



МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГОРОДА НОВОСИБИРСКА

**ЦЕНТР ОБРАЗОВАНИЯ № 82
«РАЗВИТИЕ»**

Принято на заседании
Педагогического совета
Протокол № 1 от 26.08.2026 г.

«УТВЕРЖДЕНО»

Приказ от 26.08.2026 г. № 67-од

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ: СОЗДАНИЕ ИГР И
ПРИЛОЖЕНИЙ 2D/3D»**

технической направленности

разноуровневая: базовый и продвинутый уровни

Возраст обучающихся: 15–18 лет (9–11 классы)

Срок реализации: 2 года

Объем программы: 144 академических часа

Разработчик:

Дворядкин Андрей Алексеевич,
учитель информатики и программирования,
разработчик образовательной платформы
и среды разработки AtmoCode Studio

Новосибирск, 2026

Лист внутренней экспертизы

Внутренняя экспертиза проведена. Программа рекомендована к рассмотрению на педагогическом совете учреждения.

Эксперт	Методист _____
Заключение	Программа соответствует структуре дополнительной общеобразовательной общеразвивающей программы и рекомендована к реализации.
Подпись	_____ / _____
Дата	«____» _____ 2026 г.
Примечание. Реквизиты протокола, приказа, сведения об эксперте и конкретные даты реализации уточняются и вносятся учреждением при утверждении программы.	

Паспорт программы «Разработка игр, приложений и интерактивных 2D/3D систем»

для размещения в ГИС «Навигатор дополнительного образования Новосибирской области»

№	Параметр	Сведения
1	Полное наименование программы	Дополнительная общеобразовательная общеразвивающая программа технической направленности «Разработка игр, приложений и интерактивных 2D/3D систем».
2	Публичное наименование	ДООП «Разработка игр, приложений и интерактивных 2D/3D систем» (AtmoCode Studio, 9–11 классы).
3	Краткое описание	Двухгодичная программа проектной разработки на языке microScript 2. Обучающиеся создают игры, приложения, симуляторы и интерактивные 2D/3D системы; осваивают архитектуру, алгоритмы, данные, Phaser, Arcade Physics и Box2D, интерфейсы, мультимедиа, micro 3D/Babylon, тестирование, оптимизацию, документацию и командный выпуск продукта.
4	Направленность	Техническая.
5	Уровень	Разноуровневая: 1-й год — базовый инженерный уровень; 2-й год — продвинутый проектный уровень.
6	Адресат	Обучающиеся 15–18 лет, 9–11 классы. Желательны базовые знания алгоритмов, переменных, условий, циклов и функций.
7	Срок и объём	2 года, 144 академических часа: 72 часа в первый год и 72 часа во второй год.
8	Форма обучения	Очная, групповая. Индивидуальная, парная и командная проектная работа; электронные ресурсы используются в соответствии с локальными актами учреждения.
9	Режим занятий	1 раз в неделю по 2 академических часа; 36 учебных недель в год; перерыв между академическими часами — 10 минут.
10	Цель	Формирование готовности старшеклассников самостоятельно вести разработку цифрового продукта от постановки задачи и модели до опубликованной 2D/3D системы, документации и технической защиты.
11	Ключевой результат	Обучающийся создаёт индивидуальный MVP и в команде выпускает игру, приложение или симулятор с обоснованной архитектурой, данными, интерфейсом, физикой или 3D-сценой, тестами, документацией и публичной защитой.
12	Особые условия	Набор на первый год свободный после входной диагностики. Поступление сразу на второй год возможно по портфолио и практической работе. 3D-модуль использует стабильные возможности micro 3D/Babylon текущей версии Studio; расширения осваиваются по мере готовности.
13	Разработчик и педагог	Дворядкин Андрей Алексеевич — учитель информатики и программирования, разработчик образовательной платформы и среды разработки AtmoCode Studio.
14	Материально-техническая база	Компьютерный кабинет; 10–12 компьютеров с аппаратным ускорением WebGL; устройство педагога; проектор; сеть и Интернет; актуальный браузер; наушники; AtmoCode Studio; средства проектирования и технической документации.
15	Итоговая форма	Демонстрационный день в формате хакатона/геймджема: презентация продукта, технический разбор, проверка сценариев, ответы на вопросы и ретроспектива.

Содержание программы

Раздел	Наименование
Раздел 1	Комплекс основных характеристик программы
1.1	Пояснительная записка
1.2	Цель и задачи программы
1.3	Планируемые результаты
1.4	Содержание программы и учебные планы
1.5	Система контроля и аттестации
Раздел 2	Рабочая программа воспитания
2.1	Цель, задачи и целевые ориентиры воспитания
2.2	Формы и методы воспитания
2.3	Условия организации воспитания
2.4	Календарный план воспитательной работы
Раздел 3	Комплекс организационно-педагогических условий
3.1	Календарный учебный график
3.2	Условия реализации программы
3.3	Методическое и информационное обеспечение
Раздел 4	Приложения к программе
4.1	Календарно-тематический план первого года
4.2	Календарно-тематический план второго года
4.3	Оценочные материалы
4.4	Паспорт итогового проекта и лист самооценки

Раздел 1. Комплекс основных характеристик программы

1.1. Пояснительная записка

Направленность программы. Программа «Разработка игр, приложений и интерактивных 2D/3D систем» имеет техническую направленность и объединяет программирование, разработку интерактивных систем, цифровое моделирование, дизайн интерфейсов и инженерную проектную деятельность.

