

Название: «Создать свою игру – это просто!?»

Автор: Гноевенко Иван Викторович, педагог дополнительного образования, г.Благовещенск

Направленность: техническая

Цель: формирование у обучающихся первоначальных навыков разработки компьютерных игр в процессе создания минимальной рабочей версии игры в среде разработки Unity.

Задачи: познакомить обучающихся со спецификой профессии гейм-дизайнера;

- сформировать у обучающихся представление о среде Unity;
- сформировать навыки создания минимальной рабочей версии игры в среде Unity;
- сформировать навыки работы в команде;
- развить логическое, творческое мышление, навыки публичного выступления.

Артефакт: цифровой прототип минимальной рабочей версии игры в среде Unity;

Краткое описание:

В течение мастер-класса обучающиеся смогут представить себя в роли гейм - дизайнеров, получить опыт создания минимальной рабочей версии игры в среде Unity, а также продемонстрировать свои творческие способности.

ПАСПОРТ ПРАКТИКИ	УСЛОВИЯ РЕАЛИЗАЦИИ		
	Время	Форма	Методы
	2 часа	1. Общая групповая работа 2. Работа в малых группах, 3. Групповая и индивидуальная работа	Кейс - метод, проектный метод, интерактивные методы, иллюстративно-демонстрационный метод, беседа, рассказ, приёмы дизайн-мышления, «мозговой штурм»
	КАТЕГОРИЯ ОБУЧАЮЩИХСЯ		
	Возраст	Особенности	Кол-во участников в рабочей группе
	12-17 лет	«Не адаптирована для детей ТЖС, ОВЗ»	12 человек
	РЕСУРСЫ		
	Оборудование и материалы	Базовые знания из других областей	Уровень сложности и Место в структуре курса
	ПО Unity, флипчарты, набор карандашей, бумага А4, набор магнитных карточек.	Базовые знания в области информатики: знать основы работы с персональным компьютером, иметь навыки запуска и установки программного обеспечения.	Вводный, автономный
	ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ		
	Hard-skills, предметные	Soft-Skills, метапредметные	Личностные

навыки	навыки	
<ul style="list-style-type: none"> - навык бумажного прототипирования; - навык визуализирования идеи; - навыки работы в среде Unity. 	<ul style="list-style-type: none"> - навыки работы в команде; - навыки публичного выступления; - коммуникационные навыки. 	<ul style="list-style-type: none"> - формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, проектной деятельности.
ОПЫТ РЕАЛИЗАЦИИ		
Инструкции для педагога	Открытая информация о результатах	Участие в конкурсах
Приложения 1-7 Презентация (Приложение 1) стр.13 Рекомендации по созданию игровых уровней в Unity (Приложение 2) стр.23 Пример кода для управления персонажем (Приложение 3)стр.27 Примеры рефлексии	отсутствует	Конкурс «Сердце отдаю детям» региональный этап 2023 года.

Этап 1. ПОСТАНОВКА ЦЕЛИ

Рекомендованное время: 25 минут

Задание/Активность:

№ шага	Вид деятельности	Время (мин.)
1.1.	Организационный момент	2 мин
1.2.	Мозговой штурм «Чем занимается геймдизайнер?»	10 мин
1.3.	Рассказ педагога о профессии геймдизайнера	8 мин
1.4.	Демонстрация/показ/представление педагогом варианта минимальной рабочей версии (https://drive.google.com/file/d/1WLs0a1U7vdRHmYdBA0mbbE138Of5O2kQ/view?usp=drive_link)	5 мин

1.1 Организационный момент:

педагог приветствует участников мастер-класса, знакомится с ними.

1.2 Мозговой штурм «Чем занимается геймдизайнер?»

Педагог предлагает ребятам в течение 3 минут обсудить и представить свою версию профессиональной деятельности геймдизайнера. Ребята в группах обсуждают поставленную проблему, оглашают свои варианты. Задают оппонентам вопросы. Педагог отмечает правильные ответы, обобщает информацию.

1.3 Рассказ педагога о профессии геймдизайнера

Педагог рассказывает о профессии геймдизайнера.

Геймдизайнер — специалист, отвечающий за разработку правил и содержания игрового процесса создаваемой игры.

Обязанности в разных студиях могут сильно отличаться — чем больше студия, тем уже специализация. Но чаще всего в задачи входит:

- придумывание общей концепции игры;
- разработка прототипов для тестирования идей;
- коммуникация с издателями и продюсерами;
- проработка игрового процесса;
- коммуникация с программистами и художниками;
- ведение проектной документации;
- планирование этапов разработки и контроль качества;
- написание текстов;
- аналитика метрик;
- анализ трендов рынка.

Педагог выясняет у обучающихся, хотели бы они научиться создавать игры самостоятельно?

Педагог рассказывает о базовых инструментах дизайна игр.

Придумывание общей концепции игры

Когда специалисту приходит в голову идея игры, он описывает её в документе, который называется «Концепт-документ» (Concept Document). Там он фиксирует идеи основных механик, список фиц (главных особенностей игры), список похожих существующих игр, краткий синопсис сюжета.

Многие из идей, записанных в документе на этом этапе, могут поменяться к тому моменту, когда игра дойдёт до релиза. Но, зафиксированные на бумаге, они помогают начать планировать следующие этапы разработки — например, понять, насколько большая команда специалистов понадобится для реализации проекта.

Разработка прототипов для тестирования идей

Существующий на рынке инструментарий (например, такие бесплатные игровые движки, как Unity, Unreal Engine и GameMaker) позволяет создать простой прототип игры в одиночку без глубоких знаний программирования. Имея «играбельный» прототип механики, дизайнеру будет гораздо проще оценить жизнеспособность своей идеи и объяснить её команде.

Коммуникация с издателями и продюсерами

Когда у команды разработчиков появляется первый рабочий прототип игры со всеми фицами (он называется «вертикальный срез»), начинается поиск издателя. Гейм-дизайнер создаёт на основе прототипа и концепт-документа презентацию и придумывает, как будет питчить (продвигать) будущий проект потенциальным издателям. Лично и онлайн общение с издателями продолжается до самого релиза игры и несколько месяцев после. Многие команды предпочитают путь независимых студий, выпускают игру без издателя и занимаются финансированием и продвижением своей игры сами.

Проработка игрового процесса

Игровой дизайнер отвечает как за отдельные механики, так и за весь игровой процесс (геймплей) в целом. Ещё когда игра не готова, у него есть видение какой опыт получит игрок на каждом её этапе. На этот опыт влияет и дизайн уровней игры, и то, как будут выглядеть локации, и с кем персонаж игрока на этих локациях встретится.

Коммуникация с программистами и художниками

Идеи гейм-дизайнера реализуют программисты, художники, 3D моделлеры, аниматоры, композиторы и другие члены команды. Одна из главных задач ГД — ставить им задачи и контролировать ход выполнения. При этом важно уметь найти подход к каждому члену команды и наладить в коллективе доверительные отношения — хорошо зная свою команду, гейм-дизайнер будет понимать кто с какой задачей справится лучше.

Ведение проектной документации

Важно, чтобы у всех разработчиков на проекте было единое видение игры, которую они создают. В этом помогает документация, которую гейм-дизайнер поддерживает в актуальном состоянии. К проектной документации относится как концепт-документ, о котором речь шла выше, так и более подробный дизайн-документ (Game Design Document).

Отдельные задачи для каждого специалиста фиксируются в техническом задании (ТЗ). ТЗ может быть оформлено в виде письма, отправленного на почту разработчику, а может быть в виде задачи в программе-таск трекере (например, в Trello, Asana или Jira).

Планирование этапов разработки

Если команда использует методологию Scrum, то гейм-дизайнер планирует загрузку всех разработчиков задачами на две недели вперёд. Если выясняется, что какую-то механику, локацию или персонажа не удастся реализовать как задумано, задача геймдизайнера — вовремя увидеть проблему и скорректировать задачи своей команды.

Написание текстов

Если в команде нет сценариста, дизайнер игр отвечает за написание всех сюжетных текстов: сценария, диалогов, квестов, описания предметов. Часто его могут попросить написать и маркетинговые тексты: описания игры для Steam и других площадок.

Аналитика метрик

Начинающие разработчики часто недооценивают важность метрик, хотя именно они помогают стать финансово успешным геймдизайнером. Коэффициенты, доли и пропорции помогают узнать, насколько успешной получилась игра. Часто их начинают замерять ещё на этапе прототипов, которые выпускают для небольшой тестовой аудитории. Одни из самых главных метрик по мнению маркетологов, это: количество загрузок, количество добавлений в список желаемого, число ежедневных пользователей, время, проведённое в игре, и как часто игроки возвращаются в игру снова.

Необходимые инструменты:

- идея, задумка, концепция
- креативное мышление
- бумага и карандаши
- набор карточек - магнитов и флипчарт
- уровень, модели, картинки - визуальная составляющая, которую мы видим на экране
- геймплей, игровые механики - процесс игры, действия, которые должен совершать игрок для прохождения.

Примеры механик:

Механики персонажа:

- Умение прятаться: игрок может спрятаться за объектами на уровне, чтобы избежать врагов или пройти незаметно.
- Использование предметов: игрок может собирать и использовать различные предметы, такие как ключи, бомбы, лечебные зелья и т.д.
- Перемещение: игрок может двигаться по уровню вправо и влево, а также подниматься и спускаться по лестницам.
- Прыжок: игрок может прыгать, чтобы перепрыгивать препятствия и достигать высоких платформ.
- Лазанье: игрок может лазить по стенам и другим вертикальным поверхностям, чтобы достичь новых мест на уровне.
- Сбор предметов: игрок может собирать различные предметы, такие как монеты, кристаллы и другие ценности, чтобы получить бонусы и улучшения.
- Атака: игрок может атаковать врагов, используя оружие или способности, чтобы уничтожать их и продвигаться дальше по уровню.
- Использование способностей: игрок может использовать специальные способности, такие как телепортация, создание щита или временное замедление времени, чтобы преодолеть трудности на уровне.

