Министерство образования Края
Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

СОГЛАСОВАНО:
Заведующая учебной частью
_____ Д.С. Симашова
протокол № 13
от «02» 09 2026 г

УТВЕРЖДАЮ:
Директор филиала
_____ Е. С. Тадер
Приказ № 02-02-94
от «02» 09 2026 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА НА 2026– 2027 уч. год
технической направленности

«Основы программирования / Кодвардс»

Форма реализации программы – очная
Срок реализации – 1 год
Возраст обучающихся – 7-11 лет

Составитель программы:
Киселёв Иван Михайлович
педагог дополнительного образования

г. Норильск, 2026 г.

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 3
1.1. НОВИЗНА ДООП 4
1.2. АКТУАЛЬНОСТЬ ДООП 4
1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ 5
1.4. ЦЕЛЬ ДООП 5
1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП 6
1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП 6
1.7. СРОК РЕАЛИЗАЦИИ ДООП 7
1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ 7
1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ 8
2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН 10
3. СОДЕРЖАНИЕ ДООП 12
4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ 20
5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА 20
6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП 20
7. МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ПО РАЗДЕЛАМ ПРОГРАММЫ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ 21

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования / Кодвардс» имеет техническую направленность и разработана в соответствии с основными нормативно-правовыми документами: Федеральным Законом «Об образовании» от 29.12.2012 г. № 273-ФЗ; Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам от 09.11.2018 г. № 196; Целевой моделью развития региональных систем дополнительного образования детей от 03.09.2019 г. № 467; Концепцией развития дополнительного образования детей до 2030 года утвержденной распоряжением Правительства Российской Федерации от 31.03.2022 №678-р; Санитарно-эпидемиологическими требованиями к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи от 28.09.2020 г. № 28.

Начало 21 века ознаменовано бурным развитием IT-технологий. Мировые лидеры IT-индустрии периодически обращаются к школьникам с призывом изучать программирование. Становится понятно, что чем раньше ребенок начнет овладевать навыками программирования, тем больший запас знаний и технологий он получит к моменту выбора основного рода деятельности. Даже если в будущем карьерный путь ребенка не будет связан с программированием, умение разбираться в сложных системах и взаимодействовать с новыми технологиями ему пригодится в любой сфере, ведь цифровые технологии используются повсеместно. Занятия по программированию помогут ребенку сделать первые шаги в мире создания компьютерных моделей, позволят познакомиться с сообществом таких же заинтересованных ребят, введут во все подробности и тонкости проектной деятельности.

Овладевая навыками программирования, ребенок затрагивает и смежные сферы: логика, вычислительная математика, теория вероятности, а также и другие научные области: география, биология, физика, литература, в зависимости от интересов. Когда у ребенка сформирован необходимый набор знаний и умений, выполнен ряд задач и упражнений по разным темам, он может, используя их, работать над собственным проектом. Это позволяет развивать творческие способности, проводить собственные исследования, работать в команде, и, что немаловажно, видеть результат собственной работы, вносить в неё коррективы, развивать её.

В основе курса лежит образовательная программа «Кодвардс», которая включает в себя комплекс теоретических знаний и практических заданий в игровой форме. Методология обучения была разработана специалистами ООО «РЭДМЭДРОБОТ», которое является лидером российского рынка по мобильной разработке. Компания разрабатывает флагманские мобильные бизнес-сервисы для компаний: Вымпелком (бренд Билайн), Альфастрахование, Альфа-Банк, Банк Открытие, Азбука Вкуса, РБК.

ООО «РЭДМЭДРОБОТ» является представителем профессиональной среды разработчиков программного обеспечения. Как и НПО «Игровое образование», которое профессионально занимается разработкой игровых методик. Компанией спроектировано и реализовано более 32-х проектов в сфере образования, таких, как «Мозаикум», «Картограф» и т.д. Также является разработчиком и организатором курса «Игропрактика для педагогов».

1.1. НОВИЗНА ДООП

«Кодвардс» знакомит детей с основами программирования. Уроки проводятся в игровой форме с выполнением интеллектуальных дидактических и компьютерных заданий. Изучение программирования не только в графической, но и он-лайн среде (на платформе https://studio.code.org/), позволяет организовать процесс обучения в игровой форме, что делает содержание программы новым, современным и доступным. Курс дает базовые принципы написания кода для любого языка программирования.

Авторская методика преподавания составлена в расчете на то, что прошедшие курс дети будут обладать навыками и умениями, ценными в профессиональной среде разработчиков программного обеспечения.

Очень важным новшеством является то, что педагог может отслеживать прогресс прохождения курса в своем личном кабинете на платформе https://go.codewards.ru/, где доступна статистика по ученикам, управление аккаунтами, а также результаты усвоения материалов.

1.2. АКТУАЛЬНОСТЬ ДООП

Актуальность программы состоит в ее содержании, направленном на развитие навыков креативного программирования, на которые имеется очень большой запрос от современного мира. Программа отвечает потребностям детей и родителей: формирует социально значимые знания, умения и навыки, оказывает комплексное обучающее, развивающее, воспитательное и здоровьесберегающее воздействие,

Программирование, что очень важно, развивает вычислительное мышление, которое учит формулировать проблему, искать решение и анализировать его. Обучать детей программированию нужно не для того, чтобы они в будущем стали программистами. Если ребенок на занятиях по программированию научился раскладывать большую задачу на маленькие фрагменты, находить сходства в разных элементах, выявлять и устранять незначимые детали, выстраивать фрагменты в единый алгоритм для получения результата, он сможет решить задачи в любой дисциплине.

При обычном обучении информатике, темы «алгоритмы» и «программирование» изучаются очень мало и поздно, это замедляет формирование алгоритмического и вычислительного мышления, не способствует развитию интереса учащихся в области программирования. Учащиеся, как правило, не готовы успешно выступать на олимпиадах по информатике, теряют интерес к предмету. Обучение по ДООП «Основы программирования / Кодвардс» создает благоприятные условия для более раннего «погружения» учащихся в

мир логики, математического моделирования, для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся. Курс рассчитан на сотворчество и сотрудничество педагога и воспитанников. Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

Процесс обучения выстроен с учетом используемых современных образовательных технологий: информационные технологии обучения, технология проектного обучения, что отражено в дидактических принципах обучения (индивидуальность, доступность, преемственность, результативность и др.). Наряду с технологичностью обучения в образовательном процессе решаются задачи организации и управления.

1.3. ПЕДАГОГИЧЕСКАЯ ЦЕЛЕСООБРАЗНОСТЬ

Педагогическая целесообразность программы обусловлена тем, что в соответствии с учебным планом программы на каждом этапе обучения разработаны задания, наиболее эффективные для развития познавательного потенциала обучающихся. Программа построена «от простого к сложному». Информационные и онлайн технологии обучения, применяемые в данной программе в процессе изучения программирования, обеспечивают интегративный подход и позволяют сделать обучение эффективным, доступным, гибким и мобильным. Широкое использование в процессе обучения дистанционных онлайн ресурсов несомненно очень целесообразно в условиях неблагоприятных северных погодных аномалий и актированных для школьников учебных дней.

В процессе обучения по программе ведущее место занимают интерактивные и активные методы обучения, проектные технологии, обеспечивающие эффективное достижение поставленной цели обучения. Благодаря возможностям платформы Кодвардс отслеживание результативности обучения происходит на всех этапах работы.

Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Они способствуют осознанию учащимися того важного обстоятельства, что наряду с разрабатываемыми ими программами рождается внутренний образовательный продукт: освоенные способы действий в программировании. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся, поддерживая интерес к профессиям, связанным с программированием.

1.4. ЦЕЛЬ ДООП

Цель программы – формирование понимания структуры программирования, умения оперировать базовыми понятиями и концепциями программирования без связки с синтаксисом языка программного кода.

Задачи программы:

Предметные:

– освоить понятия «алгоритм», «программа», «переменная», «сообщение», «данные» через призму практического опыта в ходе создания программных кодов;

– изучить визуальную среду программирования «Кодвардс»;

– овладеть навыками использования различных команд среды для решения задач, составления линейных, разветвляющихся и циклических алгоритмов управления исполнителями;

– обучить приемам и навыкам алгоритмизации и проектирования коротких программ в рациональном стиле программирования;

– способствовать формированию умения работать над творческим проектом, разрабатывать и осуществлять его защиту;

– формировать навыки сетевой коммуникации и безопасного использования сети Интернет.

Личностные:

– сформировать познавательную, творческую активность, фантазию и изобретательность;

– воспитать умение добиваться успеха и правильно оценивать неудачи;

– сформировать навыки самоорганизации учащихся, их уверенность в себе через выполнение самостоятельных творческих проектов и их защиту;

– укрепить готовность к повышению своего образовательного уровня, самообучению и продолжению обучения в области разработки компьютерных игр и программирования.

Метапредметные:

– сформировать навыки самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в процессе создания творческих проектов;

– развить умение самостоятельно планировать пути достижения целей, соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, оценивать правильность выполнения поставленной задачи.

1.5. ВОЗРАСТ ОБУЧАЮЩИХСЯ, УЧАСТВУЮЩИХ В ДООП

Программа «Основы программирования / Кодвардс» рассчитана на обучающихся 9-11 лет.

1.6. УСЛОВИЯ ВХОЖДЕНИЯ В ДООП

Набор на программу осуществляется в соответствии с Положением о наборе в Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске.

Поступающий не должен владеть специализированными навыками и умениями для вхождения в программу.

1.7. СРОК РЕАЛИЗАЦИИ ДООП

Программа рассчитана на 1 учебный год. Нагрузка на обучающегося составляет 144 часа за весь период обучения.

1.8. РЕЖИМ ЗАНЯТИЙ, ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ

Учебные занятия проходят в очной форме. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час - 40 минут) с обязательным перерывом.

Формы проведения занятий: беседы, игры, практические занятия, самостоятельная работа, викторины и проекты. Использование метода проектов позволяет обеспечить условия для развития у ребят навыков самостоятельной постановки задач и выбора оптимального варианта их решения, самостоятельного достижения цели, анализа полученных результатов с точки зрения решения поставленной задачи. Программой предусмотрены методы обучения: объяснительно-иллюстративные, частично-поисковые (вариативные задания), творческие, практические, задания на поиск ошибок в коде.

Программа реализуется на основе следующих **принципов**:

– Обучение в активной познавательной деятельности. Все темы учащиеся изучают на практике, выполняя различные творческие задания, общаясь в парах и группах друг с другом.

– Принцип природосообразности. Основной вид деятельности учащихся – игра, через нее дети познают окружающий мир, поэтому в занятия включены игровые элементы, способствующие успешному освоению программы.

– Преемственность. Программа «Основы программирования / Кодвардс» построена так, что каждая последующая тема логически связана с предыдущей. Данный принцип помогает понять важность уже изученного материала и значимость каждого отдельного занятия.

– Практико-ориентированность, обеспечивающая отбор содержания, направленного на решение простейших практических задач планирования деятельности, поиска нужной информации, инструментирования всех видов деятельности на базе общепринятых средств информационной деятельности, реализующих основные пользовательские возможности информационных технологий. При этом исходным является положение о том, что компьютер может многократно усилить возможности человека, но не заменить его.

– Принцип развивающего обучения (обучение ориентировано не только на получение новых знаний в области информатики и информационных технологий, но и на активизацию мыслительных процессов, формирование и развитие у школьников обобщенных способов деятельности, формирование навыков самостоятельной работы).

1.9. ОЖИДАЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ И ЭФФЕКТЫ, СПОСОБЫ ПРЕДЪЯВЛЕНИЯ И ОТСЛЕЖИВАНИЯ РЕЗУЛЬТАТОВ

После завершения освоения программы воспитанники будут **знать, понимать:**

- основы алгоритмики;
- инструменты среды визуального программирования Кодвардс;
- основы создания программируемой анимации и способы управления исполнителем;
- основы технологии разработки образовательных проектов.

будут уметь:

- алгоритмически и логически мыслить;
- разбивать большую задачу на подзадачи, для эффективного поэтапного ее решения;
- создавать анимацию и простые интерактивные игры, в которых реализуются движения и взаимодействия различных объектов с помощью инструментов среды визуального программирования Кодвардс.

Личностные результаты:

- сформировано умение взаимодействовать с членами своей группы, решающей общую задачу;
- укреплена готовность к повышению своего образовательного уровня, самообучению и продолжению обучения в области разработки компьютерных игр и программирования.

Метапредметные результаты:

- сформированы навыки самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в процессе создания творческих проектов;
- развито умение самостоятельно планировать пути достижения целей, соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, оценивать правильность выполнения поставленной задачи.

Предметные результаты — обучающиеся будут:

- владеть терминами «алгоритм», «программа», «переменная», «сообщение», «данные»;
- проектировать короткие программы в рациональном стиле программирования;
- уметь работать в визуальной среде программирования Кодвардс;
- уметь использовать различные команды для решения поставленных перед исполнителем задач, составлять линейные, разветвляющиеся и циклические алгоритмы управления исполнителями;
- создавать и выполнять программы для решения логических задач, стоящих перед исполнителем;

– уметь работать над творческим проектом и осуществлять его защиту.

Критериями оценки теоретических знаний являются: степень усвоения теоретического материала, глубина, широта и системность теоретических знаний, грамотное использование компьютерных терминов.

Критериями уровня овладения практическими умениями и навыками являются: разнообразие умений и навыков, грамотность (соответствие существующим нормативам и правилам, технологиям) практических действий, свобода владения специальным компьютерным оборудованием и программным обеспечением, качество творческих проектов учащихся — грамотность исполнения, использование творческих элементов.

Дистанционное обучение может применяться с целью индивидуального обучения учащихся, пропустивших занятия по болезни, или другим причинам, а также в условиях ограничительных мероприятий (медицинских или по метеорологическим условиям). Дистанционное обучение осуществляется с применением сервисов сети Интернет.

Методы определения уровня обученности: собеседование, наблюдение, фронтальный опрос, тестирование, экспертная оценка компьютерного проекта, деловая игра, презентация продукта деятельности, самостоятельная практическая работа, внутригрупповой конкурс, диагностическая игра, игра-испытание.