Концепция программы. В 9–11 классах игра и приложение рассматриваются как программная система: игровой цикл — архитектурный паттерн; сцены и экраны — конечный автомат; карта — источник данных и сущностей; UI — контракт с пользователем; физика — вычислительная модель; 3D-сцена — система координат, камеры, геометрии, материалов и света. AtmoCode Studio служит лабораторией полного цикла от прототипа до публикации.

Актуальность

Старшекласснику важно не только написать работающий код, но и поставить задачу, выбрать модель, оценить ограничения, организовать данные, проверить качество и объяснить решение. Курс формирует практическую цифровую грамотность и инженерную культуру, создаёт портфолио для конференций, НТИ, олимпиад проектного типа, хакатонов, геймджемов и дальнейшего профессионального образования.

Содержание согласовано с ориентирами ФГОС основного и среднего общего образования в части проектной, исследовательской, информационной и коммуникативной деятельности, а также с Концепцией развития дополнительного образования детей до 2030 года. Программа расширяет школьную информатику практикой жизненного цикла программного продукта.

Отличительные особенности и новизна

- Метафорически-технический подход: знакомая игровая механика переводится в точные понятия архитектуры, данных, событий, алгоритмов и вычислительной модели.
- Единый 2D-стек: Canvas/screen, screen.ui, карты, entities, storage/data, Phaser, Arcade Physics, Box2D, tweens, particles, lighting, filters, curves и audio API.
- Практическая 3D-линия: micro 3D/Babylon, координатные системы, камера, примитивы и модели, материалы, текстуры, свет, туман, тени и интерактивная сцена.
- Два законченных продукта: индивидуальный MVP первого года и командный выпускной проект второго года.
- Инженерные артефакты: техническое задание, схема архитектуры, модель данных, backlog, журнал дефектов, инструкция, страница проекта и материалы защиты.
- Межпредметные симуляторы по алгебре, геометрии и физике с параметрами, измерениями, графиками и проверяемой гипотезой.
- Профильная дифференциация: разработка, game design, 2D/3D-графика, звук, UX, тестирование, аналитика и управление проектом.
- Стабильное ядро и перспективный слой: обязательные результаты опираются на доступные API, новые возможности Babylon включаются после проверки педагогом.

Адресат и возрастные особенности

Программа предназначена для обучающихся 15–18 лет (9–11 классы), группа 10–12 человек. Старшеклассники способны работать с абстрактными моделями и длительным проектом, нуждаются в автономии, профессиональной пробе и содержательной оценке результата. Поэтому педагог задаёт инженерные ограничения и контрольные точки, сохраняя выбор темы, роли и технического решения.

Условия приёма и перехода между уровнями

Входная диагностика проверяет чтение простого кода, переменные, условия, циклы, функции, координаты и поиск ошибки. Недостающие навыки компенсируются стартовым модулем. Для второго года требуется завершение первого года либо портфолио и практическая диагностика архитектуры, данных, интерфейса и тестирования.

Объём, срок, форма и режим

Срок реализации	2 года
Общий объём	144 академических часа
Первый год	72 часа: базовый инженерный уровень
Второй год	72 часа: продвинутый проектный уровень
Форма	очная; индивидуальная, парная и командная работа
Режим	1 занятие в неделю по 2 академических часа, 36 недель в год
Продолжительность часа	40 минут; перерыв 10 минут
Язык	русский
Модель реализации	модульно-проектная, спиральная, разноуровневая

Формы занятий

- инженерный практикум и разбор API
- лабораторная работа и вычислительный эксперимент
- парное программирование и code review
- архитектурная сессия и проектный спринт
- прототипирование UI/UX и пользовательское тестирование
- хакатон, геймджем и техническая репетиция
- демонстрация, защита и ретроспектива

Здоровьесбережение. Для 9 класса непрерывная работа с персональным компьютером планируется не более 30 минут, для 10–11 классов — не более 35 минут за один рабочий отрезок. Экранные этапы чередуются с проектированием на бумаге, обсуждением, гимнастикой для глаз и двигательной паузой; режим уточняется по действующим санитарным нормам.

1.2. Цель и задачи программы

Цель программы: формирование готовности старшеклассников самостоятельно вести разработку игры, приложения или интерактивной 2D/3D системы от постановки задачи и построения модели до тестирования, публикации, документации и технической защиты.

