развлечение: Дом культуры → Кинотеатр → VR-клуб), добавляются новые возможности и трудности с развитием технологий (так, например, на последнем этапе вводится показатель уровня экологии в городе и необходимость находить баланс между затратами на дорогую «зеленую» промышленность и дешевую «черную»).

Игроку необходимо контролировать количество, добываемых на трех предприятиях (лесопилка, рудник, шахта). Их запасы не бесконечные, а здания без них не построить. Кроме того, каждому зданию (кроме домов) требуются работники – они обеспечиваются строительством свободных жилых домов. Без свободных людей здания не смогут

работать. У каждого человека есть предпочтения в работе, выполнение которого (т.е. наличие свободных мест на данной работе) влияет на его уровень удовлетворённости. Кроме этого влияет наличие и доступ к некоторым видам зданий и, начиная со второго этапа, обеспечение электроэнергией, с которой тоже все не так просто. На третьем этапе вводится параметр экологии, который также прямо связан с уровнем счастья людей. Электростанции различаются по количеству выбросов в атмосферу, радиусу действия и цене содержания. Игроку необходимо найти баланс между тратой денег и экологичности энергетики, а главной целью игры становится замена всех предприятий на экологически чистые.

Казна города пополняется за счет траты людей денег в коммерческих зданиях. Их траты зависят от уровня удовлетворенности.

Раздел 5.

Составьте описание игрового мира, как он выглядит, где происходят события, из каких локаций состоит:

Игровой мир представляет собой разрастающийся за счёт действий игрока город (изначально деревню), с делением на зоны, согласно назначению зданий, среди которых и работа (на ферме, заводе), и промышленность (лесопилка, рудник, шахта), и развлечение (кино, цирк), и культура с образованием (церковь, школа, университет). По генерируемой сети города всегда передвигаются люди.

В совокупности с low-rolу графикой, стилизацию игре придают модели, выполненные в стиле минимализма, и ограниченная цветовая палитра, которая меняется на каждом этапе, подчеркивая контраст и визуально разделяя их:

Палитра деревни – теплая, в оранжевых и зеленых тонах. После перехода к городу палитра сменяется на более серую и монохромную. А на последнем этапе все выполнено в синих и фиолетовых тонах.

Раздел 6.

Опишите какие эмоции и настроение должна создавать игра, к чему побуждать или мотивировать, какая музыка используется для этого:

Игра должна создавать мотивационное воздействие на игрока. Из ностальгической и «теплой» атмосферы деревни он своими усилиями переходит сначала в довольно успешный, но серый город, а после и его более совершенный вариант. Кроме того, благодаря ориентированности на экологическую составляющую на последнем этапе игры, мы обеспечиваем популяризацию идеи сохранения природы и заботы об окружающем мире.

Саундтрек должен погружать игрока в эпоху текущего уровня, т.е. подходить по стилю и времени. Кроме того, в определенные моменты, музыка также отражает настроение происходящей ситуации (катастрофа → гнетущая музыка, новая победа → радостная музыка).

Раздел 7.

Придумайте и опишите игровые механики, которые используются в игре:

Базовый геймплей градостроительного симулятора (строительство зданий и дорог, их взаимодействие) будет подкреплён интересными особенностями:

- баланс добычи ресурсов, которые ограничены, но требуются для строительства зданий*
- регулирование свободного населения, которое имеет свои предпочтения в работе*
- контроль за уровнем прибыли и расходов*
- деление промышленности на экологичное и неэкологичное; трудности с содержанием более дорогих, но приятных жителям предприятий, также отличающихся по радиусу действия*

- регулирование уровня счастья жителей, от которого зависит прибыль города

Раздел 8.