При оценке творческих работ педагог должен руководствоваться следующими критериями:

- общая художественная выразительность;
- самостоятельность и оригинальность замысла;
- оптимальность и правильность написанного программного кода;
- техническая сложность проекта.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Тема занятия	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Раздел 1. Первый опыт программирования на платформе https://studio.code.org/ (10 часов)					
1.1	Безопасная работа за компьютером. Проекты платформы https://studio.code.org/. Классический лабиринт с Angry Birds	2	1	1	
1.2	Проекты https://studio.code.org/. Приключение с Майнкрафт.	6	0,5	5,5	
1.3	Текущий контроль по разделу «Первый опыт программирования на платформе Кодвардс».	2	0,5	1,5	Фронтальный опрос, интегрированное тестирование, самостоятельная практическая работа на платформе https://studio.code.org/
Раздел 2. Введение в игровой мир Кодвардс (18 часов)					
2.1	Бортовой журнал программиста-испытателя.	2	1	1	
2.2	Базовые сведения об алгоритмах и их исполнителях.	8	1	7	
2.3	Простейшие способы классификации объектов, управление разными объектами.	6	0,5	5,5	
2.4	Текущий контроль по разделу «Введение в игровой мир Кодвардс».	2	1	1	Фронтальный опрос, интегрированное тестирование, самостоятельная практическая работа по азам программирования движения исполнителя
Раздел 3. Повторяющиеся действия в программе. Циклы (18 часов)					
3.1	Начальные сведения о циклах. Грамотное использование циклов в программах для оптимизации кода.	10	2,5	7,5	

3.2	Несколько циклов в одной программе.	6	0,5	5,5	
3.3	Текущий контроль по разделу «Повторяющиеся действия в программе. Циклы».	2	1	1	Теоретическое тестирование, самостоятельная практическая работа по с циклами в программном коде
Раздел 4. Переменные и сенсоры в Кодвардсе (26 часов)					
4.1	Вложенные циклы.	6	2	4	
4.2	Переменные и сенсоры.	10	4	6	
4.3	Введение в концепцию «условия»	4	1	3	
4.4	Цикл с предусловием	4	1,5	2,5	
4.5	Текущий контроль по разделу «Переменные и сенсоры в Кодвардсе»	2	1	1	Поиск ошибок в программном коде, практическая работа по программированию сенсорного луча
Раздел 5. Введение в проектную деятельность (10 часов)					
5.1	Планирование работы над проектом	2	1	1	
5.2	Выполнение задач по плану	6	1	5	
5.1	Текущий контроль по разделу «Введение в проектную деятельность»	2	0	2	Практическое задание на платформе Кодвардс
Раздел 6. Движение объектов в игровом мире (14 часов)					
6.1	Тренировка командной работы	4	1,5	2,5	
6.2	Программирование клавиш управления	8	3	5	
6.3	Текущий контроль по разделу «Движение объектов в игровом мире»	2	1	1	Теоретическое тестирование, практическая работа по программированию клавиш управления исполнителем
Раздел 7. Коллизии объектов, первые игровые миры (34 часа)					
7.1	Разные типы объектов в среде программирования, простые взаимодействия по коллизии	4	1,5	2,5	

7.2	Изменение свойства видимости спрайта при коллизии	2	0,5	1,5	
7.3	Движение объектов как сдвиг по осям координат	4	0,5	3,5	
7.4	Знакомство со счетом и счетчиками в игровом мире	4	1,5	2,5	
7.5	Знакомство со счетом и таймером в игровом мире	4	1	3	
7.6	Текстовые сообщения в игре	4	2	2	
7.7	Формирование программного кода для управления объектами в игровом мире с помощью мыши	10	2,5	7,5	
7.8	Текущий контроль по разделу «Коллизии объектов, первые игровые миры»	2	1	1	Теоретическое тестирование, презентация продукта деятельности «Сложная анимация»
Раздел 8. Разработка и защита итогового проекта (14 часов)					
8.1	Создание своей игры	10	2	8	
8.2	Защита проектов.	4	0	4	
	ИТОГО	144	40	104	

3. СОДЕРЖАНИЕ ДООП

Раздел 1. Первый опыт программирования на платформе https://studio.code.org/ (10 часов)

Тема 1.1 Безопасная работа за компьютером. Проекты платформы https://studio.code.org/. Классический лабиринт с Angry Birds **(2 часа).**

Теория: вводная лекция, рассказ о программировании в современном мире + профессии, связанные с ним. Безопасная работа за компьютером. Правила безопасной работы в сети Интернет. Интерфейс платформы https://studio.code.org/. Основные алгоритмические конструкции (краткий обзор).

Практическая работа: включение браузерного переводчика, работа с ресурсом «Классический лабиринт с Angry Birds» (https://studio.code.org/hoc/1).

Тема 1.2 Проекты https://studio.code.org/. Приключение с Майнкрафт **(6 часов)**.

Теория: основные команды для управления героем, основные блоки Minecraft.

Практическая работа:

- Приключение с Майнкрафт: Водное путешествие (Майнкрафт: Водное путешествие - Code.org)
- Приключение с Майнкрафт: Путешествие героя (Майнкрафт: Путешествие Героя - Code.org)
- Майнкрафт: Искатель приключений (Приключение с Minecraft - Code.org)

Текущий контроль по разделу «Первый опыт программирования на платформе https://studio.code.org/» **(2 часа,** Приложение №1**).**

Раздел 2. Введение в игровой мир Кодвардс (18 часов)

Тема 2.1 Бортовой журнал программиста-испытателя **(2 часа).**

Теория: первое знакомство с миром Кодвардс и его ремонтным роботом. Первая команда движения move [мув] (объект: robot) – шагать (двигаться). Правила работы с Бортовым журналом Кодвардса.

Практическая работа: задания из Бортового журнала — начало алгоритмики и программирования.

Тема 2.2. Базовые сведения об алгоритмах и их исполнителях **(8 часов)**.

Теория: понятие «Алгоритм». Важность порядка действий в алгоритме. Понятие «Исполнитель алгоритмов». Новые команды робота: rotate [ротейт] – поворот, left [лэфт] – налево, right [райт] – направо. Готовые алгоритмы. Ошибки в алгоритмах и их исправление.

Практическая работа: работа с командами движения move, rotate, right, left. Создание алгоритмов для ремонтного робота.

Тема 2.3. Простейшие способы классификации объектов, управление разными объектами **(6 часов).**

Теория: понятие «формальный» и «неформальный» исполнитель. Связь «формальный исполнитель» - «система команд». Понятия «объект» и «свойство объекта». Истинность и ложность высказываний. Дополнительные функции load [лоад] - взять, put [пут] - положить (объект: crane).

Практическая работа: создание алгоритмов. Освоение команд движения move, rotate, right, left, load, put. Составление программ для управления несколькими объектами (объекты: truck, loader0, loader1).

Текущий контроль по разделу «Введение в игровой мир Кодвардс» **(2 часа,** Приложение №2).

Раздел 3. Повторяющиеся действия в программе. Циклы (18 часов).

Тема 3.1. Начальные сведения о циклах. Грамотное использование циклов в программах для оптимизации кода **(10 часов).**

Теория: понятие «цикл». Операции. Примеры циклов. Способы использования циклов в программах. Понятия «оптимизация», «оптимальный путь». Понятия «многофункциональный» и «монофункциональный» объект. Последовательные циклы.

Практическая работа: выделение повторяющихся действий. Продолжение освоения команд движения move, rotate, right, left, load, put. Освоение оператора цикла loop [лууп]… end [энд]. Определение оптимального пути решения задачи. Применение принципов оптимизации. Определение и применение конструкции loop … end в коде. Освоение команды бурения drill [дрил]. Использование последовательных циклов.

Тема 3.2. Несколько циклов в одной программе **(6 часов).**

Теория: последовательные циклы. Оптимизация.