Предметные задачи

Базовый инженерный уровень (первый год):

- систематизировать microScript 2: функции, списки, объекты, события, модули, стиль кода и обработка ошибок;
- освоить game loop, конечные автоматы, сцены, сущности, data-driven карты и хранение состояния;
- проектировать многоэкранный UI, HUD, формы, настройки, доступность и пользовательскую обратную связь;

- применять Phaser Arcade Physics и Box2D для тел, форм, контактов, запросов, joints и управляемых физических моделей;
- использовать графический и звуковой редакторы, audio API, tweens, particles, lighting, filters, camera effects и curves;
- строить цифровой эксперимент по математике или физике и анализировать влияние параметров;
- освоить основы micro 3D/Babylon: сцена, камера, геометрия, трансформации, материал, текстура и свет;
- создать, протестировать, документировать и защитить индивидуальный MVP;

Продвинутый проектный уровень (второй год):

- формировать требования, архитектуру, модель данных и план поставки командного продукта;
- углубить 2D-физику: Box2D shapes, joints, contacts, queries, tilemap и terrain;
- создавать 3D-сцены с камерой, иерархией объектов, материалами, текстурами, освещением, туманом и тенями;
- реализовывать взаимодействие в 3D, выбор объекта, управление, интерфейс и хранение состояния;
- проектировать игровые подсистемы: поведение, инвентарь, прогресс, диалоги, экономика, настройки и сохранение;
- проводить профилирование, оптимизацию, кросс-устройственное и пользовательское тестирование;
- вести backlog, версии, code review, интеграцию, документацию и публичный выпуск;
- подготовить проект к школьной конференции, олимпиаде НТИ, хакатону или геймджему;

Метапредметные задачи

- переводить проблему в требования, ограничения, критерии приёмки и проверяемую модель;
- декомпозировать систему, сравнивать технические альтернативы и обосновывать выбор;
- выдвигать гипотезу, управлять параметрами эксперимента, интерпретировать данные и ограничения модели;
- искать и критически оценивать документацию, примеры, лицензии и источники данных;
- планировать командную работу, согласовывать интерфейсы модулей, фиксировать решения и риски;
- создавать понятные схемы, технические тексты, демонстрации и выступления для разной аудитории;
- использовать рефлекссию, метрики качества, журнал дефектов и обратную связь для улучшения продукта;

Личностные и воспитательные задачи

- развивать самостоятельность, настойчивость и ответственность за инженерное решение
- формировать уважение к авторству, данным, безопасности и этическим последствиям технологии
- поддерживать профессиональное самоопределение и реалистичную оценку собственных компетенций
- воспитывать культуру сотрудничества, аргументированного несогласия и признания вклада
- ориентировать цифровое творчество на пользу пользователю, школе и обществу

1.3. Планируемые результаты

Личностные результаты

- самостоятельно ставит образовательную и проектную цель, доводит работу до проверяемого результата
- несёт ответственность за качество, сроки, безопасность и индивидуальный вклад
- соблюдает авторское право, правила работы с данными и этику публикации
- принимает экспертную обратную связь и аргументированно пересматривает решение
- учитывает доступность, общественную пользу и риски цифрового продукта
- соотносит опыт проекта с направлениями дальнейшего образования и профессиями

Метапредметные результаты

Регулятивные:

- формулирует требования, ограничения, критерии готовности и риски
- строит roadmap/backlog, оценивает ресурсы и корректирует план
- организует версии, контрольные точки, тестирование и ретроспективу

Познавательные:

- строит формальную или вычислительную модель процесса
- выделяет сущности, данные, состояния, события и зависимости
- сравнивает алгоритмы и архитектурные решения по критериям
- проводит эксперимент, анализирует данные, погрешность и границы применимости

Коммуникативные:

- согласует роли и контракты между модулями
- проводит code review и пользовательское тестирование без перехода на личность
- ведёт техническую документацию и защищает решение перед аудиторией

Информационные:

- работает с документацией и примерами API
- оценивает источник, лицензию и допустимость использования ассета
- структурирует данные проекта и безопасно управляет публикацией

Предметные результаты первого года

- пишет структурированный код microScript 2 и объясняет архитектуру проекта;
- реализует сцены, состояния, карты, entities и data-driven параметры;
- создаёт UI/HUD, настройки, сохранение и обработку некорректных данных;
- применяет Phaser Arcade Physics и базовые возможности Box2D;
- создаёт мультимедийную систему из графики, анимации, звука и эффектов;
- строит цифровую модель движения или другого процесса, проводит серию экспериментов;
- создаёт базовую 3D-сцену micro 3D/Babylon с камерой, объектами, материалом и светом;
- выпускает индивидуальный MVP с тестовым отчётом и документацией;

Предметные результаты второго года

- проектирует модульную архитектуру и модель данных командного продукта;
- использует расширенные Box2D-тела, shapes, joints, contacts, queries, карту или terrain;
- строит 3D-сцену, управляет камерой, трансформациями, материалами, текстурами, светом, туманом и тенями;
- реализует интерактивность, интерфейс и состояние 3D-приложения;
- разрабатывает связанные подсистемы поведения, прогресса, экономики, диалогов или симуляции;
- профилирует загрузку и отрисовку, оптимизирует ресурсы и проверяет разные устройства;

- ведёт backlog, журнал дефектов, техническое описание, инструкцию и страницу проекта;
- представляет устойчивую демонстрацию и аргументированно отвечает на технические вопросы;

Метапредметный результат курса. Выпускник способен вести инженерный цикл: определить проблему и пользователя, построить модель, выбрать архитектуру и технологии, создать прототип, проверить решение данными и тестами, организовать командный выпуск и публично обосновать результат.

