

Министерство образования и науки Самарской области

Государственное бюджетное образовательное учреждение
дополнительного образования Самарской области
«Самарский областной центр детско-юношеского технического творчества»

Принята на заседании
Методического Совета
Протокол № 2

от « 20 » июня 2023 г.



Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«Создание игр в среде Unity»

Возраст детей: 12-18 лет

Срок обучения: 1 год

Разработчик:

Яковлев Дмитрий Николаевич
педагог дополнительного образования
первой квалификационной категории

Самара, 2023

Оглавление

Пояснительная записка	3
Учебно-тематический план	6
Содержание	8
Методическое обеспечение	11
Список литературы	14
Приложение Календарно-тематическое планирование	15

Пояснительная записка

Дополнительная общеобразовательная программа «Создание игр в среде Unity» является общеразвивающей программой *технической* направленности. Автор программы - Яковлев Дмитрий Николаевич.

Актуальность программы. Совокупный оборот гейм-индустрии, в которую входят игры и киберспорт, более чем вдвое превышает общий объем мировых кассовых сборов кинотеатров, всего рынка потоковой передачи музыки и продаж альбомов, а также бюджета пяти самых богатых спортивных лиг в мире, вместе взятых.

Технологии, изучаемые в данной программе, наилучшим образом позволяют быстро познакомить учащихся с основами создания мобильных приложений.

- C# – современный, простой и удобный язык программирования. Он универсален, пригоден для создания самых разных программ, от игр до веб-сайтов;

- Unity – одна из наиболее популярных сред, используемых при создании игровых приложений как для десктопов, так и для мобильных платформ.

Педагогическая целесообразность программы заключается в том, что учащиеся приобретут навыки, которые позволят им создавать работающие приложения и станут основой для дальнейшего изучения программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации обучающихся к углубленному изучению программирования, как одной из компьютерных наук. Гибкая структура программы позволит для каждого учащегося сформировать собственный, наиболее для него подходящий путь вхождения в профессию.

Отличительная особенность программы состоит в том, что она в краткой форме знакомит учащихся с основами языка гейм-дизайна. В каждом модуле программы присутствует как теоретическая, так и практическая часть, поэтому учащиеся будут видеть конкретный результат освоения предмета. Особая направленность программы на создание игр позволяет больше заинтересовать учащихся.

Адресат программы - дети от 12 до 18 лет. Наполняемость групп: 5 – 12 человек.

Объем и срок освоения программы. Дополнительная общеобразовательная общеразвивающая программа рассчитана на 1 год обучения.

Режим занятий: 72 академических часа в год, 2 академических часа в неделю.

Форма реализации: очная. Имеется возможность временно проводить занятия дистанционно через платформы для видеосвязи.

В каникулярное время занятия проводятся в соответствии с календарным учебным графиком, допускается изменение форм занятий, проведение воспитательных мероприятий.

Цель программы: познакомить учащихся с современными информационными технологиями посредством изучения языка программирования C# и создания игр в среде Unity.

Задачи:

Обучающие:

- сформировать у учащихся представление об основных этапах разработки игровых приложений;
- сформировать навыки работы в среде Unity;
- познакомить с синтаксисом языка программирования C#;
- способствовать приобретению навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования C#.

Развивающие:

- совершенствовать аналитические навыки;
- формировать навык алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование информации при решении задач;
- развивать умение планировать свои действия с учётом фактора времени;

Воспитательные:

- воспитывать в учащихся усидчивость, аккуратность, умение доводить начатое дело до конца;
- формировать коммуникативные навыки.

Личностные, предметные и метапредметные результаты выполнения программы.

Личностные результаты.

- Формирование навыка доводить дело до конца
- Формирование мировоззрения, необходимого в современном цифровом мире.
- Формирования позитивного отношения к коллегам, их мнению.
- Формирование ответственного отношения к обучению

Предметные результаты.

- Умение разбивать решение задачи на подзадачи, составлять алгоритм;
- Умение объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;

- Умение находить и обрабатывать ошибки в коде;
- Умение писать грамотный, красивый код;
- Умение анализировать как свой, так и чужой код;
- Умение импортировать модули в программу;
- Умение работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- Умение грамотно строить коммуникацию, исходя из целей и ситуации;
- Умение работать с вычислительной техникой.
- Знание основных классических алгоритмов и способы их реализации;
- Знание основных элементы программирования
- Умение создавать оконные приложения.