Практическая работа: определение и применение конструкции «цикл» в коде. Использование последовательных циклов. Отработка навыков использования команд движения move, rotate, right, left, load, put. Освоение оператора цикла loop… end.

Текущий контроль по разделу «Повторяющиеся действия в программе. Циклы» **(2 часа,** Приложение №3).

Раздел 4. Переменные и сенсоры в Кодвардсе (26 часов).

Тема 4.1. Вложенные циклы **(6 часов)**.

Теория: понятие «Конвейер». Понятие «Вложенные циклы». Применение концепции вложенных циклов в программном коде.

Практическая работа: управление работой конвейера и нескольких роботов для организации обработки и перевозки минералов с помощью команд move, rotate, left, right, load, put (объекты: belt, loader, crane) и оператора цикла loop…end.

Тема 4.2 Переменные и сенсоры **(10 часов)**.

Теория: понятие «переменная» и необходимость её использования. Способы использования переменной в программном коде. Способы присвоения переменной разных значений. Переменная как свойство объекта. Процесс сортировки и классификации информации. Понятие «сенсор» (датчик, измеритель). Оператор var [вар] – оператор переменной. Переменная как буль. Були: TRUE – [тру] истина, FALSE – [фолс] ложь. Знак присвоения: = [одинарное равно]. Знак сравнения: == [двойное равно]. Условие: if [иф] – если. Новые команды switch [свич] – повернись (в отношении датчика), push [пуш] – нажать для объектов valve [валв] – вентиль и pump [памп] – помпа.

Практическая работа: использование конструкции кода cart.stepToFlag [карт стэп ту флэг] — любое расстояние до флажка и переменной dist [дист] – расстояние. Использование переменной: var height=loader.blockHeight [лоудер

блок хэйт] – сортировка по высоте для объекта loader. Использование переменной var fire=sensor.isFire [сэнсор из файер] – если есть огонь для объекта firetruck [файертрак]. Использование переменной valve.switch TRUE/FALSE для тушения огня.

Тема 4.3 Введение в концепцию «условия» **(4 часа)**.

Теория: понятия «истина» и «ложь». Введение в концепцию «условия». Ознакомление с конструкцией кода if [иф] – если … else [элс] - иначе. Использование в конструкции условия переменной. Знакомство со свойствами объектов:

- Tail [тэйл] – хвост.
- Fur [фур] – шерсть, мех.
- White [уайт] – белый.
- Black [блэк] – черный.
- Predator [предатор] – хищник.

Практическая работа: использование переменной var animal = robot.scan для проверки условия if animal.hasTail == TRUE/FALSE [энимал хэз тэйл] – если у животного есть хвост. Работа со значениями переменной:

- robot.animalHasTale [робот энимал хэз тэйл] – робот определяет состояние – есть ли хвост.
- robot.animalHasFur [робот энимал хэз фур] – робот определяет состояние – есть ли шерсть.
- robot.animalIsPredator [робот энимал из предатор] – робот определяет состояние – хищник или нет.
- robot.animalColor [робот энимал кала] – робот определяет цвет животного.

Тема 4.4 Цикл с предусловием **(4 часа)**.

Теория: ознакомление с «циклом с предусловием». Ознакомление с конструкцией кода while. Использование в конструкции условия переменной. Знакомство со свойствами объектов:

- isBaby [из бэйби] – детёныш.
- crystalColor [кристал кала] – цвет кристалла – здесь про цвет кристалла, который в колесе

Новые функции:

- callAnimals [колл энималс] – подзывать животных.
- take food [тэйк фуд] – возьми еду.
- take toy [тэйк той] – возьми игрушку.
- rotate right [ротейт райт] – поверни направо.
- flicker [фликер] – зажгись.

Практическая работа: использование циклической конструкции while platform.isPressed == true [уайл платформ из прессд] – до тех пор пока платформа нажата и условной конструкции if animal.isBaby [из энимал из бэйби] –

если животное детёныш для игры со зверями в зоопарке. Отработка программного кода:

1. var clr = ring.keyColor [вар кала равно ринг ки кала] – записываем в переменную «информацию» о цвете, который нужен для открытия.
2. while ring.crystalColor != clr [уайл ринг кристал кала не того же цвета, что кала] – пока колесо определяет цвет кристалла не равный нужному цвету для открытия.
3. ring.rotate right [ринг ротейт райт] – колесо поворачивайся направо.

Текущий контроль по разделу «Переменные и сенсоры в Кодвардсе» **(2 часа,** Приложение №4).

Раздел 5. Введение в проектную деятельность (10 часов).

Тема 5.1 Планирование работы над проектом **(2 часа).**

Теория: введение в проектную деятельность. Знакомство с понятием «проект». Закрепление понятия «цель». Знакомство с понятием «план проекта». Таймлайн проекта.

Практическая работа: формирование умения декомпозировать задачи и планировать выполнение задач.

Тема 5.2 Выполнение задач по плану **(6 часов).**

Теория: конструкция и принципы постройки подводных судов. Новый объект picker [пикер] – сборщик кет. Новые функции rise [райз] – поднимать и turn [тюрн] – поворот на 45 градусов.

Практическая работа: закрепление концепций программирования, изученных ранее. Работа с кодом для сбора кегов:

1. var keg = picker.isKeg [вар кег пикер из кег] – в переменную записываем «информацию» есть перед нами есть кега или нет.
2. if keg == true [иф кег тру] – если перед нами кега.
3. picker.load [пикер лоад] – сборщик возьми.
4. end [энд] – конец.

Работа с кодом:

1. var height= robot.pillarHeight [вар хайт робот пила хайт] – в переменную записывается «информация» о высоте столба
2. robot.rise height [робот райз хайт] – робот подними на нужную высоту.

Работа с кодом для балансировки субмарины:

1. while vol0 != vol1 [уайл вольюм ноль не равен вольюм один] – пока объем не равен объему1.
2. tank1.pump 1 [танк памп один] – бак накачай одну единицу воды.

3. var vol1 = tank1.volume [вар вольюм одинтанк вольюм] – в переменную записывается «информация» о свободном объеме бака.
4. end [энд] – конец.

Текущий контроль по разделу «Введение в проектную деятельность» **(2 часа,** Приложение №5).

Раздел 6. Движение объектов в игровом мире (14 часов).

Тема 6.1 Тренировка командной работы **(4 часа).**

Теория: вспоминаем и закрепляем пройденный материал.

Практическая работа: вспоминаем и закрепляем пройденный материал.

Тема 6.2 Программирование клавиш управления **(8 часов)**.

Теория: знакомство с понятием «координатная плоскость», с отрицательным числом (знаком минус) в аргументе и концепцией управления объектом с клавиатуры. События (events): @onKey [онки] – по нажатию клавиши. Новые команды move [мув] – команда двигаться по горизонтали, lift [лифт] двигаться по вертикали, jump [джамп] – прыгать, setX [сэт икс] – переместить по оси X, setY [сэт игрик] – переместить по оси Y, set [сэт] – переместить в точку с координатами…. Новые объекты this [зис] - этот, key [ки] – клавиша.