1.4. Содержание программы и учебные планы

Учебный план первого года — базовый инженерный уровень

№	Модуль	Всего	Теория	Практика	Продукт
1	Инженерный старт microScript 2	6	2	4	Технический прототип
2	Game loop, состояния и архитектура	8	2	6	Многоэкранная система
3	Карты, сущности и данные	8	2	6	Data-driven приключение
4	UI-паттерны, профиль и storage	8	2	6	Прикладной интерфейс
5	Phaser Arcade Physics и Box2D	10	3	7	Физический прототип
6	Графика, звук и визуальные эффекты	8	2	6	Мультимедийная сцена
7	Цифровые модели: математика и физика	8	3	5	Исследовательский симулятор
8	Введение в micro 3D/Babylon	6	2	4	Интерактивная 3D-сцена
9	Индивидуальный MVP	10	2	8	Игра / приложение / модель
	Итого	72	20	52	

Содержание первого года. Стартовый модуль выравнивает знания microScript 2 и вводит культуру кода. Затем обучающиеся проектируют игровой цикл, состояния и сцены; связывают карты, сущности и параметры; создают многоэкранные интерфейсы, профиль и сохранение. В физическом модуле сравниваются Arcade Physics и Box2D, тела, столкновения, контакты и управляемое движение.

Мультимедийный модуль объединяет авторскую графику, анимацию, audio API, tweens, particles, lighting, filters, камеру и кривые. Цифровая модель связывает программирование с функциями, векторами, скоростью, ускорением, силами и измерениями. Введение в micro 3D/Babylon охватывает сцену, систему координат, камеру, примитивы, трансформации, материал, текстуру и свет. Год завершается индивидуальным MVP.

Учебный план второго года — продвинутый проектный уровень

№	Модуль	Всего	Теория	Практика	Продукт
1	Проектная инженерия и версии	6	2	4	ТЗ и архитектурный прототип
2	Продвинутая 2D-физика Box2D	8	2	6	Физическая лаборатория
3	3D-сцена, координаты и камера	8	2	6	Навигация в 3D
4	Геометрия, материалы, свет и тени	8	2	6	3D-галерея
5	Интерактивность и модели в 3D	10	3	7	3D-симулятор
6	Игровые подсистемы, UI и данные	8	2	6	Вертикальный срез

№	Модуль	Всего	Теория	Практика	Продукт
7	Оптимизация, тестирование и доступность	6	2	4	Отчёт качества
8	Командный выпускной проект	14	3	11	Игра / приложение / симулятор
9	Публикация и техническая защита	4	0	4	Демонстрационный день
	Итого	72	18	54	

Содержание второго года. Команда формирует техническое задание, архитектуру, модель данных, backlog и правила интеграции. Продвинутая 2D-физика включает формы тел, joints, contacts, queries, tilemap и terrain. 3D-линия последовательно раскрывает координатные системы, камеру, иерархию объектов, геометрию, материалы, текстуры, свет, туман, тени и интерактивное управление.

Выпускной продукт объединяет UI, данные, поведение, прогресс, сохранение, физическую или 3D-модель. Команда проводит функциональное, пользовательское и нагрузочное тестирование, оптимизирует ассеты и отрисовку, оформляет документацию и публикацию. Возможные проекты: RPG vertical slice, стратегия/экономика, физический конструктор, интерактивный учебный симулятор, 3D-галерея или мини-симулятор.

Баланс программы. Общий объём — 144 часа: 38 часов теории и 106 часов практики. Обязательное ядро использует стабильные инструменты Studio; новые возможности расширенного Babylon-рендера включаются после педагогической апробации.

1.5. Система контроля и аттестации

Оценивание основано на доказательствах: работающий продукт, код, схема архитектуры, модель данных, результаты эксперимента, тестовый отчёт, документация, индивидуальный вклад и защита. Техническая сложность оценивается вместе с надёжностью, понятностью и пользой решения.

Вид	Срок	Форма	Что оценивается
Входной	Начало каждого года	Практическая задача, собеседование, портфолио	Стартовая траектория и поддержка
Текущий	Каждое занятие	Checkpoint, code review, лабораторный протокол	Понимание и качество инженерного действия
Модульный	После модуля	Прототип, демонстрация, технический разбор	Работоспособность подсистемы и перенос знаний
Промежуточный	Конец первого года	Индивидуальный MVP, документация, защита	Готовность к командной разработке
Итоговый	Конец второго года	Командный продукт, портфолио, техническая защита	Предметные, метапредметные и личностные результаты

Диагностические инструменты

- чек-лист инженерных навыков и карта технологий
- рубрика MVP и выпускного проекта
- архитектурная схема и паспорт модели
- тестовые сценарии, журнал дефектов и метрики производительности
- портфолио, журнал задач и подтверждение индивидуального вклада

- самооценка, peer review и техническое собеседование

Уровни освоения: начальный — решение воспроизводится по образцу; базовый — обязательная задача выполнена с отдельными подсказками; повышенный — решение самостоятельно переносится и обосновывается; творческий — обучающийся улучшает архитектуру или модель, подтверждает эффект тестами и помогает интеграции команды.