Механики для уровней:

- Двигающиеся объекты, которые могут быть как препятствием, так и помощником в прохождении уровня. Игрок может использовать их для перемещения или прятаться за ними от врагов.
- Также на уровнях можно встретить объекты, за которые можно спрятаться, чтобы избежать опасности. Это может быть куст, ящик или другой предмет, который скроет игрока от врагов.
- Одной из важных механик являются объекты, на которые можно взобраться. Они могут быть как статичными, так и движущимися, и помогают игроку преодолеть препятствия и достичь новых мест.
- Ловушки - это еще один элемент, который делает игру более сложной. Они могут быть различными: шипы, лазеры, ямы и т.д. Игрок должен быть осторожен и избегать их, чтобы не потерять жизни.
- Разрушаемые объекты - это предметы, которые можно разрушить, используя оружие или другие способы. Они могут скрывать секреты или просто помочь игроку пройти уровень.

- Телепортаторы - это объекты, которые перемещают игрока в другую часть уровня. Они могут быть как статичными, так и движущимися, и помогают игроку быстро перемещаться по уровню.
- Объекты, которые можно активировать, могут быть различными: выключатели, кнопки, рычаги и т.д. Они могут открывать двери, активировать механизмы или просто помочь игроку пройти уровень.
- Наконец, объекты, которые можно перемещать, могут быть как статичными, так и движущимися. Игрок может использовать их для создания мостов, преодоления препятствий или просто для того, чтобы достичь новых мест.

Механики боссов:

- Босс с несколькими фазами: босс имеет несколько форм, каждая из которых имеет свои уникальные атаки и слабости. Игрок должен адаптироваться к каждой фазе и использовать различные стратегии, чтобы победить босса.
- Босс с уязвимыми точками: босс имеет несколько уязвимых точек, которые игрок должен атаковать, чтобы нанести урон. Однако, эти точки могут быть защищены или скрыты, поэтому игрок должен использовать свои навыки и интуицию, чтобы найти их.
- Босс с оружием: босс может использовать оружие, которое игрок должен украсть или уничтожить, чтобы победить. Игрок должен быть осторожен и использовать свои навыки, чтобы избежать атак босса и нанести урон.
- Босс с миньонами: босс может иметь несколько миньонов, которые помогают ему в битве. Игрок должен сначала уничтожить миньонов, чтобы сосредоточиться на боссе, или использовать их для своей пользы.
- Босс с ловушками: босс может создавать ловушки, которые игрок должен избегать или уничтожать. Игрок должен быть осторожен и использовать свои навыки, чтобы избежать ловушек и нанести урон боссу.
- Босс с препятствиями: босс может создавать препятствия, которые игрок должен преодолеть, чтобы достичь его. Игрок должен использовать свои навыки и интуицию, чтобы преодолеть препятствия и нанести урон боссу.
- Босс с временными ограничениями: босс может иметь временные ограничения, которые игрок должен использовать в свою пользу. Например, босс может быть уязвим только во время определенных атак или иметь ограниченное время на каждую фазу. Игрок должен использовать свои навыки и интуицию, чтобы использовать эти ограничения и победить босса.
- Босс с неожиданными атаками: босс может использовать неожиданные атаки, которые игрок должен быстро распознать и избежать. Игрок должен быть осторожен и использовать свои навыки, чтобы избежать атак и нанести урон боссу.

1.4 Демонстрация/показ/представление педагогом варианта минимальной рабочей версии

Педагог демонстрирует пример готового прототипа на бумаге (Рис.1, Рис.2)



Рис 1. Пример создания уровня с указанием важных элементов и механик уровня (Вариант 1)



Рис 2. Пример создания уровня с указанием важных элементов и механик уровня (Вариант 2)

Педагог демонстрирует сборку демонстрационного варианта прототипа на флипчарте с использованием карточек – магнитов (Рис.3)



Рис.3 Набор карточек - магнитов

Этап 2. ОСНОВНАЯ ЧАСТЬ

Рекомендованное время: 1 час 25 минут

Задание/Активность:

№ шага	Вид деятельности	Время (мин.)
2.1	Сборка демонстрационного варианта прототипа на флипчарте в группах	15 мин
2.2	Презентация готовых проектов лидерами групп	15 мин
2.3	Анализ допущенных ошибок и корректировка результатов	10 мин
2.4	Рассказ педагога о среде разработки Unity	5 мин
2.5	Демонстрация педагогом работы необходимых инструментов в Unity	15 мин
2.6	Индивидуальная работа обучающихся по переносу созданных прототипов в Unity	25 мин

2.1 Сборка демонстрационного варианта прототипа на флипчарте в группах

Педагог предлагает обучающимся представить, что они геймдизайнеры, которым поручили создать проекты приключенческих игр, востребованных потребителями.

Педагог помогает обучающимся разделиться на 3 команды и самостоятельно создать прототип игры выбрав один из вариантов заданий, указанных ниже:

Варианты заданий для команд

1. "Поиски сокровища"

Уровень 1: Игрок должен пройти через лабиринт, собирая ключи и избегая ловушек, чтобы добраться до сокровища.

Уровень 2: Игрок должен сражаться с пиратами и монстрами, чтобы защитить сокровище.

Уровень 3: Игрок должен сбежать с острова, на котором находится сокровище, на корабле, избегая преследования пиратов.

2. "Путешествие во времени"

Уровень 1: Игрок должен путешествовать во времени, собирая предметы и

решая головоломки, чтобы добраться до следующей эпохи.

Уровень 2: Игрок должен сражаться с врагами и защищать исторические личности, чтобы сохранить правильный ход истории.

Уровень 3: Игрок должен вернуться в настоящее время, избегая временных параллелей и сохраняя изменения, которые он внес в прошлое.

3. "Побег из тюрьмы"

Уровень 1: Игрок должен найти способ выбраться из своей камеры, собирая предметы и решая головоломки.

Уровень 2: Игрок должен избегать охранников и сражаться с другими заключенными, чтобы добраться до выхода.

Уровень 3: Игрок должен сбежать из тюрьмы, избегая преследования полиции и находя способы скрыться.

4. "Поиски пропавшего города"

Уровень 1: Игрок должен исследовать заброшенный город, собирая предметы и решая головоломки, чтобы найти следы пропавшего города.

Уровень 2: Игрок должен сражаться с монстрами и защищать себя от опасностей, которые скрываются в заброшенном городе.

Уровень 3: Игрок должен найти пропавший город и разгадать тайну его исчезновения.

5. "Путешествие по джунглям"

Уровень 1: Игрок должен пройти через джунгли, собирая предметы и избегая опасностей, чтобы добраться до следующей зоны.

Уровень 2: Игрок должен сражаться с дикими животными и защищать себя от опасностей, которые скрываются в джунглях.

Уровень 3: Игрок должен найти выход из джунглей, избегая ловушек и опасностей, которые могут подстеречь его на пути.

6. "Поиски затерянного храма"

Уровень 1: Игрок должен исследовать джунгли, собирая предметы и решая головоломки, чтобы найти следы затерянного храма.

Уровень 2: Игрок должен сражаться с монстрами и защищать себя от опасностей, которые скрываются в джунглях, чтобы добраться до храма.

Уровень 3: Игрок должен найти затерянный храм

Требования к демонстрационному прототипу игры (Приложение 2)

- Целостность: уровень должен быть связным и логически завершенным, чтобы игрок мог понимать, что происходит и что ему нужно делать.
- Баланс: уровень должен быть достаточно сложным, чтобы вызывать интерес и вызывать желание пройти его, но не слишком сложным, чтобы игрок не терял интерес и не бросал игру.
- Прогрессия: уровни должны быть упорядочены по возрастанию сложности, чтобы игрок мог постепенно улучшать свои навыки и достигать новых целей.
- Разнообразие: уровни должны быть разнообразными и интересными, чтобы игрок не устал от однообразия и мог наслаждаться игрой на протяжении всего прохождения.
- Интуитивность: уровень должен быть понятным и интуитивно понятным, чтобы игрок мог быстро понимать, что ему нужно делать и как это сделать.

- Визуальная привлекательность: уровень должен быть визуально привлекательным и иметь хорошо продуманный дизайн, чтобы игрок мог наслаждаться игрой и получать удовольствие от прохождения уровней.

Обучающиеся в группах выбирают лидера и выполняют задания, готовятся к защите концепций с помощью инструкции (памятки)

Педагог оказывает, отвечает на вопросы обучающихся, задает наводящие вопросы для более широкого и понятного раскрытия идеи игры и механик. Вводными вопросами и подсказками помогает детям придумать более сложные механики и структуры уровней

2.2 Презентация готовых проектов лидерами групп

По завершению работы лидеры демонстрируют получившиеся прототипы на флипчартах, объясняя механики, которые были добавлены (Рис. 4). Педагог и обучающиеся задают дополнительные вопросы по концепции, позволяя команде найти слабые стороны и изменить их, либо дополнить.



Рис.4 Примеры прототипов игр, выполненных обучающимися

2.3 Анализ допущенных ошибок и корректировка результатов

Педагог помогает обучающимся сделать анализ своих работ, коллективно выбирается лучшее решение для создания итоговой совместной версии.

Риски этапа:

На данном этапе обучающиеся могут не справиться с заданием по причине недостаточной сформированности развития навыков командной работы, публичного выступления, недостаточного опыта формулировки идей и их визуализации.

Варианты минимизации рисков:

Своевременная педагогическая поддержка в случае затруднений может минимизировать возможные риски.

2.4 Рассказ педагога о среде разработки Unity

Педагог рассказывает о среде разработки Unity, которая позволяет перенести прототип игры в цифровое пространство для дальнейшей работы.

Педагог показывает, как установить программу Unity и знакомит с основными инструментами работы в данной среде.

Обучающиеся слушают и задают вопросы.