Опишите, с какими трудностями будет встречаться игрок:

Ключевой целью игрока является строительство города и решение проблем по его управлению, которые будут расширяться с каждым уровнем. Так, на первом этапе (деревня) будет необходимо обеспечить добычу ресурсов (дерево – лесопилка, уголь – шахта) и их рациональное использование (на здания, дороги) и добычу (их количество ограничено). У каждого жителя есть свой уровень счастья, который зависит от нескольких факторов – наполненность города разными зданиями, удовлетворённость в предпочитаемой им работе, электричество в доме и уровень экологии в городе. С продвижением по эволюционной лестнице трудности и возможности наращаются - придется справляться с загрязнениями, находить баланс между легким, но вредящим и трудным, но правильным.

Раздел 9.

Опишите сюжетные ролики, которые будут использоваться в игре:

В самом начале игрок наблюдает ролик от первого лица с приездом внука к дедушке в деревню.

В качестве визуального элемента сюжета, будут встроены диалоги с дедушкой, в начале обучающие, а после направляющие главного героя по сюжету. Интерактив обеспечивает смена спрайта-эмоций персонажей (на мотив визуальных новелл).

При переходе между этапами будут вставлены видеоролики преобразования деревни – ее плавного улучшения до следующего уровня.

Раздел 10.

Опишите бонусные материалы, которые получит игрок после победы:

После победы игрок получит доступ к бесконечной игре и к новому режиму сложности, где жители будут ставить мэра в еще более сложное положение своими предпочтениями, а к экологичной энергии добавляется водоснабжение и продовольствие, что мотивирует игрока перепроходить игру

Баллы: 100 баллов.

Блок Программирование генерации

Задача 1: Генерация локаций

Сейчас вам предстоит создать генератор квартала города из своих блоков! Вам необходимо в своем любимом движке создать демо-приложение, которое будет генерировать квартал на основе случайного зерна с учётом игровой сетки. При выборе одинакового зерна должны создаваться одинаковые кварталы. Зерно должно влиять на расположение зданий, дорог, объектов оформления квартала, тип квартала/зданий. Но не всё так просто...

Вашему гейм-дизайнеру предстоит разработать схему или текстовое описание алгоритма генерации, который будет учитывать элитарность — в дорогом квартале не могут стоять дешёвые здания, а потенциальные потоки людей не должны сильно пересекать квартал элитарной застройки.

В качестве ответа вам необходимо будет на следующих шагах загрузить результаты работы своего гейм-дизайнера и программиста.

На первом шаге вы загружаете блок-схему и/или текстовое описание алгоритма на псевдокоде. Также вам необходимо будет добавить текстовое описание механики генерации.

На втором шаге вы должны будете прикрепить результаты разработки своего демо-приложения: ссылка на [github/gitlab](#) репозиторий с исходным кодом.

*Кварталы здесь — небольшая система зданий (точек интереса), с которой жители смогут в последствии полноценно взаимодействовать.

Вам необходимо загрузить исходный код своего проекта на один из ниже представленных сервисов:

GitLab

GitHub

И предоставить доступ на чтение для аккаунта со следующими никнеймом: ONTI-GameDev

Код должен быть в главной ветке (master или main)

Решение задачи 1:

_____Seed (Зерно)_____:

seed - числовая переменная в 18 знаков, с помощью которой происходят все выборы в генерации.

Изменение зерна происходит после каждой операции выбора с использованием зерна.

Изменение зарна происходит по такой формуле: $PI * \text{Sqrt}(\text{seed}) * (\text{seed} \% 1e9 + 1)$

_____Генерация дорог (первая генерация)_____:

Данный алгоритм генерирует дороги на основе свободных точек, выбирая данные точку по приоритету, который определяется с помощью кода зерна.

Чтобы эффективно определять элемент (точку), необходимо использовать корневую декомпозицию (в дальнейшем будем называть её СД - структура данных)

Алгоритм генерации:

- 1) Создаётся начальная дорога и добавляется в СД;
- 2) Запускается цикл, который работает до тех пор, пока не наберёт нужное кол-во дорог, либо не превысит ограничение по времени задаваемое пользователем.