Клавиши:

- arrowLeft [эроу лэфт] кнопка налево
- arrowRight [эроу райт] кнопка направо
- arrowUp [эроу ап] кнопка наверх
- arrowDown [эроу даун] кнопка вниз

Практическая работа: программирование действий по нажатию клавиши на клавиатуре:

```
1.  @onKey = (key) =>
2.  if key == arrowRight
3.      this.move(1)
4.  end
5.  if key == arrowUp
6.      this.lift(1)
7.  end
8.  if key == arrowDown
9.      this.lift(-1)
10. end
11. if key == arrowLeft
12.     this.move(-1)
13. end
14. end
```

Работа с координатной сеткой для определения координат объекта. Отработка механики определения и использования координат объекта в программном коде. Формирование программного кода для управления объектами в игровом мире с помощью клавиатуры.

Текущий контроль по разделу «Движение объектов в игровом мире» **(2 часа,** Приложение №7).

Раздел 7. Коллизии объектов, первые игровые миры (34 часа).

Тема 7.1 Разные типы объектов в среде программирования, простые взаимодействия по коллизии **(4 часа)**.

Теория: знакомство с понятием «спрайт» в игровой среде. Простые события спрайтов. Новые понятия:

- Тайл.
- Коллизия – взаимодействие.
- Обработчик событий.

События (events):

- @onKey [онки] – по нажатию клавиши
- @onCollide [онколайд] – по столкновению

Практическая работа: формирование программного кода для обработки события простого столкновения спрайтов. Закрепление умения формировать программный код для управления объектами в игровом мире с помощью клавиатуры.

Тема 7.2 Изменение свойства видимости спрайта при коллизии **(2 часа)**.

Теория: свойство объекта setVisible [сэтвизибл] – установить видимость.

Практическая работа: настройка коллизий объектов по образцу

```
@onCollide = (obj) =>
  if obj.name == "Jason"
    this.setVisible(false)
  end
end
```

Тема 7.3 Движение объектов как сдвиг по осям координат **(4 часа)**.

Теория: команда glide [глайд] – скользить (glideX – скользить по оси X, glideY – скользить по оси Y). Свойство объекта glideInProgress [глайд ин прогрес] – глайд исполняется.

Практическая работа: программирование перемещения спрайтов с использованием команды «скольжение».

Тема 7.4 Знакомство со счетом и счетчиками в игровом мире **(4 часа)**.

Теория: ознакомление с понятием «счётчик» в игре. Ознакомление с базовыми принципами балансирования компьютерной игры. Ознакомление с базовыми принципами формирования системы наград и проигрышей в компьютерной игре. Новая команда destroy [дистрой] – исчезновение. Знаки: ==, +=, -=, <=, >=. Счетчик: counter.value [каунтер вэлью] – значение счетчика.

Практическая работа: формирование программного кода для создания счетчика в игровом мире.

Тема 7.5 Знакомство со счетом и таймером в игровом мире **(4 часа)**.

Теория: знакомство с понятием «таймер» в игре. Событие (events) @onTimerComplete [он таймер комплит] – по завершении таймера. Счет: counter.value [каунтэр вэлью] значение счетчика, reset [ресэт] обнулить и перезапустить. Таймер: set (time) [сэт тайм] установить время, start [старт] запустить таймер.

Практическая работа: программирование таймера в игре.

Тема 7.6 Текстовые сообщения в игре **(4 часа)**.

Теория: ознакомление с концепцией «обмен сообщениями» спрайтов. Ознакомление с типами текстов и местами расположения в игровом мире. Команд: setText [сэт текст] - отобразить текст, popUp() [поп ап] – показать текст, attachTo() [атач ту] присоединить к спрайту.

Практическая работа: формирование программного кода для создания обмена сообщениями спрайтов в компьютерной игре.

Тема 7.7 Формирование программного кода для управления объектами в игровом мире с помощью мыши **(10 часов)**.

Теория: понятия «Препятствие» и «Противник». Событие (events) @onUpdate [онапдейт] – автоматическая проверка. Новые команды tween [твин] – двигаться между (сокр.от англ. between) и tweenCircle [твин сёкл] – двигаться по кругу. Ознакомление с режимом игры «супер мод». Событие (events) @onClick [онклик] – по клику мыши. Оператор переменной var global.superMode [глобал супер мод] – супер режим. Знакомство с понятием «сторителлинг».

Практическая работа: программирование коллизий с препятствиями и противниками. Формирование умения написать короткий сюжет для игры.

Текущий контроль по разделу «Коллизии объектов, первые игровые миры» **(2 часа,** Приложение №7**).**

Раздел 8. Разработка и защита итогового проекта (14 часов).

Тема 8.1 Создание своей игры **(10 часов)**.

План работы над проектом:

1. Разработка сюжетной линии, выбор жанра будущей игры.
2. Выбор главного персонажа.

3. Проработка «бэкграунда» (фоновой картинки).
4. Настройка анимации и программирование поведения главного персонажа.
5. Загрузка вспомогательных персонажей — врагов, препятствий и т.д,
6. Программирование персонажей, событий и коллизий.
7. Настройка счетчиков и таймера.
8. Вывод количества жизней и собранных бонусов на экран.
9. Настройка диалогов персонажей.
10. Отладка и тестирование на ошибки.

Тема 8.2 Защита проектов **(4 часа).**

План защиты проекта:

1. План проекта.
2. Цель игры.
3. Сюжет игры – история.
4. Главный герой – кто он и что умеет делать.
5. «Враги» и объекты-помощники – траектории движения или моменты появления.
6. Игровой баланс: расчет времени и баллов.
7. Тексты в игре: вводный текст, завершающий текст.
8. Схема уровня с расстановкой спрайтов и тайлов.
9. Программирование – реализация.
10. Тестирование и отладка.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Помещения для аудиторных и практических занятий обучающихся должны быть оснащены персональными компьютерами с возможностью подключения к сети Интернет, доской с маркером и интерактивной доской.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Вордерман, К. . Программирование для детей. Иллюстрированное руководство по языкам Scratch и Python К. Вордерман, Ш. Макаманус, Д. Вудкок; Ред: Голиков Д. В.. – : Манн, Иванов и Фербер, 2019. – 224 с.
2. Воронин, И. . Программирование для детей. От основ к созданию роботов И. . Воронин, В. . Воронина. – : Питер, 2018. – 192 с.
3. Морган, Н. JavaScript для детей. Самоучитель по программированию Н. Морган. – : Манн, Иванов и Фербер, 2018. – 289 с.
4. Уитни, Д. . Программирование для детей. Учимся создавать сайты, приложения и игры. HTML, CSS и JavaScript Д. . Уитни. – : Питер, 2020. – 208 с.

5. Спрол, А. Думай как программист. Креативный подход к созданию кода. C++ версия А. Спрол; Переводчик: Райтман М. А. – : Бомбора, 2018. – 272 с.

6. Сухин, И. Г. 800 логических и математических головоломок И. Г. Сухин. – 2018 : Издательство АСТ, 2018. – 256 с.

7. Филиппов, С. А. Уроки робототехники. Конструкция. Движение. Управление С. А. Филиппов. – : Лаборатория знаний, 2018. – 193 с.

Электронные ресурсы:

1. Международный образовательный портал некоммерческой организации Code.org [Электронный ресурс]. - Режим доступа https://code.org/ - свободный (2013)

2. Онлайн-платформа для обучения детей основам программирования [Электронный ресурс]. - Режим доступа https://codewards.ru свободный (2016)

3. Урок цифры — всероссийский образовательный проект в сфере цифровой экономики [Электронный ресурс]. - Режим доступа https://урок-цифры.рф/, свободный (2015)

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Киселёв Иван Михайлович, педагог дополнительного образования, преподаватель по направлениям «Кодвардс» и «Креативное программирование в Scratch 3.0 и Алгоритмике».

7. МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ПО РАЗДЕЛАМ ПРОГРАММЫ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Приложение №1

Текущий контроль по разделу «Первый опыт программирования на платформе https://studio.code.org/»

Теоретическое тестирование:

Вопрос 1. Если повернуть стрелку на 90 градусов по часовой стрелке, то на какую цифру она укажет?

1. На цифру 3
2. На цифру 8
3. На цифру 9

Вопрос 2. Кто в данном диалоге будет говорить первым?

1. Волк
2. Заяц
3. Оба одновременно

Вопрос 3. При выполнении какого скрипта спрайт совершит один полный оборот вокруг своей оси?

Вопрос 4. Объясните, что делает программа:

1. Вращает мотор «сюда» 2 секунды со скоростью 20
2. Вращает мотор «сюда» 2 секунды со скоростью 2
3. Вращает мотор «туда» 20 секунд со скоростью 20
4. Вращает мотор «сюда» 20 секунд со скоростью 20

Вопрос 5. В чем отличие двух программ?

1. Программы работают одинаково
2. Первая программа вращает мотор 10 секунд, вторая 1 секунду
3. Первая программа не останавливает мотор через 1 секунду, вторая останавливает

4. Первая программа останавливает мотор после 1 секунды, вторая не останавливает

Вопрос 6. В чем отличие двух программ?

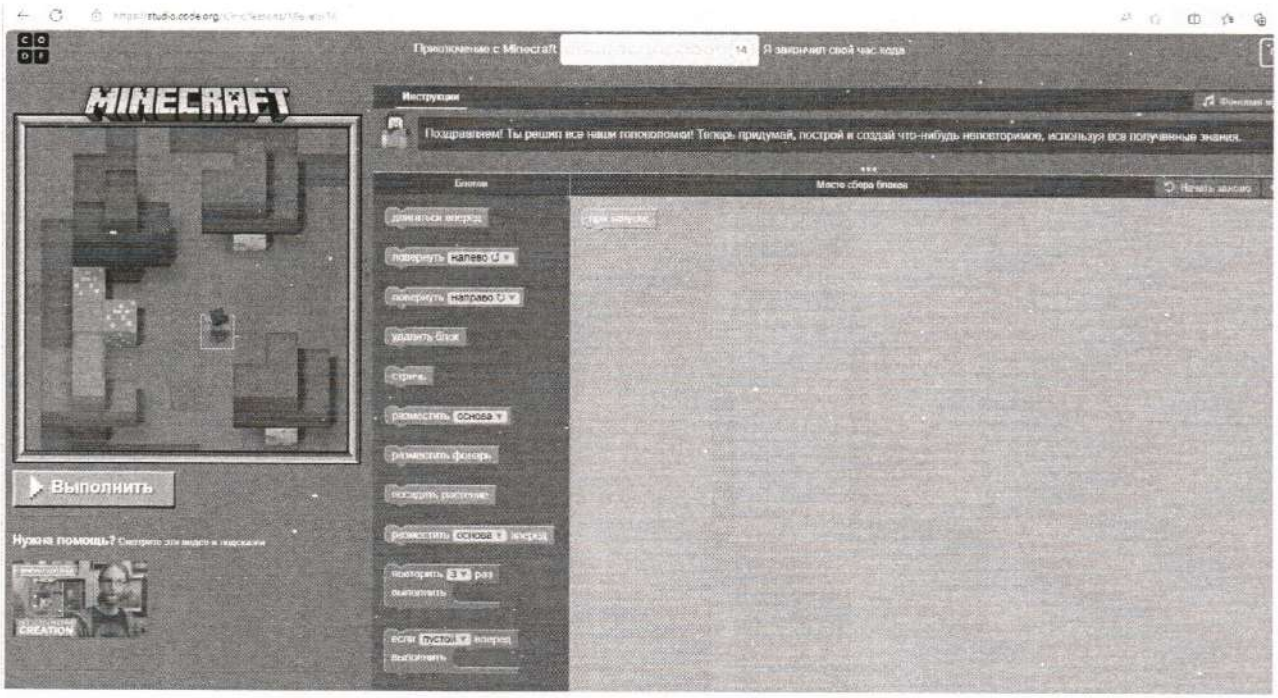
1. Обе программы работают 100 секунд, но скорость меняется только во второй программе
2. Скорость несколько раз меняется только в первой программе
3. Ни в чем
4. Скорость несколько раз меняется только во второй программе

Вопрос 7. В какой программе мотор остановится раньше и почему?

Вопрос 8. Какая из программ закончит выполняться раньше?

1. Первая
2. Вторая
3. Программы закончат работать одновременно
4. Программы никогда не закончат свою работу

Практическое задание: придумать, построить и создать что-нибудь неповторимое, используя все полученные знания. Оценивается оригинальность построенного мира и сложность программного кода.



Приложение №2

Текущий контроль по разделу «Введение в игровой мир Кодвардс»

Теоретическое тестирование:

Вопрос 1. Напиши, что обозначают команды

Move_____ Right_____

Rotate_____ Load_____

Left_____ Put_____

Вопрос 2. Тебе нужно купить в магазине хлеб. Пронумеруй шаги алгоритма этого процесса в нужном порядке.

Войди в магазин.

Пройди на кассу.

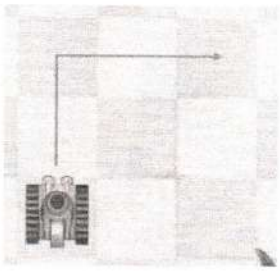
Пройди к прилавку с хлебом.

Расплатись за покупку.

Положи его в корзину.

Выбери подходящую булку хлеба.

Вопрос 3. По какой программе робот проделает путь, показанный на рисунке?



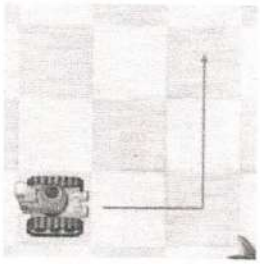
1.
robot.move 2
robot.rotate right
robot.move 1
robot.move 1

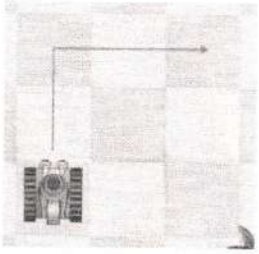
2.
robot.move 2
robot.rotate left
robot.move 2

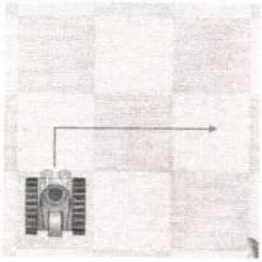
3.
robot.move 2
robot.move 2
robot.rotate right

Вопрос 4. Какая картинка соответствует программе:

robot.move 2
robot.rotate left
robot.move 2







Вопрос 5. Отметь в списке те свойства, которым обладает объект на картинке.