Метапредметные результаты

- Умение создавать модели для решения практических задач.
- Умение разрешать конфликты на основе выработки общей позиции
- Умение критически оценивать сроки реализации задуманного проекта.
- Умение вносить изменение в проект, корректировать изначальный план
- Умение ставить для себя новые задачи

Учебно-тематический план

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
1	Знакомство со средой Unity	36	18	18
2	Этапы разработки игровых приложений	18	6	12
3	Основы командной разработки приложений.	18	6	12
	Итого:	72	30	42

Модуль 1

№ п/п	Тема, содержание	Общее кол-во часов	Тео рия	Практ ика
	Модуль 1. Знакомство со средой Unity	36	18	18
1.	Установка и настройка Unity. 2D и 3D игры.	2	1	1
2.	Знакомство с основными элементами редактора, интерфейс пользователя.	2	1	1
3.	Структура проекта. Основная сцена.	2	1	1
4.	AssetStore, спрайты. Префабы.	2	1	1
5.	Инспектор объектов. Работа с цветом и формой. Трансформации.	2	1	1
6.	Физика. Rigidbody2D.	2	1	1
7.	Коллайдеры.	2	1	1
8.	Материалы.	2	1	1
9.	Джойнты	2	1	1
10.	TileMap	2	1	1
11.	Шестиугольная и изометрическая проекции	2	1	1
12.	Создание и использование скриптов.	2	1	1
13.	События от мыши, клавиатуры, таймера.	2	1	1
14.	Полезные классы и модули.	2	1	1
15.	Создание и добавление объектов с помощью кода.	2	1	1
16.	Работа с сетью.	2	1	1
17.	Добавление звука.	2	1	1

18.	Знакомство с 3D.	2	1	1
-----	------------------	---	---	---

Модуль 2

№ п/п	Тема, содержание	Общее кол-во часов	Теория	Практика
	Модуль 2. Этапы разработки игровых приложений	18	6	12
1.	Этапы разработки игры.	2	1	1
2.	Дизайн игры и дизайн уровня.	2	1	1
3.	Основы системы Android. Особенности программирования под мобильные устройства.	2	1	1
4.	Инструменты для планирования разработки. Методология Agile.	2	1	1
5.	Разработка интерфейса. Система меню.	2	1	1
6.	Анимации в Unity.	2		2
7.	Навигация. Алгоритмы поиска пути.	2	1	1
8.	Разработка инди-игры «Гонки»	4		4

Модуль 3

№ п/п	Тема, содержание	Общее кол-во часов	Теория	Практика
	Модуль 3. Основы командной разработки приложений.	18	6	12
1.	Командная разработка	2	2	
2.	Инструменты контроля версий. Git.	2	1	1
3.	Знакомство с Github.	2	1	1
4.	GitFlow и GitHubFlow	2	1	1
5.	Выбор темы проекта. Анализ рынка и аналогов. Создание проекта в команде.	6		6
6.	Портирование игры под различные платформы. Размещение в магазинах приложений.	2	1	1
7.	Защита проекта.	2		2

Содержание программы «Разработка игр в среде Unity»

Модуль 1

Знакомство со средой Unity

1. Установка и настройка Unity. 2D и 3D игры.

Теория. Техника безопасности на занятии. Понятие игрового движка. Обзор существующих решений.

Практика. Установка и настройка Unity.

2. Знакомство с основными элементами редактора, интерфейс пользователя.

Теория. Знакомство с панелями и пунктами меню.

Практика. Создание простейшего проекта.

3. Структура проекта. Основная сцена.

Теория. Типичная структура проекта.

Практика. Создание простейшего проекта.

4. AssetStore, спрайты. Префабы.

Теория. Понятие спрайта. Sprite Creator. Sprite Editor. Упаковщик спрайтов. Маски.

Практика. Создание и размещение на сцене спрайтов.

5. Инспектор объектов.

Теория. Работа с цветом и формой. Трансформации.

Практика. Создание объекта произвольной формы и цвета.

6. Физика. Rigidbody2D.

Теория. Двухмерное твердое тело

Практика. Создание и добавление двумерного тела. Гравитация.

7. Коллайдеры.

Теория. Circle Collider, Box Collider, Polygon Collider, Edge Collider

Практика. Работа с разными видами коллайдеров.

8. Материалы.