Программу необходимо скачать на официальный сайте <https://unity.com/>

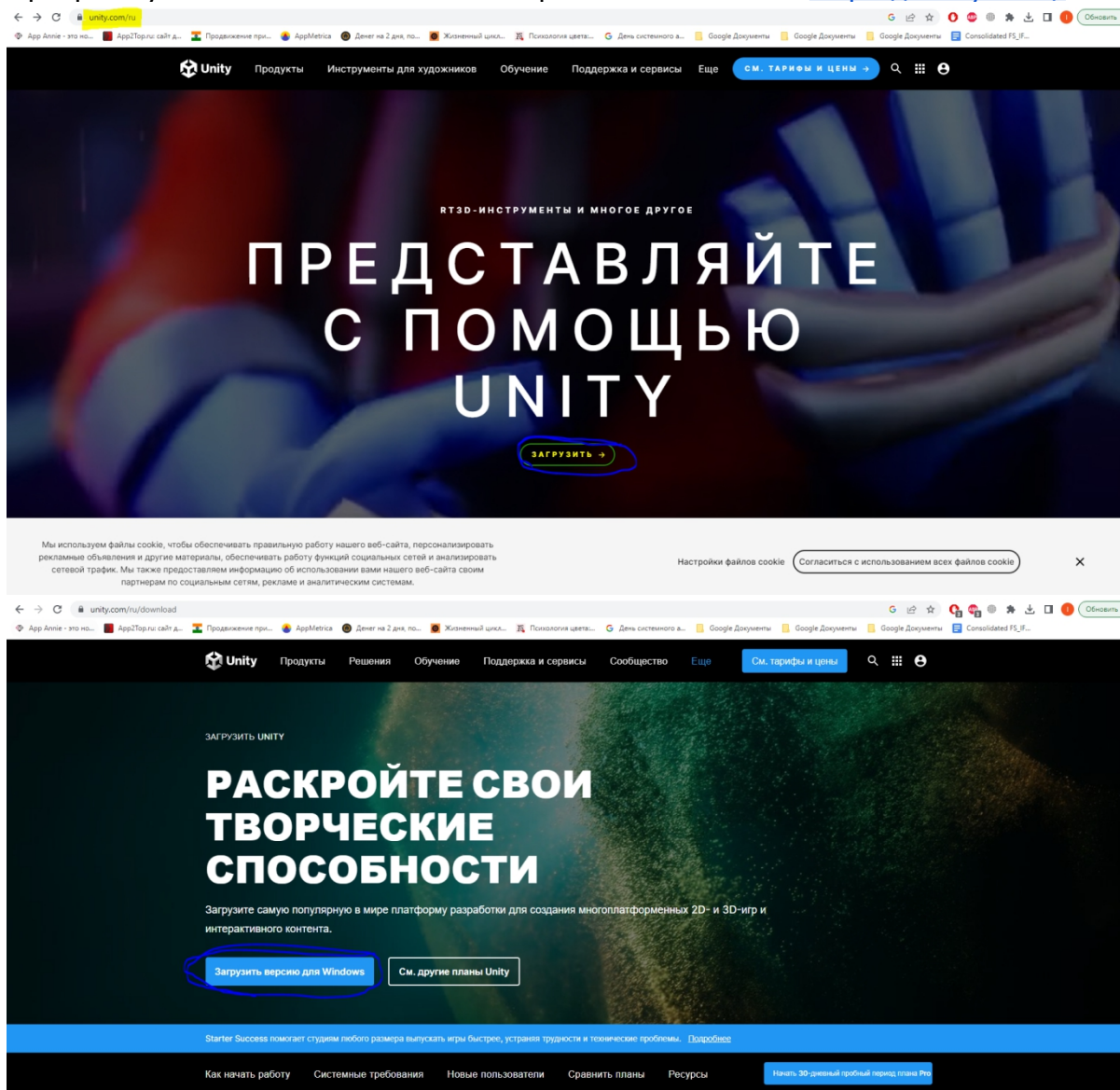


Рис.5 Главный экран официального сайта Unity

Педагог рассказывает, как запустить скачанное приложение.

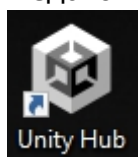


Рис.6 Ярлык для запуска UnityHub

Педагог показывает, как установить необходимую версию Unity, отмечая при этом отличие версий программы. А также объясняет, как создать собственный проект в данной программе.

2.5 Демонстрация педагогом работы необходимых инструментов в Unity

Педагог демонстрирует работу базовых инструментов в Unity: основные рабочие окна программы, их функции, инструменты для работы с объектами на сцене. Знакомит обучающихся с основными приемами дизайна уровней. Настраивает игровую логику и программирует код для перемещения персонажем.

Риски этапа:

Недостаточный опыт работы у педагога с программой Unity.

Варианты минимизации рисков:

Педагогу предлагается Инструкция для педагога по созданию игровых уровней в Unity. (Приложение 1)

2.6 Индивидуальная работа обучающихся по переносу созданных прототипов в Unity

Обучающиеся самостоятельно переносят созданные на флипчартах прототипы в программу Unity, усложняя уровни с помощью инструментов Unity. По окончании работы представляют выполненные прототипы.

Педагог консультирует обучающихся по мере необходимости.

Этап 3. РЕФЛЕКСИЯ/ИТОГИ

Рекомендованное время: 10 минут

Задание/Активность:

№ шага	Вид деятельности	Время (мин.)
3.1	Определение лучших проектов	5 мин
3.2	Рефлексия	5 мин

3.1 Определение лучших проектов

Педагог совместно с обучающимися с помощью открытого голосования определяет лучшие варианты прототипов с присвоением статусов «Идея на миллион» и «Лучший дизайн уровня».

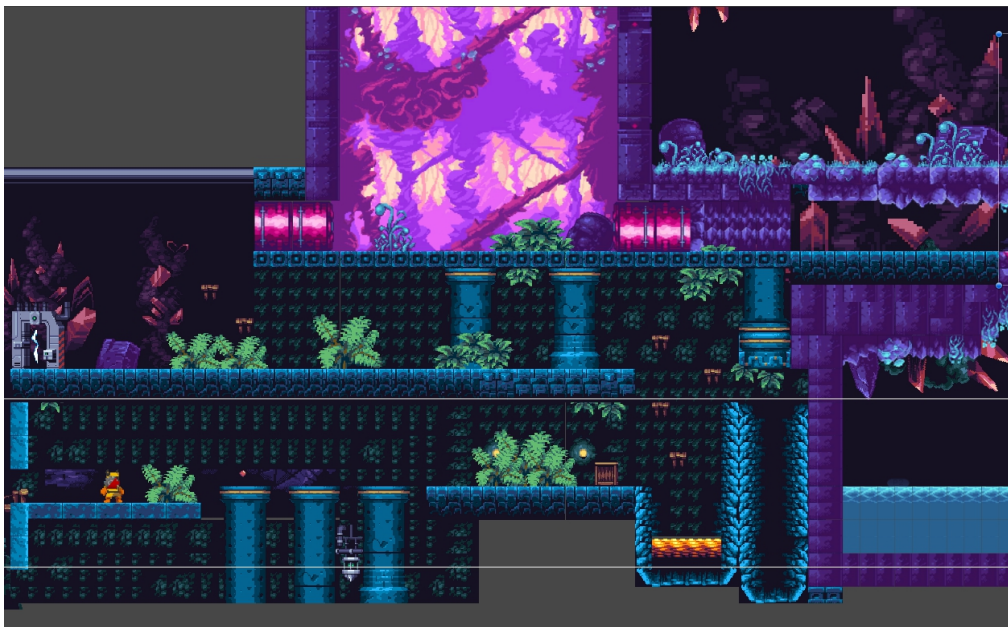


Рис.7 Пример итогового уровня

3.2 Рефлексия

Педагог предлагает обучающимся выбрать один знак препинания в конце темы урока «Создать свою игру – это просто!?».

Обучающиеся поднимают карточки с изображением вопросительных или восклицательных знаков. (Если обучающиеся поднимают восклицательный знак, значит они успешно справились с заданием. Если вопросительные знаки – задание вызвало дополнительный интерес к изучению темы.)

Возможны другие варианты рефлексии (Приложение 3).

Риски этапа:

При определении лучших проектов может возникнуть спорная ситуация среди обучающихся.

Варианты минимизации рисков:

Педагог при выборе лучшего проекта уточняет признаки качественного информационного продукта.

ИСПОЛЬЗОВАННЫЕ РЕСУРСЫ

- для педагога

1. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C#. - М.: Питер, 2018. - 608 с.
2. Паласиос, Хорхе. Unity 5.x. Программирование искусственного интеллекта в играх / Хорхе Паласиос. - М.: ДМК Пресс, 2016. - 849 с.
3. Финни, К. 3D-игры. Все о разработке (+ CD-ROM) / К. Финни. - М.: Бином. Лаборатория знаний, 2011. - 976 с.
4. Финни, К. 3D-игры: Все о разработке (+ CD-ROM) / К. Финни. - М.: Бином. Лаборатория знаний, 2015. - 133 с.
5. Хорхе, Паласиос. Unity 5.x. Программирование искусственного интеллекта в играх. Руководство / Паласиос Хорхе. - М.: ДМК Пресс, 2017.
6. Чехлов Д.А. Визуализация в Autodesk Maya: Mental R Renderer., 2015.-696 с.
7. Вагнер Б. Эффективное программирование на C#. Вильямс, 2017.- 224 с.

8. Вернон В. Предметно-ориентированное проектирование., 2017.- 160 с.
- для обучающихся:
1. Крейтон, Р.Х. Unity Game Development Essentials / Р.Х. Крейтон, PacktPublishing, 2010. - 83 с.
 2. Крёнке, Д Теория и практика построения баз данных: 8-е издание/ Д. Крёнке. - СПб, Питер, 2006, - 542 с.
 3. Мартынов, Н.Н. С# для начинающих / Н.Н. Мартынов. - Москва; Кудиц-пресс, 2007. - 272 с.
 4. Шандров, Б.В. Технические средства автоматизации: учебник для студентов высших учебных заведений/ Шандров Б.В. Москва; "Академия", 2007. - 310 с.
 5. Волков, О.И., Девяткин, О.В. Экономика предприятия фирмы/ О.И. Волков, Москва, издательство инфра Ф 2008, 269с.
 6. Павловская, Т.А. С#. Программирование на языке высокого уровня / Т.А. Павловская - СПб; ПИТЕР, 2009 432 с.
 7. Подбельский, В.В. Язык С#. Базовый курс / Подбельский В.В. СПб, БХВ = Петербург, 2011. - 384 с.
 8. СьюБлэкман, Beginning 3D Game Development with Unity/ СьюБлэкман; Apress, 2011, 992 с.
 9. Фленов, М. Библия С# / Фленов М. СПб.: БХВ - Петербург, 2011. - 560 с.
 10. Фленов, М. С#. Секреты программирования / М. Фленов - СПб, ПИТЕР, 2006. - 457с.
 11. Шилдт, Г. С# Полное руководство / Г. Шилдт И.В. Берштейн Москва., Вильямс, 2011. - 1056 с.
 12. Культин, Н.Б. MicrosoftVisual С# в задачах и примерах / Культин Н.Б. СПб, БХВ-Петербург, 2009, 320 с.
 13. Блог Devby.ru
- для родителей:
1. Шелл Д. Искусство Геймдизайна. Джесси Шелл, 2008.-435 с.