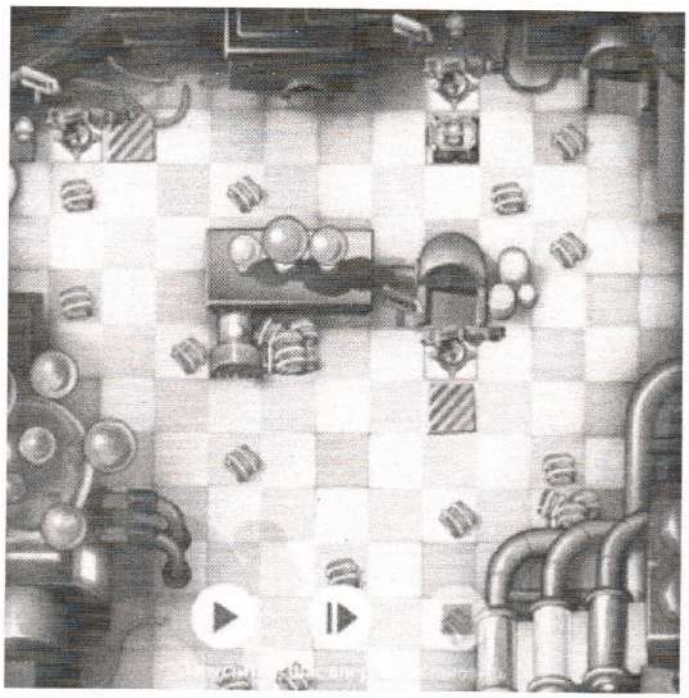
Сделан из металла
Может летать
Красного цвета
Сделан из картона
Желтого цвета
Перевозит людей

Вопрос 6. Нужно перевезти от погрузчика (loader1) топливный элемент к другому погрузчику (loader0) и разгрузить его. Программа для этого слишком большая и сложная. Нужно переписать её, чтобы путь был более коротким и правильным.

Исходный код:	
1. truck.rotate right	1.
2. truck.move 1	2.
3. truck.rotate right	3.
4. truck.move 7	4.
5. truck.rotate right	5.
6. truck.move 4	6.
7. truck.rotate right	7.
8. truck.move 1	8.
9. truck.rotate left	9.
10. truck.move 2	10.
11. loader1.rotate right	11.
12. loader1.rotate right	12.
13. loader1.load	13.
14. loader1.rotate right	14.
15. loader1.rotate right	15.
16. loader1.put	16.
17. truck.rotate right	17.
18. truck.rotate right	18.
19. truck.move 2	19.
20. truck.rotate right	20.
21. truck.move 1	
22. truck.rotate left	
23. truck.move 4	
24. truck.rotate left	
25. truck.move 7	
26. truck.rotate left	
27. truck.move 1	
28. loader0.rotate right	

29. loader0.load 30. loader0.rotate right 31. loader0.rotate right 32. loader0.put	21. 22. 23.

Практическое задание (на платформе Кодвардс): сначала нужно разгрузиться у погрузчика (loader0). Затем забрать топливный элемент у погрузчика (loader1) и снова привезти его и разгрузить у погрузчика (loader0). Код нужно написать самому.



Приложение №3

Текущий контроль по разделу «Повторяющиеся действия в программе. Циклы»

Теоретическое тестирование:

Вопрос 1. Ваш друг использовал циклы, чтобы оптимизировать программу (сделать программу короче). Выбери правильный вариант программы.

```
robot.move 5
robot.rotate left
robot.move 5
robot.rotate left
robot.move 5
robot.rotate left
```

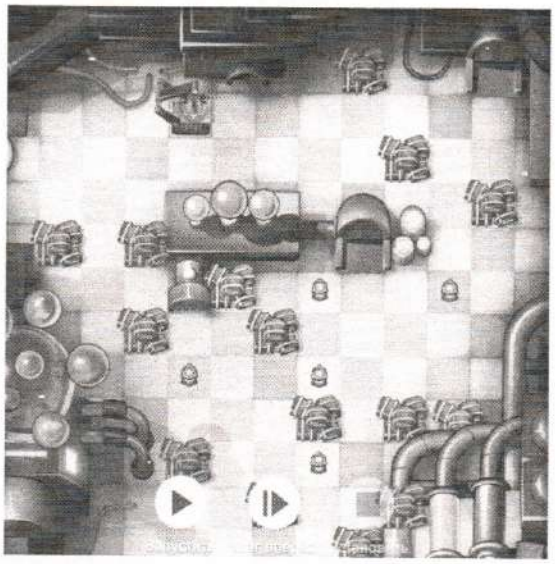
1.
```
loop 3
 robot.move 5
```

2.
```
loop 2
 robot.move 5
```

3.
```
loop 4
 robot.move 5
```

robot.rotate left end	robot.turn left end	robot.rotate left end

Вопрос 2. Нужно собрать все топливные элементы. В программе точно есть ошибки. Найди и исправь их.

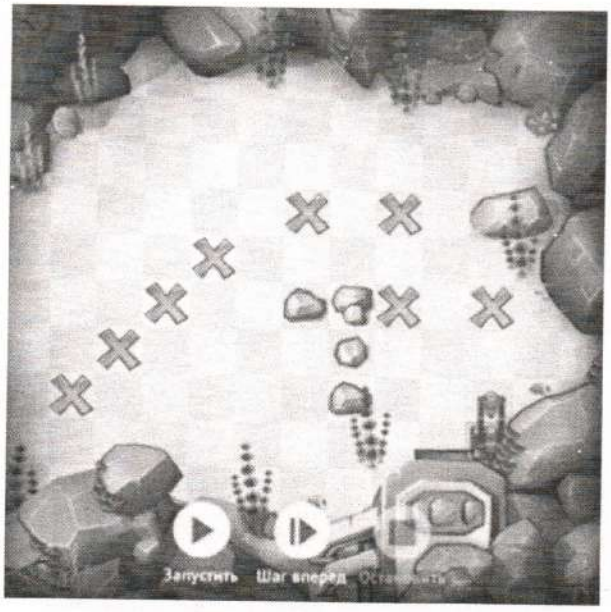


```
1. picker.move 6
2. picker.rotate right
3. picker.move 3
4. loop 3
5.         picker.load
6.         picker.move 1
7.         picker.rotate right
8.         picker.move 1
9.         picker.load
10.        picker.move 1
11.        picker.rotate right
12.        picker.move 1
13. end
14. picker.rotate left
15. picker.move 2
16. picker.rotate right
17. picker.move 4
18. picker.rotate left
19. picker.load
```

Вопрос 3. Нужно добыть все кристаллы, отмеченные оранжевым крестиком. Найди ошибки в коде.

```
1. loop 3
2.      driller.rotate right
3.      robot.move 2
4.      driller.drill
5. end
6. loop 3
7.      driller.rotate left
8.      driller.move 1
9.      driller.move 1
10.     driller.rotate left
11.     driller.move 2
12. end
```

Практическое задание (на платформе Кодвардс): твоя задача — добыть все кристаллы, отмеченные оранжевым крестиком. Нужно самому написать оптимальный код.