Теория. Физический 2D материал

Практика. Добавление физического материала. Работа с трением и упругостью.

9. Джойнты

Теория. Distance Joint, Fixed Joint, Friction Joint, Hinge Joint, Relative Joint, Slider Joint, Spring Joint, Target Joint, Wheel Joint

Практика. Связывание объектов.

10. TileMap

Теория. Прямоугольная плитка. Tile Assets, Tile Palette Preferences

Практика. Создание лабиринта

11. Шестиугольная и изометрическая проекции

Теория. Понятие проекции.

Практика. Работа с разными видами отображения карты уровня.

12. Создание и использование скриптов.

Теория. Использование языка программирования с# при создании скриптов в Unity.

Основные функции.

Практика.

13. События от мыши, клавиатуры, таймера.

Теория. Понятие события. Связывание событий с объектами.

Практика. Добавление реакции на нажатия на клавиатуру и мышь. Создание игры “Arcanoid”

14. Полезные классы и модули.

Теория. Object, Transform, Vectors, Quaternion, ScriptableObject, Time, Mathf, Random

Практика. Создание игры с элементами случайности.

15. Создание и добавление объектов с помощью кода.

Теория. Основы проектирования иерархии классов для игры.

Практика. Динамическое создание лабиринтов.

16. Работа с сетью.

Теория. Основы многопользовательских игр. Network Manager.

Практика. Создание простейшей игры для двух человек.

17. Добавление звука.

Теория. Аудио возможности Unity

Практика. Добавление к игре звуков.

18. Знакомство с 3D.

Теория. Камера, освещение. Ориентация. Векторы в трехмерном пространстве.

Практика. Разбор примера Roll-a-Ball из руководства Unity.

Модуль 2

Этапы разработки игровых приложений

1. Этапы разработки игры.

Теория. Идея игры, прототип, написание игрового концепта.

Практика. Мозговой штурм – придумывание темы для игры.

2. Дизайн игры и дизайн уровня.

3. Особенности программирования под мобильные устройства.
Теория. Основы системы Android.
Практика. Создание игры с управлением от датчиков положения.
4. Инструменты для планирования разработки. Методология Agile.
Теория. Гибкие методологии разработки.
Практика. Знакомство с Jira
5. Разработка интерфейса. Система меню.
Теория. Основы UX/UI дизайна.
Практика. Разработка удобного меню для игры.
6. Анимации в Unity.
Теория. Animator Controllers
Практика. Добавление в игру анимации.
7. Навигация. Алгоритмы поиска пути.
Теория. Алгоритмы поиска пути на графе.
Практика. Создание персонажей, находящих дорогу на карте.
8. Разработка инди-игры “Гонки”

Модуль 3

Основы командной разработки приложений.

1. Командная разработка
Теория. Роли, арт. инженеры, дизайнеры, менеджмент, этапы разработки.
Практика. Разбивка на команды и определение ролей.
2. Инструменты контроля версий. Git.
Теория. Понятие коммита и веток
Практика. Установка git, работа с репозиторием.
3. Знакомство с Github.
Теория. Сайты для совместной разработки приложений.
Практика. Добавление проекта на Github. Push изменений.
4. GitFlow и GitHubFlow.
Теория. Понятие пулл-реквеста.
Практика. Совместная работа над одним проектом.
5. Выбор темы проекта. Анализ рынка и аналогов. Создание проекта в команде.
6. Портирование игры под различные платформы. Размещение в магазинах приложений.
7. Защита проекта.

Материально-техническое обеспечение программы

- столы, стулья (по росту и количеству детей);
- демонстрационная магнитная доска;
- технические средства обучения (ТСО) (мультимедийное устройство);
- ноутбуки (10 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО;

Формы подведения итогов реализации программы

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: промежуточную и итоговую аттестацию результатов обучения детей.

Промежуточная аттестация проводится в конце первого полугодия.

Итоговый контроль проводится в конце года с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения.

Аттестация обучающихся может проходить на итоговом занятии в форме презентации своего проекта.

Методы и приемы работы

Особенности организации образовательного процесса: очная форма. Возможна реализация программы в дистанционном формате.

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения, дискуссии);
- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);
- метод проблемного изложения;
- эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);
- исследовательский.