ПРИЛОЖЕНИЕ 1. Рекомендации по созданию игровых уровней в Unity.

После того как мы создали проект нам необходимо познакомиться с базовыми инструментами работы в Unity (Рис.1), необходимо объяснить учащимся значение базовых рабочих окон, какие функции они выполняют, инструменты для перемещения по сцене, какие функции выполняет клавиатура, быстрые клавиши, управление мышью.

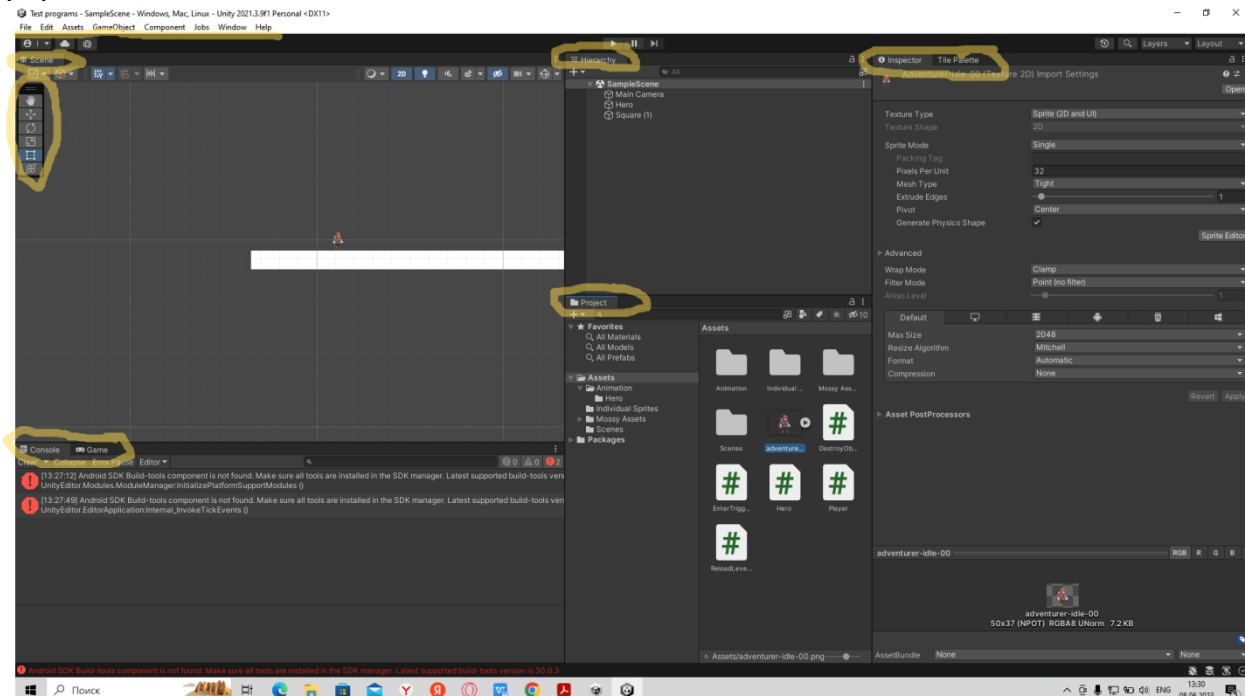


Рис.1

После того как мы познакомились с инструментами работы Unity нам нужно скачать объекты с помощью которых мы перенесем наши прототипы с флипчартов в Unity.

Сделать это можно на сайтах:

<https://itch.io/game-assets>

<https://assetstore.unity.com/>

После того как мы скачали нужные нам объекты их нужно добавить в наш проект, просто перетащив папку с объектами в папку проекта Unity. Либо откры их в Unity через Package Manager - My Assets (Рис.2)

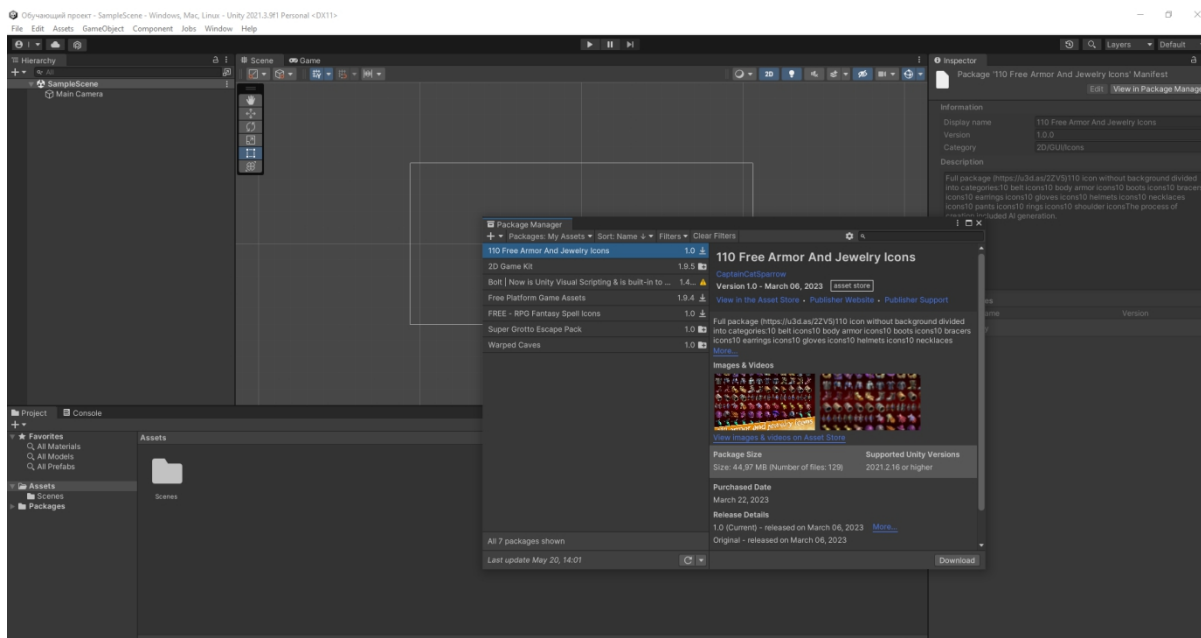


Рис.2

После того как мы добавили нужные объекты в проект нам нужно их отредактировать и подогнать под общую стилистику игры и пропорции

Для этого выбираем нужные объекты в папке проекта и переходим в окно inspector (Рис.3) где можно произвести данные операции

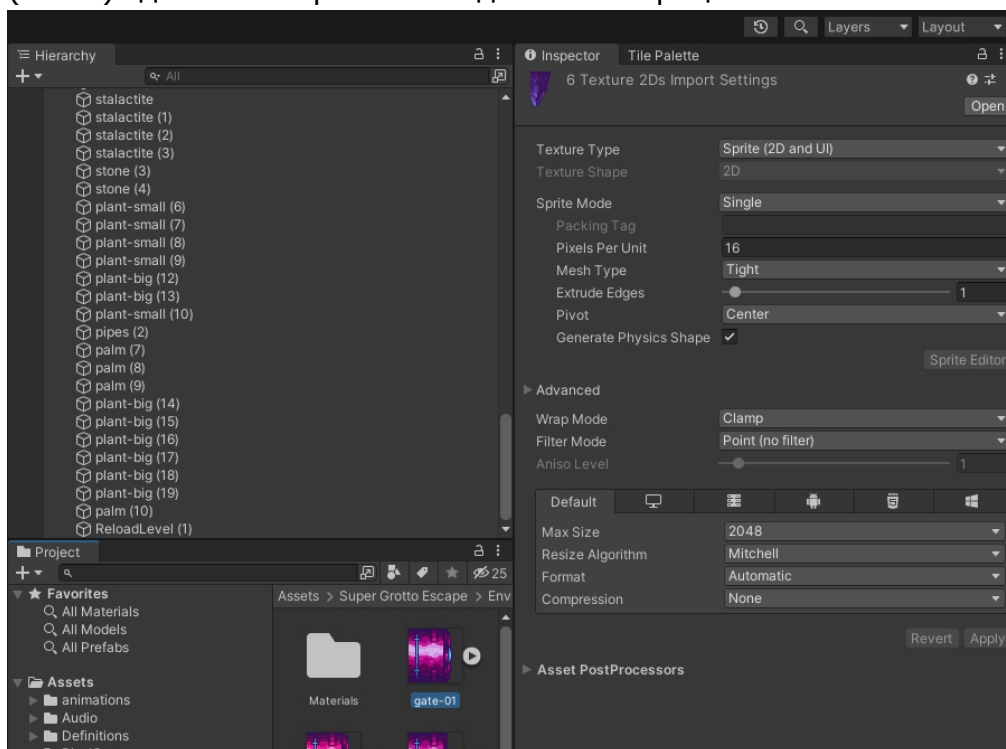


Рис.3

Объекты бывает 2х типов одиночные и комбинированные, если в случае с одиночными мы можем произвести все необходимые изменения в окне Inspector, то с множественными необходимо сделать дополнительную нарезку нажав кнопку Sprite Editor (Рис.4).