Раздел 2. Рабочая программа воспитания

2.1. Цель, задачи и целевые ориентиры воспитания

Цель воспитательной работы — формирование ответственного молодого разработчика, способного принимать этические инженерные решения, сотрудничать, уважать авторство и данные, видеть общественные последствия технологии и осознанно строить профессиональную траекторию.

- ответственность за качество, безопасность, сроки и достоверность представления результата
- уважение к пользователю, команде, интеллектуальной собственности и персональным данным
- профессиональная честность: признание ограничений, ошибок и реального вклада
- готовность к аргументированному диалогу, взаимопомощи и разрешению конфликтов
- ориентация на доступность, общественную пользу и устойчивое использование ресурсов

2.2. Формы и методы воспитания

- командный договор, роли и правила code review
- анализ этических кейсов ИИ, данных, монетизации и цифровых зависимостей
- аудит лицензий, приватности и доступности проекта
- профессиональная проба и встреча с разработчиком
- социально значимый цифровой проект
- хакатон/геймджем, демодень и публичная экспертиза
- ретроспектива и индивидуальный план развития

2.3. Условия организации воспитания

Педагог создаёт среду профессионального уважения: критика относится к решению и доказательствам, роли и вклад прозрачны, ошибки обсуждаются как инженерный материал. Публикация не раскрывает лишние данные; источники, лицензии и ограничения модели указываются честно; соревнование не отменяет взаимопомощь.

2.4. Календарный план воспитательной работы

Дата	Событие / тема	Форма	Целевой ориентир
Сентябрь	Командный кодекс и инженерная ответственность	Договор, роли, критерии качества	Ответственность
Октябрь	Авторство, лицензии и открытый код	Аудит зависимостей и ассетов	Правовая культура
Ноябрь	Данные, приватность и безопасность	Threat-кейс и настройки публикации	Цифровая безопасность
Декабрь	Доступность и этика интерфейса	UX-тест и исправления	Уважение к пользователю
Январь	Честность цифровой модели	Анализ допущений и погрешности	Познавательная честность
Февраль	Профессии разработки	Профессиональная проба	Самоопределение
Март	Технология для школы и общества	Социальный проект	Созидательная позиция
Апрель	Хакатон: ограничения и командное решение	Интенсив и peer review	Сотрудничество
Май	Демодень и профессиональная рефлексия	Защита, благодарность, план развития	Самооценка

Раздел 3. Комплекс организационно-педагогических условий

3.1. Календарный учебный график – https://www.s-82.ru/sites/default/files/2026/kalendarnyy_uchebnyy_grafik_urovnya_os_novnogo_obshchego_obrazovaniya_na_2026-2027_uchebnyy_god.pdf

3.1.1 Календарный учебный график

Год	Период	Неделя	Режим	Часов	Каникулы
Первый год	01.09.2026–31.05.2027	36	1 раз в неделю по 2 часа	72	По календарю учреждения
Второй год	01.09.2027–31.05.2028	36	1 раз в неделю по 2 часа	72	По календарю учреждения
Всего	2 учебных года	72	—	144	—

Конкретные даты занятий, каникул, переносов и аттестации ежегодно уточняются приказом учреждения. Общий объём и последовательность модулей сохраняются.

3.2. Условия реализации программы

Материально-технические условия

- кабинет на 10–12 рабочих мест, соответствующий санитарным и противопожарным требованиям
- компьютер или ноутбук на обучающегося с актуальным браузером, WebGL и аппаратным ускорением
- устройство педагога, проектор/панель, локальная сеть, Интернет и наушники
- AtmoCode Studio; тестовый проект для проверки Phaser, Box2D и micro 3D/Babylon
- средства резервного копирования, схем, протоколов эксперимента и проектной документации
- тестирование производительности перед началом 3D-модуля и облегчённый 2D-маршрут при технических ограничениях

Кадровые и педагогические условия

Педагог владеет microScript 2, архитектурой интерактивных приложений, Phaser/Box2D, основами 3D-графики, проектной методикой, тестированием и цифровой безопасностью. Перед 3D-модулем педагог проверяет доступность стабильных API текущей версии AtmoCode Studio и актуализирует примеры без изменения планируемых результатов.

Доступность и дифференциация

Обязательное ядро допускает 2D- или 3D-реализацию одинакового инженерного результата. Поддержка включает шаблон, API-карточки, парную работу и частично готовые модули. Углубление: сложная физика, оптимизация, процедурная генерация, расширенный Babylon API, собственный формат данных или исследовательская модель.