- 3) В цикле:

Из СД передаётся точка;

Определяется угол и длина по зерну;

Находится вторая точка новой дороги;

Проверяются коллизии новой дороги с помощью проверки пересечения прямых;

Если коллизии не обнаружены, то дорога создается и её вторая точка добавляется в СД, а также крайние точки дороги добавляются в двумерный массив поля;

В цикле находятся $\min X$, $\min Y$, $\max X$, $\max Y$ для поиска центра города по формулам:

$$\text{center}X = (\max X + \min X) / 2;$$

$$\text{center}Y = (\max Y + \min Y) / 2;$$

Генерация кварталов (вторая генерация):

Данный алгоритм генерирует кварталы с помощью заполнения двумерного массива индексами кварталов для последующего использования при генерации домов.

Чтобы удобнее было взаимодействовать с точками, будем хранить их в очереди. Доставая каждую, мы будем расходиться от него во все соседние по вершине незанятые клетки и заполнять их кварталом.

Алгоритм

1) В центр записывается индекс квартала (выбирается по зерну) и добавляет точку в очередь;

2) Заполняет начальные положения кварталов: всего 8 начальных клеток, из которых заполняется 4 клетки (кол-во кварталов), выбранные с помощью зерна. Каждая выбранная клетка добавляется в очередь;

3) Запускается цикл, который работает до тех пор, пока не заполнит весь массив кварталами и очередь не станет пуста;

4) В цикле:

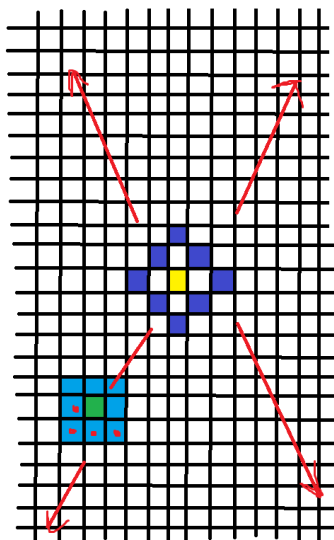
Из очереди забирается точка;

Проверяются все свободные места вокруг;

В свободные места ставятся индексы того же квартала;

Новые точки, которые мы заполнили, добавляются в очередь;

Жёлтый - центр города
 Синий - начальное место
 расположение
 кварталов
 Красный - пути
 заполнения массива
 кварталами
 Зелёный - точка, взятая
 из очереди в одну из
 итераций цикла
 Голубой - проверяемые
 точки
 Красные точки - точки,
 выбранные из зелёной
 ячейки, которые
 попадут в очередь и
 будут дальше
 распространять квартал



_____Генерация парковок (третья генерация)_____:

Данный алгоритм генерирует парковки через точки около дорог.

Для случайного расставления парковок будем использовать корневую декомпозицию, в которую определим все доступные точки с приоритетами, выбираемыми по зерну.

Также, для равномерного расположения парковок по городу, создана переменная, показывающая кол-во парковок в квартале, а для поддержания этого количества парковок заведём обычный массив, где в ячейке с номером квартала хранится кол-во парковок в данном квартале.

Алгоритм:

1) Запускается цикл, который перебирает все дороги:

Вычисляется прямая дороги, её длина и угол;

Запускается цикл, который перебирает все целочисленные длины от 1 до длины дороги (итератор len):

Находится точка на расстоянии len от начала дороги;

Находятся точки по обе стороны дороги;

Эти точки добавляются в СД с приоритетами, выбираемыми по зерну;

2) Запускается цикл, который работает до тех пор, пока не наберёт нужное кол-во парковок:

Из СД забирается точка;

Проверяется кол-во парковок в квартале;

Если оно не превышает заданного максимума, тело цикла продолжается;

Если кол-во точек в СД меньше либо равно кол-ву парковок, которые нужно достроить, то тело выполняется в любом случае, независимо от предыдущего условия;

Для парковки в этой точке проверяются коллизии с помощью пересечений прямоугольников с прямыми.

Если коллизии не найдены, то ставится парковка, в этой точке и прибавляется счётчик парковок в квартале, далее она добавляется в двумерный массив поля.