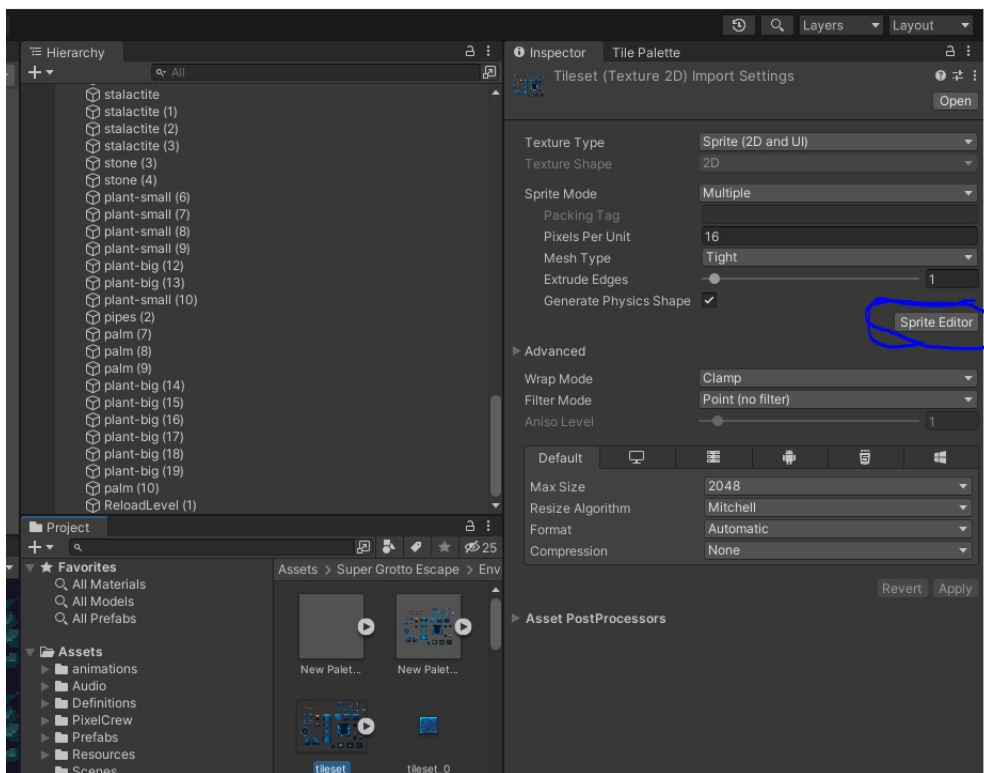


Рис.4

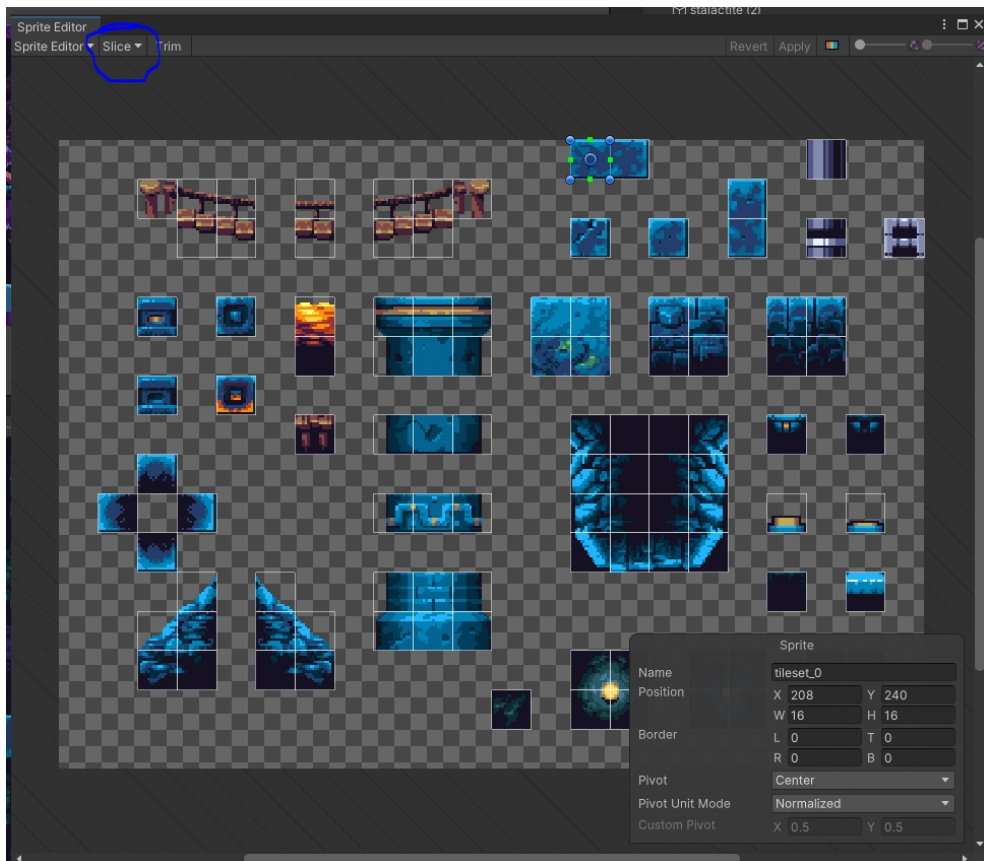


Рис.5

Есть несколько способов нарезки спрайтов (Рис.5), по количеству пикселей, по количеству рядов и колонок и автоматическая нарезка. Нужно выбрать способ необходимый для вашего набора спрайтов (Рис.6).

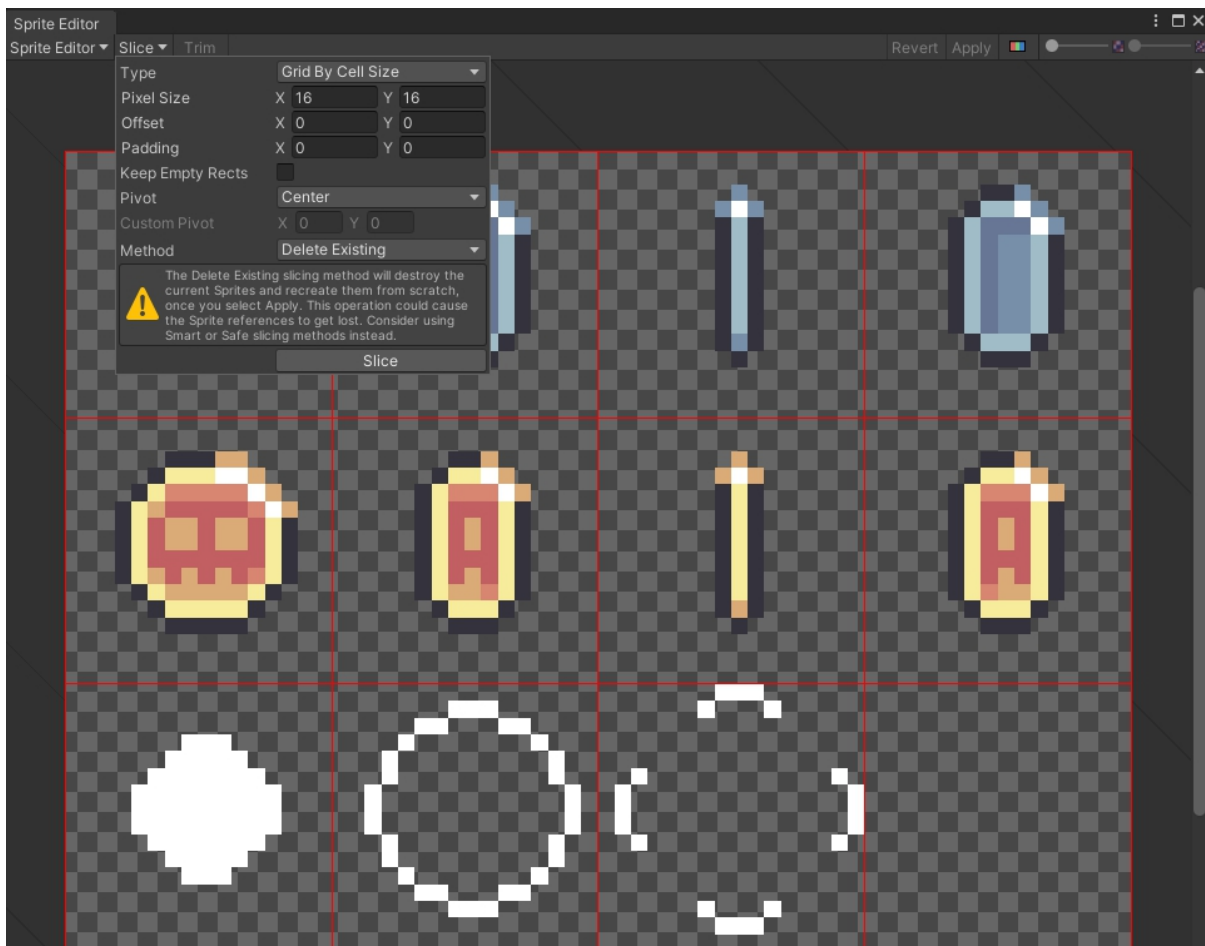


Рис.6

Теперь когда мы подготовили все необходимые объекты можем приступать к сборке уровня для этого нам необходимо добавить рабочее окно Tile Pallet (Рис.7).

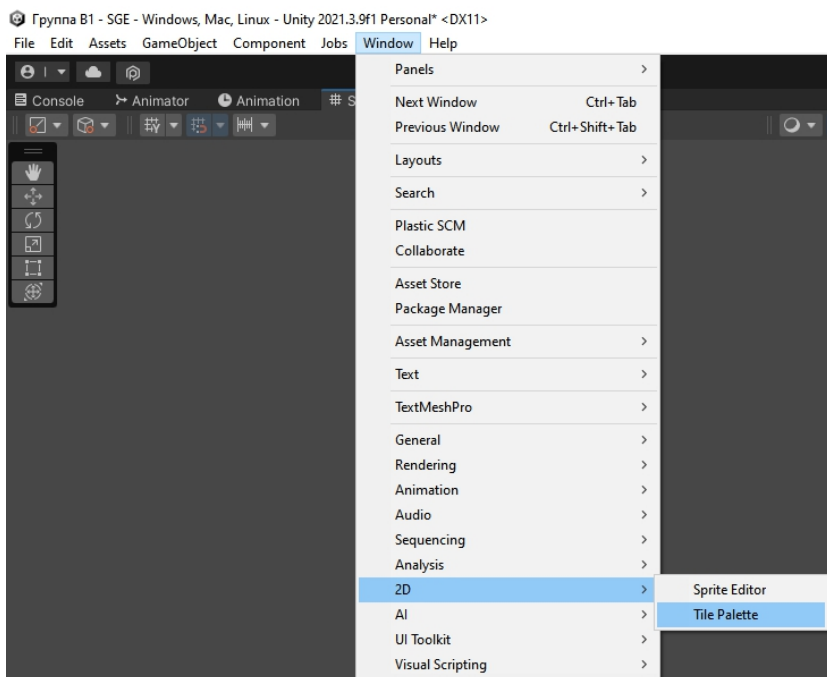


Рис.7

Также нужно добавить сетку Tilemap в окно Hierarchy (Рис.8).