3.3. Методическое и информационное обеспечение

Методические принципы

- от требований и модели — к архитектуре, коду и доказательствам качества
- короткая техническая теория — лаборатория — интеграция в продукт
- вертикальные срезы вместо длительной сборки невидимых частей
- тестирование позитивных, граничных, ошибочных и нагрузочных сценариев
- документация решений, допущений, рисков и границ применимости
- ретроспектива по данным: дефекты, производительность, отзывы и выполненные критерии

Нормативно-правовая основа

- [Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»](#)
- [Приказ Минпросвещения России от 27.07.2022 № 629 о порядке организации дополнительного образования](#)
- [Концепция развития дополнительного образования детей до 2030 года, распоряжение Правительства РФ от 31.03.2022 № 678-р](#)
- [ФГОС основного общего образования, приказ Минпросвещения России от 31.05.2021 № 287](#)
- [ФГОС среднего общего образования, приказ Минобрнауки России от 17.05.2012 № 413](#)
- [Федеральная образовательная программа основного общего образования, приказ Минпросвещения России от 18.05.2023 № 370](#)
- [Федеральная образовательная программа среднего общего образования, приказ Минпросвещения России от 18.05.2023 № 371](#)
- [Письмо Минпросвещения России от 29.09.2023 № АБ-3935/06 о методических рекомендациях](#)
- [СанПиН 1.2.3685-21, гигиенические нормативы](#)
- [СП 2.4.2.4283-26, санитарные требования к организациям воспитания и обучения](#)
- [Федеральный закон от 27.07.2006 № 149-ФЗ об информации и защите информации](#)
- [Федеральный закон от 29.12.2010 № 436-ФЗ о защите детей от вредной информации](#)

Электронные образовательные ресурсы

- [AtmoCode — российская образовательная платформа](#)
- [AtmoCode Studio — среда разработки](#)
- [Сообщество и методическая поддержка AtmoCode](#)
- [Единое содержание общего образования](#)
- [Российская электронная школа](#)
- [Урок цифры](#)
- [Навигатор дополнительного образования Новосибирской области](#)
- [Официальный интернет-портал правовой информации](#)

Раздел 4. Приложения к программе

4.1. Календарно-тематический план первого года

Базовый инженерный уровень: 36 занятий по 2 академических часа. Даты уточняются после утверждения расписания.

№	Срок	Тема занятия	Всего	Т	П	Форма	Контроль
1	Сен., нед. —	Диагностика. Требования и безопасная разработка	2	1	1	Практикум	Артефакт
2	Сен., нед. —	Структура AtmoCode-проекта и версии	2	1	1	Практикум	Артефакт
3	Сен., нед. —	microScript 2: функции, объекты и стиль кода	2	1	1	Практикум	Артефакт
4	Сен., нед. —	Game loop и жизненный цикл	2	1	1	Практикум	Артефакт
5	Окт., нед. —	Конечный автомат состояний	2	0.5	1.5	Практикум	Артефакт
6	Окт., нед. —	Сцены, переходы и сервисы	2	0.5	1.5	Практикум	Артефакт
7	Окт., нед. —	Архитектурный прототип	2	0.5	1.5	Практикум	Артефакт
8	Окт., нед. —	Карта как структура данных	2	0.5	1.5	Практикум	Артефакт
9	Ноя., нед. —	Entities и компоненты	2	0.5	1.5	Практикум	Артефакт
10	Ноя., нед. —	Триггеры, события и параметры	2	0.5	1.5	Практикум	Артефакт
11	Ноя., нед. —	Data-driven приключение	2	0.5	1.5	Практикум	Артефакт
12	Ноя., нед. —	Пользовательский сценарий и wireframe	2	0.5	1.5	Практикум	Артефакт
13	Дек., нед. —	screen.ui: компоненты и события	2	0.5	1.5	Практикум	Артефакт
14	Дек., нед. —	HUD, настройки и доступность	2	0.5	1.5	Практикум	Артефакт
15	Дек., нед. —	storage: профиль и миграция данных	2	0.5	1.5	Практикум	Артефакт
16	Дек., нед. —	Phaser Arcade: тела и группы	2	0.5	1.5	Практикум	Артефакт
17	Янв., нед. —	Arcade: collision, overlap и контроллер	2	0.5	1.5	Практикум	Артефакт
18	Янв., нед. —	Box2D: тела, shapes и материалы	2	0.5	1.5	Практикум	Артефакт
19	Янв., нед. —	Box2D: contacts, queries и joints	2	0.5	1.5	Практикум	Артефакт
20	Янв., нед. —	Физический прототип и тест параметров	2	0.5	1.5	Практикум	Артефакт
21	Фев., нед. —	Графический редактор и ассет-пайплайн	2	0.5	1.5	Практикум	Артефакт
22	Фев., нед. —	Audio API и звуковые события	2	0.5	1.5	Практикум	Артефакт
23	Фев., нед. —	Tweens, particles и camera effects	2	0.5	1.5	Практикум	Артефакт