_____Генерация коммерции (четвёртая генерация)_____:

Данный алгоритм работает аналогично генерации парковок, кроме проверки коллизий. Тут они проверяются с помощью пересечений прямоугольников с другими прямоугольниками.

_____Генерация домов (пятая генерация)_____:

Данный алгоритм заполняет весь оставшийся город домами на основе кварталов и зерна.

Алгоритм:

1) Запускается цикл, который перебирает все дороги - дома ставятся пока не закончится место:

Вычисляется прямая дороги, её длина и угол;

Запускается цикл, который перебирает все целочисленные длины от 1 до длины дороги (итератор len):

Находится точка на расстоянии len от начала дороги;

Находятся точки по обе стороны дороги;

Выбирается дом для каждой точки (на основе квартала в точке и зерна);

Проверяются коллизии для этих домов (с помощью пересечений прямоугольников с прямоугольниками и пересечений прямоугольников с прямыми);

Если у дома нет коллизий, то он ставится.

_____Генерация графа (шестая генерация)_____:

Данный алгоритм заполняет список смежностей (для хранения графа) и два словаря: для перевода объекта в индекс в графе и для перевода индекса в графе в объект. С помощью словарей мы сможем получать быстрый переход между индексом и объектом.

Граф в нашей системе нужен для прокладывания маршрутов машин и людей.

Алгоритм:

1) Заполнение словарей:

Четыре цикла по массивам, построившихся объектов (дома, паркинги, коммерция, перекрёстки);

Один из циклов на псевдокоде для лучшего понимания:

```
for (int i = 0; i < Количество домов; ++i) {
    словарь из объекта в индекс.Add(i-ый элемент массива домов,
    количество элементов в данном словаре);
    словарь из индекса в объект.Add(количество элементов в данном
    словаре, i-ый элемент массива домов);
}
```

2) Заполнение графа:

Запускается цикл, который перебирает все пары (ключ, значение) в словаре из объекта в индекс:

В цикле есть два варианта развития событий:

1) Если это какая-то постройка

Запускается цикл, итерирующийся по дорогам, присоединённым к постройке:

Запускается цикл, итерирующийся по перекрёсткам, принадлежащим этой дороге:

Рассчитывается длина между перекрёстком и постройкой

Создаётся ребро с точками в постройке и перекрёстке и данной длиной

2) Если это перекрёсток

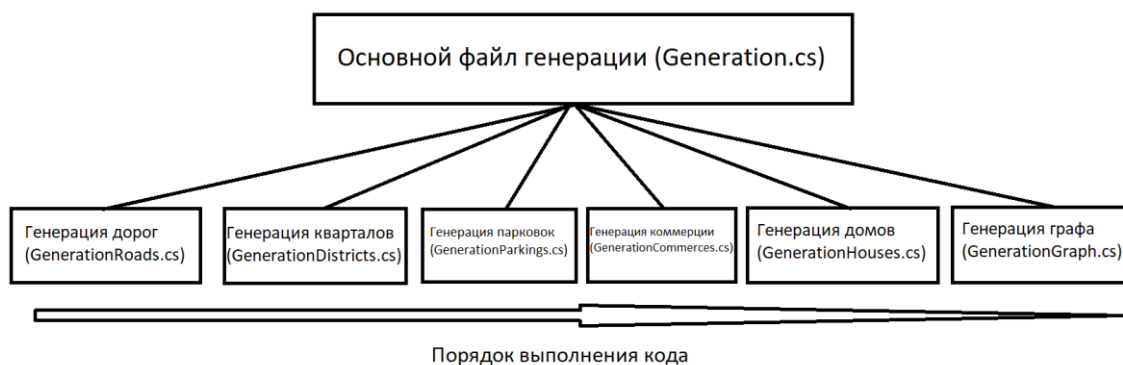
Запускается цикл, итерирующийся по дорогам, присоединённым к перекрёстку:

Запускается цикл, итерирующийся по перекрёсткам, принадлежащим этой дороге:

Если ребра между нашим перекрёстком и перекрёстком в цикле нет, то вычисляется длина и создаётся ребро с точками в этих перекрёстках и длиной и данной длиной

 Главный файл генерации (Схема прилагается):

В данном файле включаются все функции генерации по очереди. Также в нём изменяется зерно и рассчитывается время.