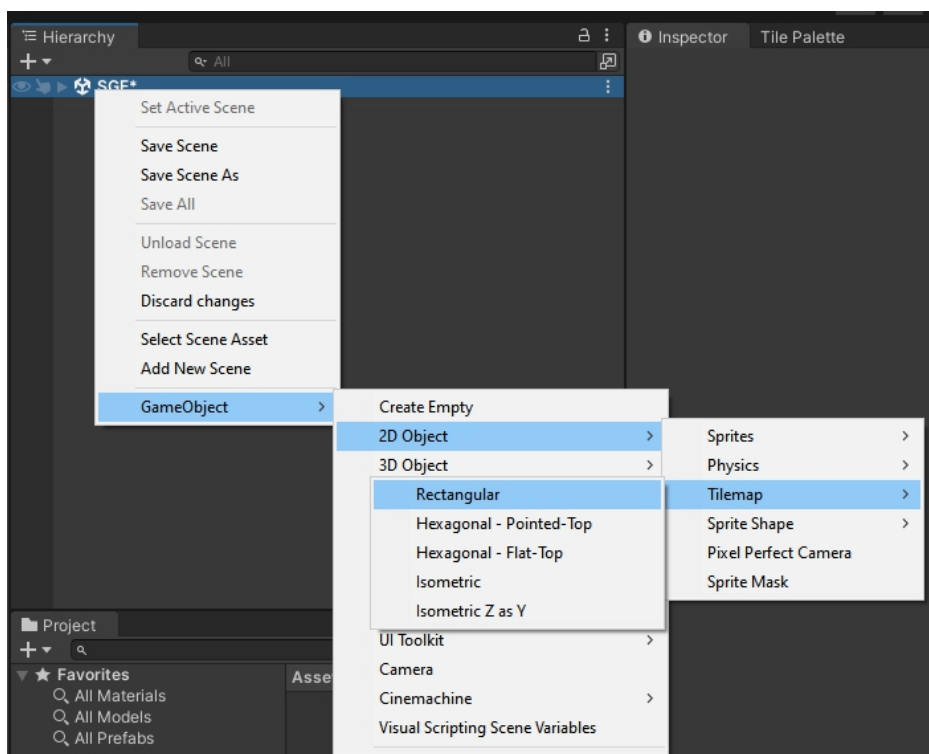


Рис.8

Создаем папку в которой будут находиться тайлы для создания уровня, перетаскиваем сами изображения с помощью нажатия и удержания правой кнопки мыши в окно Tile Palette (Рис.9.1, Рис.9.2.)

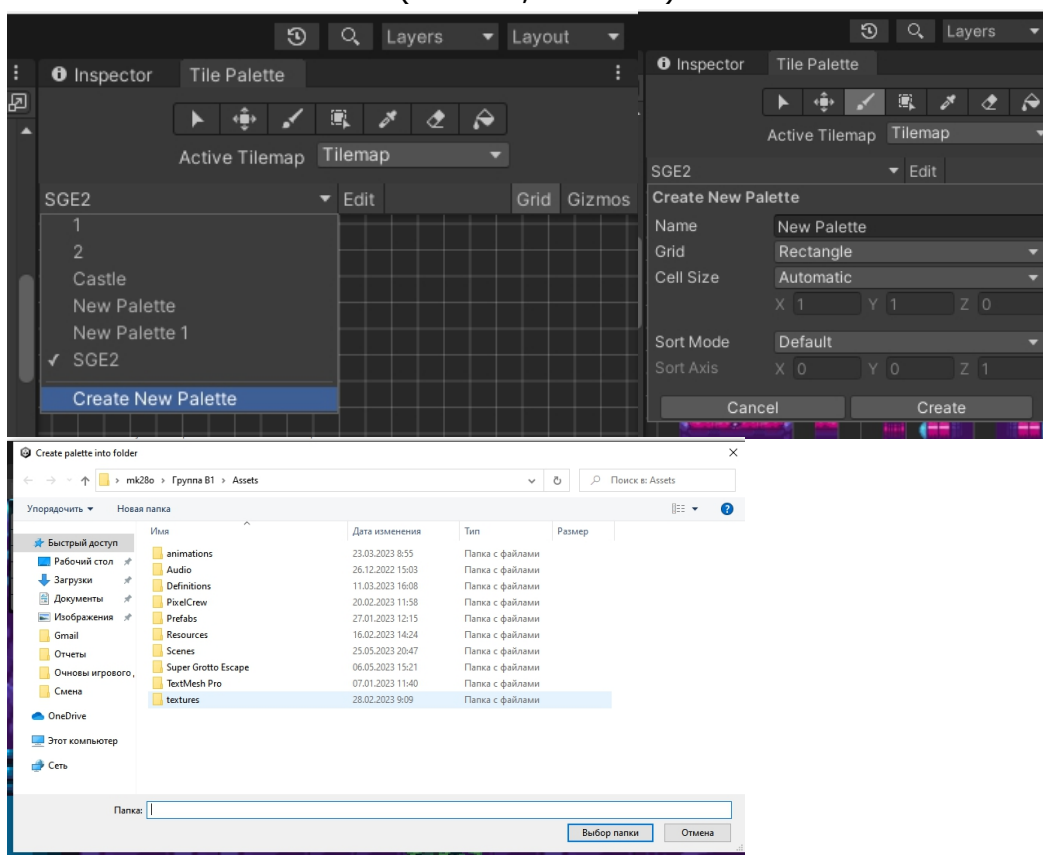


Рис.9.1

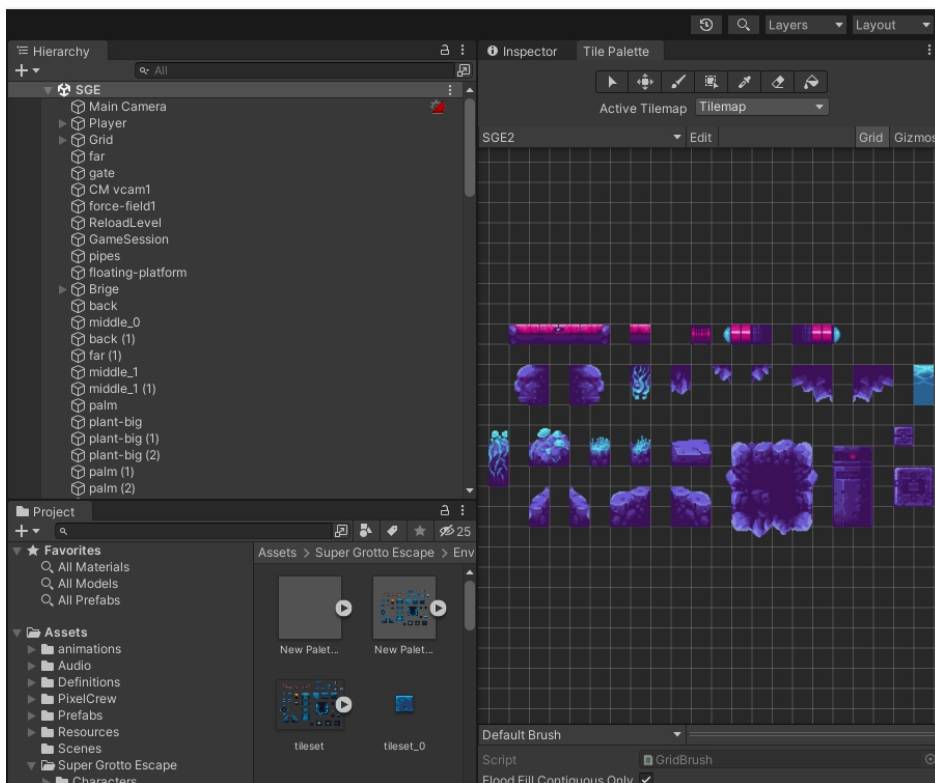


Рис.9.2

Все готово к работе теперь мы можем добавлять объекты на сцену в быстрой и удобной форме используя инструменты Tile Pallet. В окне Tile Pallet есть набор инструментов для быстрого редактирования.

Также чтобы добавить глубины и красочности нашему уровню познакомимся с визуальными слоями в окне Inspector (Рис.10), для этого нужно выбрать объект на сцене

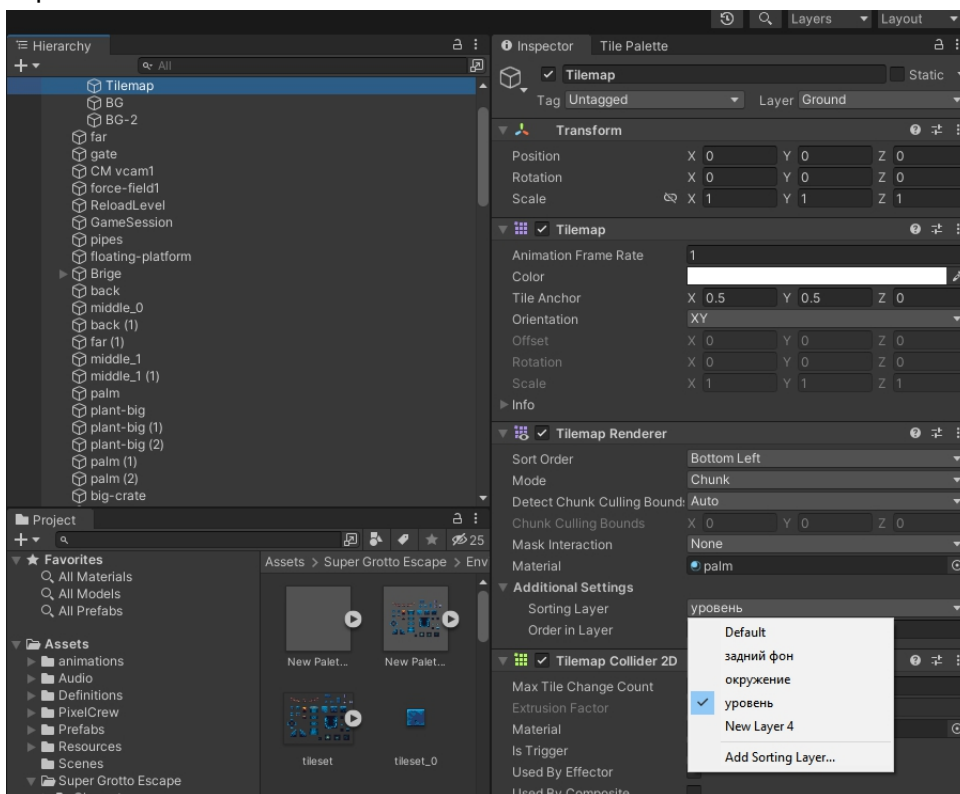


Рис.10

Теперь когда мы познакомились со всеми необходимыми инструментами, можете переносить проекты с флипчартов в Unity.