№	Срок	Тема занятия	Всего	Т	П	Форма	Контроль
24	Фев., нед. —	Lighting, filters и curves	2	0.5	1.5	Практикум	Артефакт
25	Мар., нед. —	Модель: переменные, параметры и гипотеза	2	0.5	1.5	Практикум	Артефакт
26	Мар., нед. —	Векторы, скорость и ускорение	2	0.5	1.5	Практикум	Артефакт
27	Мар., нед. —	Силы, столкновения и измерения	2	0.5	1.5	Практикум	Артефакт
28	Мар., нед. —	Графики, погрешность и вывод	2	0.5	1.5	Практикум	Артефакт
29	Апр., нед. —	micro 3D/Babylon: сцена и координаты	2	0.5	1.5	Проект	Артефакт
30	Апр., нед. —	Камера, примитивы и трансформации	2	0.5	1.5	Проект	Артефакт
31	Апр., нед. —	Материалы, текстуры и свет	2	0.5	1.5	Проект	Артефакт
32	Апр., нед. —	Тема и критерии MVP	2	0.5	1.5	Проект	Артефакт
33	Май, нед. —	Архитектура и прототип MVP	2	0.5	1.5	Проект	Артефакт
34	Май, нед. —	Тестирование и оптимизация	2	0.5	1.5	Проект	Артефакт
35	Май, нед. —	Документация и публикация	2	0.5	1.5	Проект	Артефакт
36	Май, нед. —	Защита индивидуального MVP	2	0.5	1.5	Защита	Артефакт

4.2. Календарно-тематический план второго года

Продвинутый проектный уровень: 36 занятий по 2 академических часа. Даты уточняются после утверждения расписания.

№	Срок	Тема занятия	Всего	Т	П	Форма	Контроль
1	Сен., нед. —	Диагностика, роли и командный договор	2	1	1	Практикум	Артефакт
2	Сен., нед. —	ТЗ, ограничения и критерии приёмки	2	1	1	Практикум	Артефакт
3	Сен., нед. —	Архитектура, модель данных и backlog	2	1	1	Практикум	Артефакт
4	Сен., нед. —	Box2D: advanced shapes и sensors	2	1	1	Практикум	Артефакт
5	Окт., нед. —	Joints, contacts и queries	2	0.5	1.5	Практикум	Артефакт
6	Окт., нед. —	Tilemap и Box2D terrain	2	0.5	1.5	Практикум	Артефакт
7	Окт., нед. —	Физическая лаборатория	2	0.5	1.5	Практикум	Артефакт
8	Окт., нед. —	3D-система координат и сцена	2	0.5	1.5	Практикум	Артефакт
9	Ноя., нед. —	Камера и управление обзором	2	0.5	1.5	Практикум	Артефакт

№	Срок	Тема занятия	Всего	Т	П	Форма	Контроль
10	Ноя., нед. —	Иерархия и трансформации	2	0.5	1.5	Практикум	Артефакт
11	Ноя., нед. —	Навигационный 3D-прототип	2	0.5	1.5	Практикум	Артефакт
12	Ноя., нед. —	Геометрия и загрузка моделей	2	0.5	1.5	Практикум	Артефакт
13	Дек., нед. —	Материалы и текстуры	2	0.5	1.5	Практикум	Артефакт
14	Дек., нед. —	Источники света и тени	2	0.5	1.5	Практикум	Артефакт
15	Дек., нед. —	Туман, фон и композиция	2	0.5	1.5	Практикум	Артефакт
16	Дек., нед. —	Выбор объекта и интерактивность	2	0.5	1.5	Практикум	Артефакт
17	Янв., нед. —	Управление и столкновения как модель	2	0.5	1.5	Практикум	Артефакт
18	Янв., нед. —	3D-симулятор: параметры и измерения	2	0.5	1.5	Практикум	Артефакт
19	Янв., нед. —	UI поверх 3D и состояние	2	0.5	1.5	Практикум	Артефакт
20	Янв., нед. —	Тестирование 3D-сценария	2	0.5	1.5	Практикум	Артефакт
21	Фев., нед. —	Поведение и конечные автоматы	2	0.5	1.5	Практикум	Артефакт
22	Фев., нед. —	Инвентарь, прогресс и экономика	2	0.5	1.5	Практикум	Артефакт
23	Фев., нед. —	Диалоги, настройки и сохранение	2	0.5	1.5	Практикум	Артефакт
24	Фев., нед. —	Вертикальный срез продукта	2	0.5	1.5	Практикум	Артефакт
25	Мар., нед. —	Профилирование и бюджет ассетов	2	0.5	1.5	Практикум	Артефакт
26	Мар., нед. —	Оптимизация отрисовки и загрузки	2	0.5	1.5	Практикум	Артефакт
27	Мар., нед. —	Доступность и разные устройства	2	0.5	1.5	Практикум	Артефакт
28	Мар., нед. —	Идея, аудитория и питч команды	2	0.5	1.5	Практикум	Артефакт
29	Апр., нед. —	Спринт 1: каркас и интеграция	2	0.5	1.5	Проект	Артефакт
30	Апр., нед. —	Спринт 2: системы и контент	2	0.5	1.5	Проект	Артефакт
31	Апр., нед. —	Альфа-тест и журнал дефектов	2	0.5	1.5	Проект	Артефакт
32	Апр., нед. —	Спринт 3: исправления и полировка	2	0.5	1.5	Проект	Артефакт
33	Май, нед. —	Документация, лицензии и страница проекта	2	0	2	Проект	Артефакт
34	Май, нед. —	Release candidate и репетиция	2	0	2	Проект	Артефакт
35	Май, нед. —	Демонстрационный день и техническая защита	2	0	2	Проект	Артефакт