Баллы: 50 баллов.

Задача 2: Загрузка на Git

Задачи и критерии: предоставить доступ на чтение для аккаунта со следующим никнеймом: ONTI-GameDev, код должен быть в главной ветке (master или main)

Решение задачи 2:

 master ▾  5 branches  0 tags

[Go to file](#) [Code ▾](#)

 ProMix0 Logging

5977495 21 days ago  105 commits

 Assets	Logging	21 days ago
 Packages	Restructurize project and upgrade to Unity 2019.4.20f1	26 days ago
 Project/Assets/Resources	Merge branch 'models'	22 days ago
 ProjectSettings	Some changings?..	24 days ago
 docs	Restructurize project and upgrade to Unity 2019.4.20f1	26 days ago
 .gitattributes	Restructurize project and upgrade to Unity 2019.4.20f1	26 days ago
 .gitignore	Restructurize project and upgrade to Unity 2019.4.20f1	26 days ago

Баллы: 10 баллов.

Блок демонстрации геймплея и генерации кварталов

Основная задача блока: записать видеоролик с демонстрацией геймплея вашей игры.

Длительность видео до 2 минут.

Что нужно продемонстрировать в видеоролике?

1. Генерация квартала (начальной локации).
2. Запуск мира.
3. Создание (прокладка) дополнительных дорог и создание зданий рядом с дорогами.
4. Функционирование всей итоговой локации.

Необходимо отправить сюда ссылку на ваше видео в YouTube: (в качестве ответа в системе Stepik)

Пункт №1. Генерация квартала (начальной локации)

Основные критерии:

1. Сетка уровня
2. Создание первой локации
3. Создание второй локации
4. Создание третьей локации
5. Создание первой локации снова
6. Все локации создаются без перезапуска игры, различных склеек и т.п.
7. Все статические элементы должны соответствовать сетке мира (различные здания, дороги не должны накладываться друг на друга)
8. Наличие визуального интерфейса

1) Создать сетку уровня

Сетка уровня — система, по которой будет определяться постройка, расположение, размер зданий и дорог.

Делаем удобным вам и вашему геймплею способом.

2) Создать генератор локаций города

Вам необходимо в своем любимом движке создать демо-приложение, которое будет генерировать квартал (или другой вид небольшой локации, соответствующей вашему игровому миру) на основе случайного зерна. При выборе одинакового зерна должны создаваться одинаковые локации. Зерно должно влиять на расположение зданий, дорог, объектов оформления квартала, тип квартала/зданий.

Демонстрация работы данного алгоритмы — это первые **10-30 секунд** видеоролика, в нём должны присутствовать:

1. Создание локации по случайному зерну 1
2. Создание локации по случайному зерну 2 без перезапуска приложения
3. Создание локации по случайному зерну 3
4. Создание локации по случайному зерну 1 (должен создаваться локация того же типа, что и в первый раз)

Сгенерированная локация должна содержать различные «точки интереса» для жителей, которые будут приходить к данными точкам и взаимодействовать с ними (парк — жители будут там гулять, школа — ученики будут ходить в школу и какое-то время пребывать внутри, ферма папоротников под стеклянным куполом (может быть и такая точка интереса, если, например, действия происходят на Марсе) взаимодействия могут быть абсолютно разные, всё зависит от вашей задумки).

Пункт №2. Запуск мира

Основные критерии:

1. **Функционирование жителей** (визуальное качество моделек, анимация, параметры жителя, поведение)
2. **Функционирование зданий и построек** (визуальное качество зданий, адекватность расположения, влияние на жителей)
3. **Дополнительная наполненность города**, необходимая для вашего конкретного геймплея (машины, телефонные будки, заправки) .

*Визуальное качество моделек — это, в большей степени, соответствие общему внутри-игровому стилю, который вы сами выбрали.