Советы по созданию уровней:

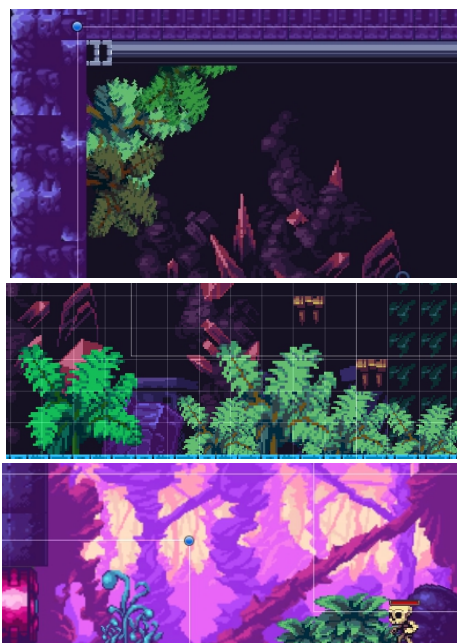
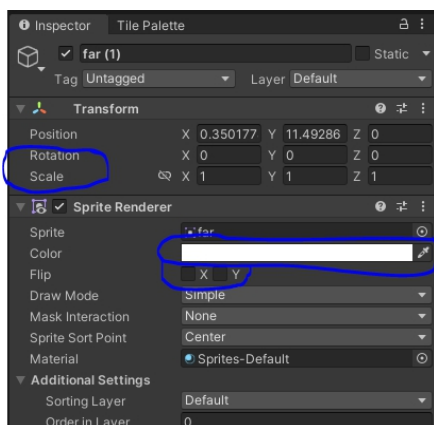
- вы можете создать несколько тайлмапов и задать для них отдельные слои для визуального отображения, это позволит создавать задние и передние слои уровня которые насытят уровень глубиной и отдельными объектами, которые не будут мешать геймплею персонажа



-изменяя цвет объектам в поле color вы можете создавать из одного объекта несколько разных, что дополнительно насытит уровень красками



-в полях Rotation и Scale вы можете менять размеры объектам и разворачивать их под разными углами, что добавить нестандартные варианты использования



Добавляем спрайт персонажа на сцену, устанавливаем ему нужный визуальный слой и добавляем компоненты Capsule Collider (чтобы у персонажа была оболочка с помощью которой он мог взаимодействовать с объектами в Unity) и Rigidbody 2D (добавляет физическое тело, через которое мы сможем реализовать механику

прыжка), с помощью кнопки Add Component в поле Inspector добавляем указанные выше компоненты и создаем код для управления персонажем в окне проекта.

```
using UnityEngine;

public class Player : MonoBehaviour
{
    // Скорость перемещения персонажа
    public float speed = 5f;

    // Сила прыжка персонажа
    public float jumpForce = 10f;

    // Флаг, указывающий, находится ли персонаж на земле
    private bool isGrounded;

    // Ссылка на компонент Rigidbody2D
    private Rigidbody2D rb;

    // Вызывается при запуске игры
    void Start()
    {
        // Получаем ссылку на компонент Rigidbody2D
        rb = GetComponent<Rigidbody2D>();
    }

    // Вызывается каждый кадр
    void FixedUpdate()
    {
        // Получаем ввод от клавиатуры
        float moveHorizontal = Input.GetAxis("Horizontal");
        float moveVertical = Input.GetAxis("Vertical");

        // Вычисляем направление движения
        Vector2 movement = new Vector2(moveHorizontal, moveVertical);

        // Добавляем силу движения к Rigidbody2D
        rb.AddForce(movement * speed);

        // Проверяем, нажата ли кнопка прыжка
        if (Input.GetButtonDown("Jump") && isGrounded)
        {
            // Добавляем силу прыжка к Rigidbody2D
            rb.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);

            // Устанавливаем флаг, указывающий, что персонаж находится на земле
            isGrounded = false;
        }
    }
}
```

```

// Вызывается, когда персонаж сталкивается с другим объектом
void OnCollisionEnter2D(Collision2D collision)
{
    // Если столкновение произошло с объектом, имеющим тег "Ground", то
    // устанавливаем флаг, указывающий, что персонаж находится на земле
    if (collision.gameObject.CompareTag("Ground"))
    {
        isGrounded = true;
    }
}
}

```

Добавляем наш скрипт через кнопку Add Component в поле Inspector
 Настраиваем параметры скорости перемещения персонажем в поле Speed и силу прыжка в поле Jump Force (Рис.1)

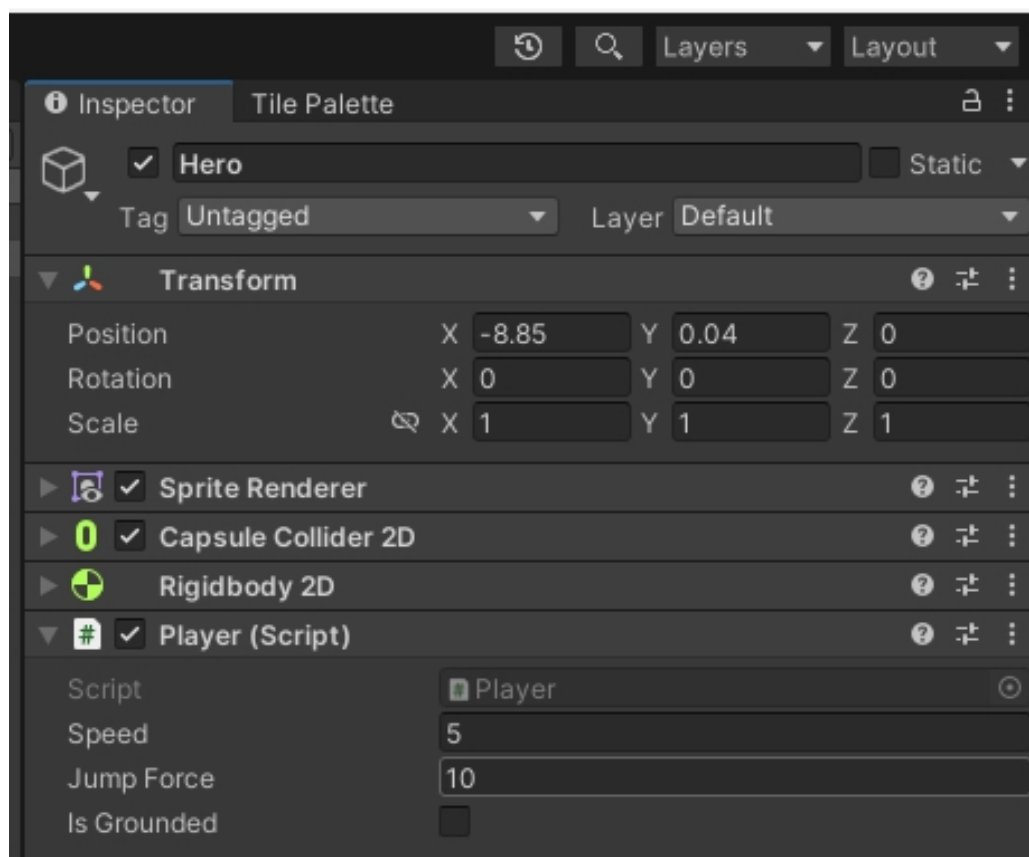


Рис.1

Также нужно добавить Tilemap Collider (Рис.2) для нашего уровня, чтобы персонаж мог по нему перемещаться, также нужно назначить тег Ground, чтобы персонаж мог прыгать на объектах с данным тегом (в коде управлением персонажа мы указали, что он может прыгать когда находится на объекте с тегом "Ground")