№	Срок	Тема занятия	Всего	Т	П	Форма	Контроль
36	Май, нед. —	Ретроспектива и профессиональный маршрут	2	0	2	Защита	Артефакт

4.3. Оценочные материалы

Входная практическая диагностика

- прочитать фрагмент microScript 2 и объяснить поток управления
- добавить состояние или функцию в небольшой проект
- предложить структуру данных для предметной задачи
- локализовать дефект по наблюдаемому поведению и сообщению
- объяснить правила лицензирования, приватности и безопасной публикации

Рубрика индивидуального MVP первого года

Критерий	Показатель	Баллы
Проблема и модель	требования, данные, состояния и допущения согласованы	0–3
Архитектура и код	структура понятна, решения объяснимы, критические ошибки устранены	0–3
Техническая система	физика, UI, 3D или модель применены корректно	0–3
Эксперимент и тесты	есть сценарии, данные проверки и интерпретация	0–3
Документация	инструкция, источники и ограничения оформлены	0–3
Защита	демонстрация устойчива, решения аргументированы	0–3

Интерпретация: 0–6 — начальный; 7–11 — базовый; 12–15 — повышенный; 16–18 — творческий. Для второго года требуется базовый уровень или индивидуальный план поддержки.

Рубрика выпускного проекта второго года

Критерий	Показатель	Баллы
Ценность и требования	проблема, пользователь и критерии приёмки подтверждены	0–3
Архитектура и данные	модули, состояния, структуры и контракты обоснованы	0–3
2D/3D и физика	рендер, камера, модель или физика работают корректно	0–3
Алгоритмы и подсистемы	поведение, UI, прогресс и хранение связаны	0–3
Качество и производительность	есть тесты, профиль, исправления и проверка устройств	0–3
UX, графика и звук	интерфейс доступен, мультимедиа функционально	0–3
Командная инженерия	backlog, версии, review, интеграция и вклад подтверждены	0–3
Документация и этика	источники, лицензии, данные и ограничения оформлены	0–3
Техническая защита	показ устойчив, ответы точны, рефлексия конкретна	0–3

Интерпретация: 0–9 — начальный; 10–17 — базовый; 18–23 — повышенный; 24–27 — творческий. Итог учитывает продукт, защиту, портфолио и индивидуальный вклад.

Журнал наблюдения

№	Ф.И.О.	Предметные 0–3	Метапредметные 0–3	Личностные 0–3	Комментарий / следующий шаг
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

4.4. Паспорт выпускного проекта и лист самооценки

Название и тип продукта	игра / приложение / 2D- или 3D-симулятор
Проблема, пользователь и ценность	
Команда, роли и критерии приёмки	
Архитектура и модель данных	
2D/3D-сцены, камера и физика	
Алгоритмы и игровые подсистемы	
UI/UX, доступность и управление	
Графика, модели, звук и эффекты	
Тесты, производительность и устройства	
Публикация, ссылка / QR-код и лицензии	
Ограничения и следующий релиз	

Лист самооценки

Утверждение / вопрос	Ответ
Я могу объяснить архитектуру, данные и ограничения модели	☐ пока трудно ☐ с подсказкой ☐ самостоятельно ☐ могу провести review
Я обосновал выбор 2D/3D, физики и ключевых API	☐ пока трудно ☐ частично ☐ полностью ☐ сравнил альтернативы
Я проверил основной, граничный, ошибочный и нагрузочный сценарии	☐ нет ☐ частично ☐ полностью ☐ автоматизировал часть
Продукт проверен пользователем и на целевом устройстве	☐ нет ☐ частично ☐ да ☐ улучшен по данным
Я соблюдал безопасность, авторство и лицензии	☐ не уверен ☐ с напоминанием ☐ самостоятельно ☐ провёл аудит
Мой индивидуальный вклад подтверждён	☐ не определён ☐ частичный ☐ полный ☐ помог интеграции
Самое сильное инженерное решение	
Главный дефект, измерение и исправление	
Что я изменю в следующем релизе и почему	

Финальный результат. Портфолио включает индивидуальный MVP и командный выпускной продукт: опубликованную версию, исходный проект, ТЗ, архитектурную схему, тестовый отчёт, документацию, описание личного вклада и материалы технической защиты.