Сейчас мы наполним наш город жителями:

Необходимо создать характеристики для жителя, которые будут влиять на его поведение, модельку и одежду:

1. Возраст (от 0 до 99)
2. Пол (мужской /женский или авторские)
3. Социальный статус (ребенок, школьник, студент, работающий человек, безработный, пенсионер)
4. Бюджет (может быть отрицательным)
5. Удовлетворённость в чем-либо — авторская характеристика, за которой будет необходимо следить, чтобы понимать, какие постройки необходимо строить, чтобы житель был удовлетворён (от 0 до 100 у.е.)
6. Ещё одна (похожая на вышеприведённую) авторская характеристика, основывающаяся на вашем геймплее (от 0 до 100 у.е.)
7. Предпочтения в чём либо, например, в еде (русская, американская, японская, корейская и т.п.) — также, как и прошлые две, будут толкать игрока на создание определённых зданий, условий жизни.

Вам необходимо реализовать систему хранения и отображения информации о каждом из людей в этом виртуальном мире. Информацию по ним можно получить путем **выделения одного из жителей**. При этом отображается **информация о персонаже на панели**, без блокирования возможности совершения других действий пользователем. Если количество

параметров для отображения большое, можно на панели отображать только основные, отображающие важные, по вашему мнению, параметры, а остальные отображать в дополнительном диалоге/окне/панели.

Жители также должны функционировать по их характеристикам и должны легко читаться игроком, например:

Характеристики жителя: школьник, ребёнок, скейтер, а значит детская моделька персонажа, с рюкзаком и в кепке, ходит в школу, затем обратно домой, оставляя там рюкзак, выходя на улицу со скейтом, направляясь в парк, далее катается в парке, потом возвращается домой, и далее снова в школу и так по кругу.

Все модельки должны адекватно себя вести в заданной роли, не нарушая правил игры, проигрывать анимации, соответствующие их текущим действиям.

Количество жителей на квартал должно **превышать 8**, то есть один житель олицетворяет определённый слой общества (например, школьник + спортсмен, школьник + олимпиадник, они оба ходят в школу, значит, им обоим необходима школа, но одному необходим спортивный зал, а другому научный центр).

У каждой дороги должны находиться тротуары, по которым жители будут передвигаться, а переходить дорогу в установленном для этого месте (по пешеходному переходу, подземному переходу и др.).

На видео необходимо запустить какую-либо из сгенерированных локаций, и проследить за её функционированием, а также продемонстрировать возможность выбора жителя и отображения его характеристик.

Сначала обзор на локацию **10-15 секунд**, затем выбор жителя и демонстрация его характеристик на протяжении **4-6 секунд**. Таким образом необходимо продемонстрировать **четырёх** жителей.

Пункт №3. Создание (прокладка) дополнительных дорог и создание зданий рядом с дорогами

Основные критерии:

1. Адекватность строения дорог и зданий (построенные здания и дороги соответствуют игровой сетке, не «наезжают» друг на друга, и внешний вид соответствует общей стилистике, выбранной вами)
2. Соответствие ниже представленным требованиям.

1) Продемонстрировать создание элемента (модуля) дороги к уже имеющемуся элементу дороги:

А) При нажатии определённой клавиши на клавиатуре или на UI кнопке создать **сплошной модуль** дороги к какой-либо из уже находящейся на сцене дороге.

Б) При нажатии определённой клавиши на клавиатуре или на UI кнопке создать **модуль поворота** дороги к какой-либо из уже находящейся на сцене дороге.

В) При нажатии определённой клавиши на клавиатуре или на UI кнопке создать **модуль перекрёстка трёх** дорог к какой-либо из уже находящейся на сцене дороге.

Г) При нажатии определённой клавиши на клавиатуре или на UI кнопке создать **модуль перекрёстка четырёх** дорог к какой-либо из уже находящейся на сцене дороге.