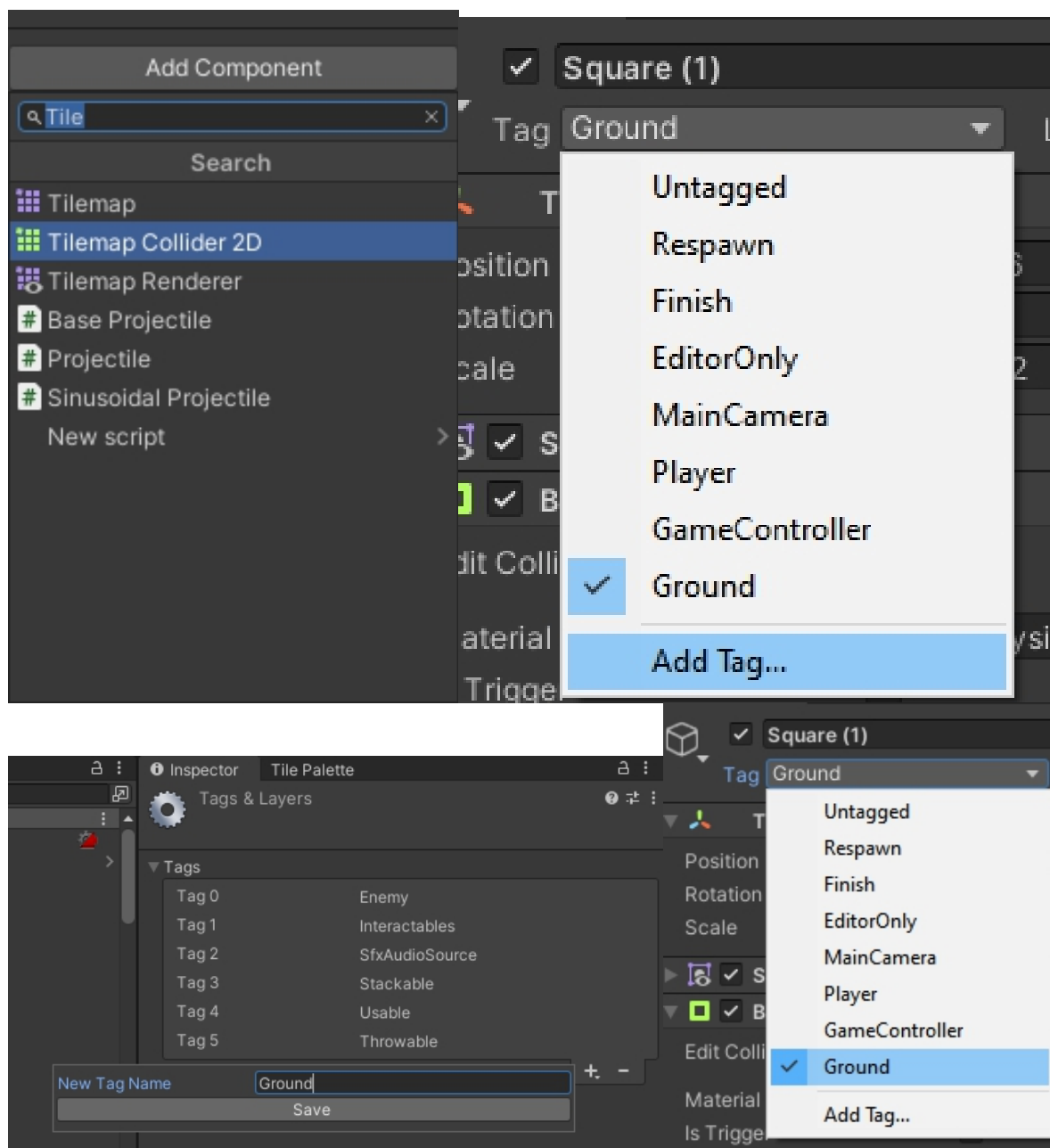


Рис.2

Теперь мы можем запустить игру с помощью клавиши плей, перемещение реализована с помощью клавиш WASD, прыжок назначен на клавишу Space (пробел). Также вы можете скачать дополнительный плагин Cinemachine окне Package Manager во вкладке Package Unity Registry (Рис.3), он позволит добавить камеру которая будет следовать за персонажем.

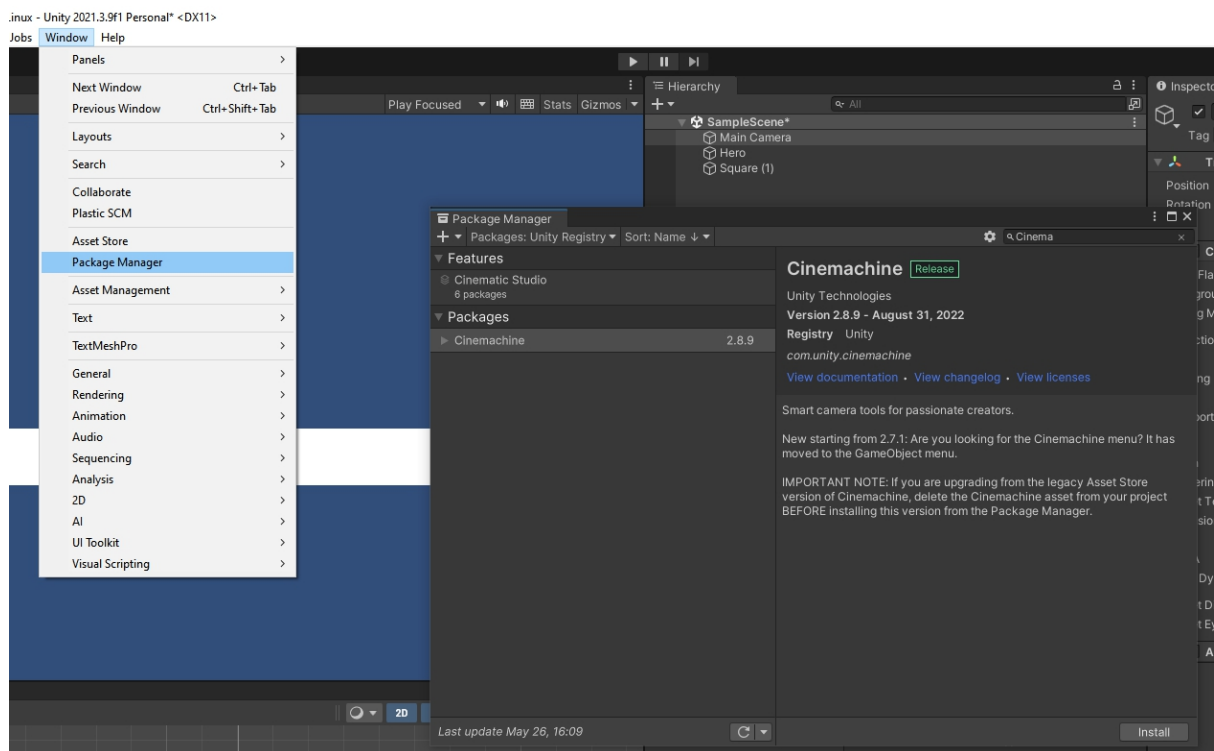


Рис.3

Добавляем в окно Hierarchy - Cinemachine/2D Camera и перетаскиваем объект нашего персонажа из окна Hierarchy в поле Follow (Рис.4) компонента CinemachineVirtualCamera

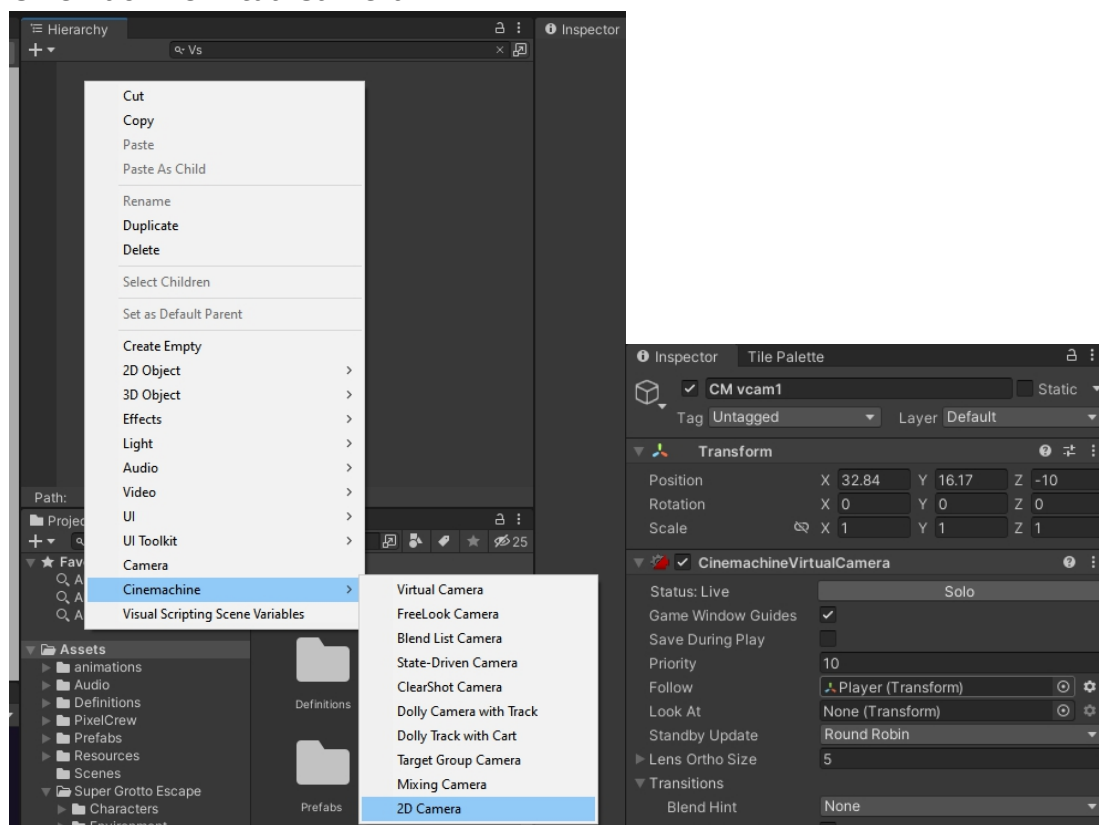


Рис.4

Упражнение «Игры — это хорошо или плохо»

Участников делят на две команды случайным образом. Задача одной команды - написать как можно больше причин, которые позволяют человеку повысить значимость и важность игр в современном мире. Соответственно, задача другой - написать как можно больше причин, из-за которых игры не нужны. «Что вы будете считать в своей группе положительными сторонами игр, а что отрицательными». По желанию каждая команда может подкрепить слова рисунками, сценками о соответствующих поступках. Каждая команда представляет свою тему. Далее идет общее обсуждение, участники соглашаются или нет со списком. В конце ведущий суммирует все сказанное, создается общий список.

Реквизит: листы бумаги А4 на две команды, ручки, цветные карандаши, фломастеры.

Время проведения: (5-7 минут – работа в группах, 10 минут – обсуждение, составление общего списка) 15-17 минут.

Упражнение «Чемодан, корзина, мясорубка»

В конце мастер-класса участникам предлагаются три больших листа, на одном из которых нарисован огромный чемодан, на втором – мусорная корзина и на третьем – мясорубка. На желтом листочке (первый, который приклеивается затем к плакату с изображением чемодана) обучающийся пишет самое важное впечатление, которое он получил после участия в мастер-классе - это то, что он готов забрать с собой и использовать в своей деятельности. На синем листке (второй) участники мастер-класса пишут то, что оказалось ненужным, бесполезным и что можно отправить в «мусорную корзину», т.е. прикрепить ко второму плакату. На сером листе (третий) дети пишут то, что оказалось интересным, но пока не готовым к применению. Таким образом, то, что нужно еще додумать, доработать, отправляется к плакату «мясорубка». Листочки пишутся анонимно и по мере готовности приклеиваются участниками самостоятельно. Далее педагог может прокомментировать результаты используемого приема рефлексии.