Формы организации образовательного процесса:

- *фронтальная* – подача материала всей учебной группе обучающихся;

- *индивидуальная* – самостоятельная работа обучающихся с оказанием педагогом помощи при возникновении затруднения;
- *групповая* – предоставление учащимся возможности самостоятельно построить свою деятельность, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности.

Формы организации учебного занятия:

- *вводное занятие* – педагог знакомит обучающихся с техникой безопасности, особенностями организации деятельности и предлагаемым планом работы на текущий год;
- *ознакомительное занятие* – педагог знакомит обучающихся с новыми методами работы в зависимости от темы занятия;
- *тематическое занятие* – на котором детям предлагается работать над моделированием по определенной теме. Занятие содействует развитию творческого воображения обучающихся;
- *занятие-проект* – на данном занятии обучающиеся получают полную свободу в выборе направления работы, не ограниченного определенной тематикой. Обучающиеся, участвующие в работе по выполнению предложенного задания, рассказывают о выполненной работе, о ходе выполнения задания, о назначении выполненного проекта;
- *конкурсное игровое занятие* – строится в виде соревнования для повышения активности обучающихся и их коммуникации между собой;
- *комбинированное занятие* – проводится для решения нескольких учебных задач;
- *итоговое занятие* – служит подведению итогов работы за учебный год, может проходить в виде мини-выставок, просмотров творческих работ и презентаций.

Алгоритм учебного занятия:

1. Мотивация обучающихся.
2. Актуализация имеющихся знаний.
3. Теоретический блок нового материала.
4. Закрепление материала.
5. Перерыв.
6. Теоретический блок нового материала.
7. Закрепление материала.

8. Рефлексия.

Контрольно-измерительный блок

Форма	Описание	Критерии оценки
Защита проекта	Игра для платформы Android, Написанная на Unity.	<ul style="list-style-type: none">• Структурная сложность• Качество кода• Оригинальность идеи• Графика• Работоспособность

Список литературы

1. Прайс Марк. С# 7 и .NET Core. Кросс-платформенная разработка для профессионалов. Издательство Питер, 2018г.
2. Гибсон Бонд. Unity и С#. Геймдев от идеи до реализации. Издательство Питер, 2019г.
3. Скит Джон. С# для профессионалов. Тонкости программирования. Издательство Вильямс, 2019г.
4. Хокинг Джозеф. Unity в действии. Мультиплатформенная разработка на С#.
5. <https://docs.unity3d.com/Manual/index.html>

Приложение 1

Календарно-тематический план

№	Тема занятия	Продолжительность	Дата
1	Установка и настройка Unity. 2D и 3D игры.	2	
2	Знакомство с основными элементами редактора, интерфейс пользователя.	2	
3	Структура проекта. Основная сцена.	2	
4	AssetStore, спрайты. Префабы.	2	
5	Инспектор объектов. Работа с цветом и формой. Трансформации.	2	
6	Физика. Rigidbody2D.	2	
7	Коллайдеры.	2	
8	Материалы.	2	
9	Джойнты	2	
10	TileMap	2	
11	Шестиугольная и изометрическая проекции	2	
12	Создание и использование скриптов.	2	
13	События от мыши, клавиатуры, таймера.	2	
14	Полезные классы и модули.	2	
15	Создание и добавление объектов с помощью кода.	2	
16	Работа с сетью.	2	
17	Добавление звука.	2	
18	Знакомство с 3D.	2	
19	Этапы разработки игры.	2	
20	Дизайн игры и дизайн уровня.	2	
21	Основы системы Android. Особенности программирования под мобильные устройства.	2	
22	Инструменты для планирования разработки. Методология Agile.	2	
23	Разработка интерфейса. Система меню.	2	
24	Анимации в Unity.	2	
25	Навигация. Алгоритмы поиска пути.	2	
26	Разработка инди-игры "Гонки"	2	
27	Разработка инди-игры "Гонки"	2	
28	Командная разработка.	2	
29	Инструменты контроля версий. Git.	2	
30	Знакомство с Github.	2	
31	GitFlow и GitHubFlow	2	

32	Выбор темы проекта. Анализ рынка и аналогов. Создание проекта в команде.	2	
33	Выбор темы проекта. Анализ рынка и аналогов. Создание проекта в команде.	2	
34	Выбор темы проекта. Анализ рынка и аналогов. Создание проекта в команде.	2	
35	Портирование игры под различные платформы. Размещение в магазинах приложений.	2	
36	Защита проекта.	2	
	Итого	72	