Длительность демонстрации постройки дорог до 15 секунд

2) Продемонстрировать постройку здания жилого типа размера М у дороги:

А) При нажатии определённой клавиши на клавиатуре или на UI кнопке выбрать место жилого здания, занимающего **М элементов сетки** (зависит от выбранного вами масштаба сетки).

Б) После выбора места должна появиться моделька стройки здания (ограждение по периметру выбранного места забором серого или коричневого оттенка), а также запуститься **таймер** постройки над зданием в **3-5 секунд** (анимация UI элемента над зданием).

В) По истечению таймера появляется здание жилого типа **размера М элементов сетки**.

3) Продемонстрировать постройку здания жилого типа размера М*2 у дороги:

А) При нажатии определённой клавиши на клавиатуре или на UI кнопке выбрать место жилого здания занимающего М элементов сетки (зависит от выбранного вами масштаба сетки).

Б) После выбора места должна появиться моделька стройки здания (ограждение по периметру выбранного места забором серого или коричневого оттенка), а также запуститься **таймер** постройки над зданием в **3-5 секунд** (анимация UI элемента над зданием).

В) По истечению таймера появляется здание жилого типа **размера М*2 элементов сетки** (увеличивается размер по сравнению с другим домом в длину или ширину).

4) Продемонстрировать постройку магазина (или другой конкретной точки интереса) у дороги:

А) При нажатии определённой клавиши на клавиатуре или на UI кнопке выбрать место магазина на вашей сетке.

Б) После выбора места должна появиться моделька стройки здания (ограждение по периметру выбранного места забором серого или коричневого оттенка), а также запуститься **таймер** постройки над зданием в **3-5 секунд** (анимация UI элемента над зданием).

В) По истечению таймера появляется здание магазина.

Итоговая длительность демонстрации постройки зданий до 15 секунд

Пункт №4. Функционирование всей итоговой локации

Данные здания, построенные в прошлом шаге, должны начать функционировать:

1) Из жилых зданий должны появляться жители, которые соответствуют критериям из пункта 2.

2) Теперь все жители должны ходить в те «точки интереса», которые им ближе всего и подходят под их роли.

Показать целостную картину поведения локации в течение 15 секунд.

Заметка: правильно представить в видео данные пункты – задача по геймдизайну. Теперь осталось правильно упаковать данные механики в целостную демку. Видео должно отражать ваш собственный геймплей игры, явно сохраняя те механики, которые даны для оценивания, то есть вы можете добавлять какие-то свои механики, которые помогут раскрыть или просто не помешают нашим критериям, например, динамическая камера (перемещение по локации, приближение - отдаление), дополнительные UI элементы.

Критерии оценивания генерации:

Алгоритм генерации	
Присутствует код/блок-схема	1
Присутствует текстовое описание	1
Соответствие текстового описания коду	1

Логичность и структура алгоритма	2
Представление алгоритма	2
Реализуемость алгоритма	2
Использование «зерна»	2
Использование алгоритма в игре	4
Пересечение зданий/дорог	2
Итог	17
Гит	
В главной ветке код генерации сетки	2
В главной ветке код всего приложения	8
Итог	10

Итого баллов: $17 \cdot 5.88 + 10 \cdot 10 = 200$ баллов.

Запуск	
Игра запускается	1
Мир создается	1
Итог	2
Генерация квартала	
Присутствует сетка уровня (разделение для зданий)	2
Создание локации по случайному зерну 1	4
Создание локации по случайному зерну 2 без перезапуска приложения	3

Создание локации по случайному зерну 3	1
Создание локации по случайному зерну 1 (ещё раз)	1
Итог	11
Характеристики жителей	
Имеется хотя бы одна характеристика	1
За каждую характеристику у перса дается 1 балл (не больше 7)	7
Показаны 4 жителя (за каждого по баллу)	4
Итог	12
Общие критерии	
Сетка присутствует	1
Грамотно сделана сетка, незаметно, но удобно	2
Итог	3

Итого баллов: $(2+11+12+3)*3.57 = 100$ баллов.

Критерии оценивания геймдизайна:

Генерация квартала	
Показано в нужной последовательности	2
Сгенерированная локация должна содержать различные «точки интереса» (жители должны заходить в них или хотя бы стоять рядом с ними)	4
Итог	6

Дизайн начальной локации	
Имеются и дороги, и здания (местоположение не важно)	2
К каждому зданию ведётся дорога	2
Здания не загромождены и не загораживают друг друга	2
Имеются пропсы, тротуары и другие элементы окружения	2
Все модели соответствуют одному визуальному стилю	4
Итог	12
Характеристики жителей	
Жители имеют чёткое визуальное разделение по возрасту (размер/оформление/etc модельки)	2
Модели жителей имеют чёткое разделение по цвету	2
Жители имеют особый элемент определённого цвета, и данный элемент соответствует роли жителя.	8
Жители не имеют цветного элемента, однако часть тела или всё тело раскрашено в цвет и соответствует роли жителя. (зачтено, если пункт выше также зачтён)	4
Моделька жителя выполнена в одном визуальном стиле с остальным городом и качественно	4
Имеется непрерывистая анимация ходьбы жителя	6
Имеется дополнительная анимация (например ожидания)	10
Жители спавняются по-очерёдно	2

Имеется 8 и более жителей, одновременно на сцене	4
Жители двигаются по тротуарам	2
Жители двигаются по тротуарам и пешеходным переходам (не нарушая правила ПДД)	6
Логика движений слажена, жители не проходят друг через друга, всё работает без багов	8
Итог	58
Создание (прокладка) дополнительных дорог и создание зданий рядом с дорогами.	
Имеется интерфейс для создания дороги	2
Имеется возможность добавления дороги	6
Дорога создаётся у другой дороги без наложений или свободного пространства	6
Можно проложить сплошной модуль дороги	4
Можно проложить модуль поворота дороги	4
Можно проложить модуль перекрёстка трёх дорог	4
Можно проложить модуль перекрёстка четырёх дорог	4
Здания создаются	4
Здание создаётся ровно относительно дороги, остальных зданий (по сетке)	4
Имеется анимированный и стилизованный UI при постройке здания (или анимация появления)	6
Имеется счётчик таймера над зданием	4

Второе здание имеет отличные текстуры от первого	4
Имеется моделька при постройке (достаточно куба)	4
Забор стилизован (моделька постройки выполнена с одинаковыми детализацией и визуальным стилем будущего здания)	8
Размер второго здания заметно больше первого в ширину и/или длину	6
Итог	70

Функционирование новых местностей, если построили прошлые здания

Жители сразу начали спавниться из жилых зданий при их постройке	4
Новые жители не «конфликтуют» с уже имеющимися жителями (не проходят друг через друга)	8
Новые жители следуют ПДД	6
Новые жители соответствуют визуальному стилю	2
В новую точку интереса ходят хотя бы какие-то NPC	8
Жители ходят в здания, которые им ближе всего	8
Итог	36

Общие критерии

Имеется движение камеры (свободная камера)	2
Имеется зум камеры	2
Общий визуальный стиль не нарушен, приятные и проработанные модельки	6

Присутствует приятное освещение или погодные условия	2
Присутствуют необычные или хорошие механики, которые дополняют геймплей, не мешая проверке	4
Видео хорошего качества (720p и выше)	2
Итог	18
Всего первичных баллов	200

Решение:

Ссылка на видео-решение:
<https://drive.google.com/file/d/1bZMxiJ7wrAqxR7ux0ruCmJvO3SpeRKhS/view?usp=sharing>

Этапы	Баллы
Индивидуальный этап	400
1) Блок Геймдизайн	1) 100
2) Блок теории программирования	2) 100
3) Блок программирования	3) 100
4) Блок моделирования	4) 100
Командный этап	600
1) Блок дизайн-документ	1) 100
2) Блок моделирование жителей	2) 100
3) Блок программирование генерации	3) 200
	4) 200

4) Блок демонстрация геймплея	
----------------------------------	--

Итого максимум 1000 баллов за второй отборочный этап.