Приложение №4

Текущий контроль по разделу «Переменные и сенсоры в Кодвардсе»

Теоретическое тестирование:

Вопрос 1. Дай определение термину «переменная»_____

_____

_____

_____

_____

_____

_____

_____

Вопрос 2. Посмотрите на эту запись и напишите, что произойдёт:

```
var dist = 5
robot.move dist
```

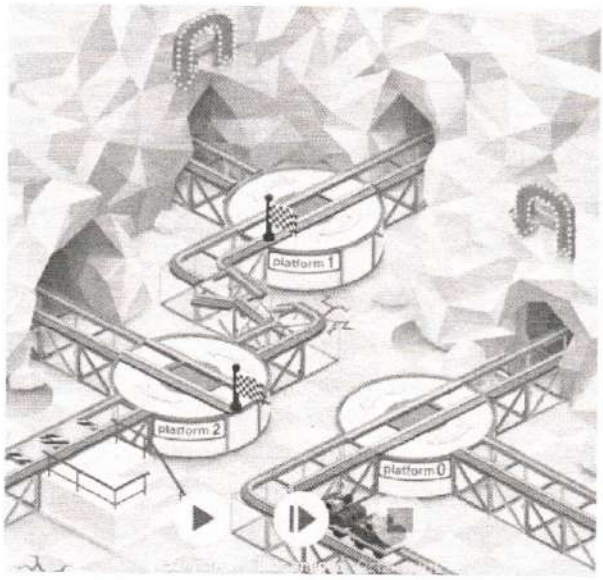
_____

_____

_____

_____

Вопрос 3. На пути два тоннеля. Используй сенсор на въезде в каждый из них. Это поможет довезти посетителей до выхода. Найди ошибки в программном коде:




```
1. var dist=cart.stepToFlag
2. cart.move distr
3. platform1.rotate left
4. var dist=cart.stepToFlag
5. cart.move 4
```

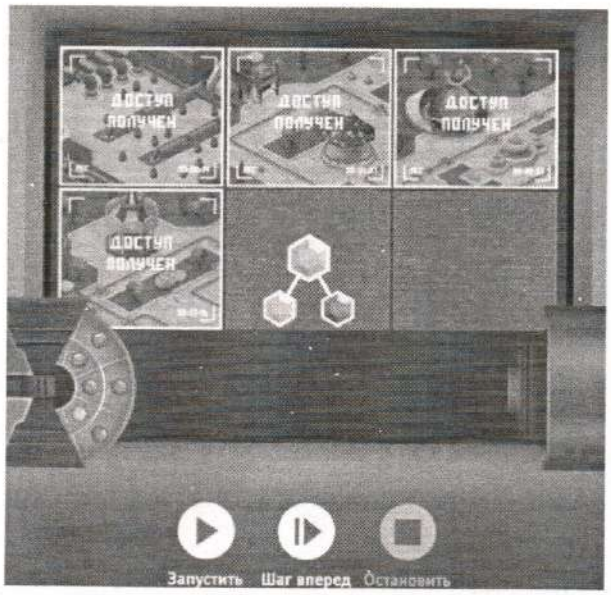
6. platform2.rotate right
7. cart.move 2

Вопрос 4. Накормить животных нужно в двух вольерах. В каждом есть детёныш. Раздай игрушки детенышам и накорми их родителей. Найди и исправь ошибки в программе:

```
1. robot.rotate left
2. robot.rotate right
3. robot.move 1
4. robot.rotate right
5. robot.callAnimals
6. while platform0.isPressed==true
7.          if robot.animalIsBaby==true
8.                    robot.take food
9.                    robot.put
10.         else
11.                   robot.take toy
12.                   robot.put
13.         ent
14.end
15.robot.rotate left
16.robot.move 1
17.robot.rotate left
18.robot.callAnimals
19.while platform1.isPressed == true
20.         if robot.animalIsBaby == true
21.                   robot.take toy
22.                   robot.load
23.         else
```

```
24.             robot.take food
25.             robot.put
26. end
27.
```

Практическое задание (на платформе Кодвардс): нам нужен оранжевый луч. Какие цвета нужно использовать, чтобы получился оранжевый? Верно! Сперва выстави желтый, потом красный. Код нужно написать самостоятельно.



Приложение №5

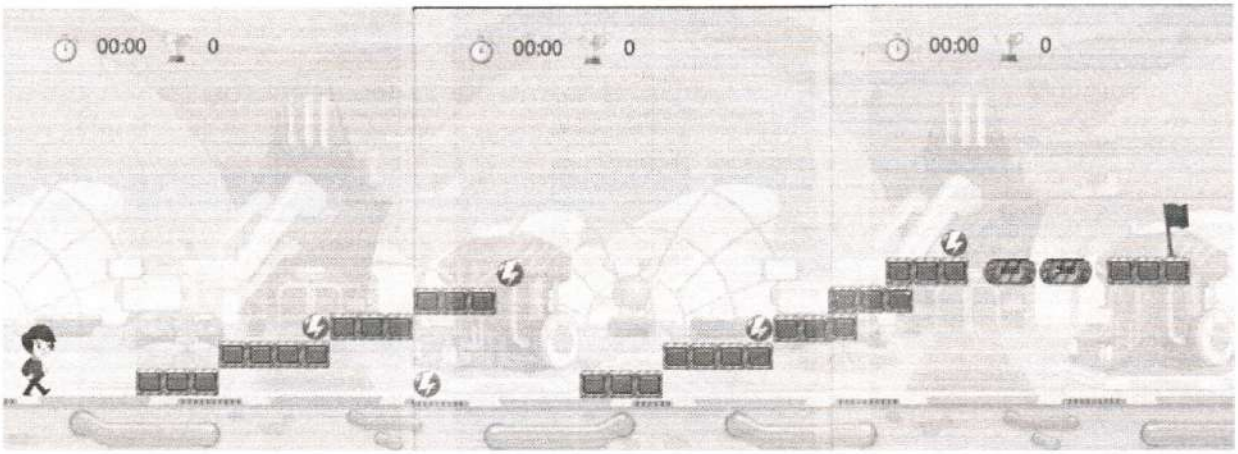
Текущий контроль по разделу «Введение в проектную деятельность»

Практическое задание (на платформе Кодвардс): плыви, ориентируясь по датчикам, если увидишь впереди препятствие – следуй за рыбками. В этом тебе поможет конструкция IF ELSE. Плыви, пока не найдёшь чекпоинт. Для этого тебе пригодится WHILE.

Текущий контроль по разделу «Движение объектов в игровом мире»

Теоретическое тестирование:

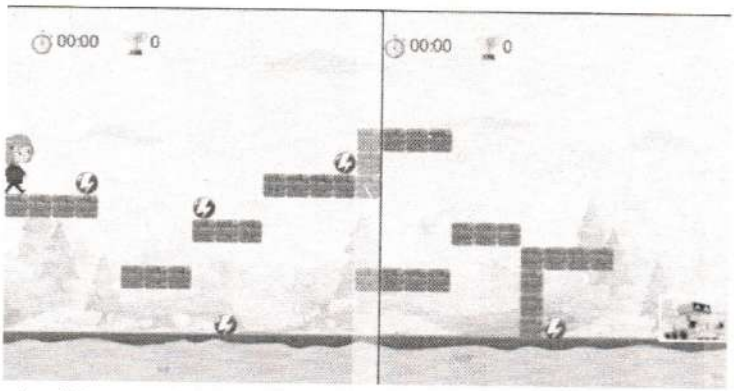
Вопрос 1. В правом столбце таблицы своими словами опиши, по какому принципу работает программный код и что будет происходить с героями проекта и при его запуске.



Jason 1. @onKey = (key) => 2. if key == arrowRight 3. this.move(2) 4. end 5. if key == arrowLeft 6. this.move(-2) 7. end 8. if key == arrowUp 9. this.jump() 10. end 11. end	
powerup1/2/3/4/5 1. @onCollide = (obj) => 2. if obj.name == "Jason" 3. this.destroy() 4. counter.value += 2 5. end 6. end	
Platform 1. @onUpdate = () => 2. if counter.value >= 10 3. this.setVisible(true) 4. else 5. this.setVisible(false) 6. end 7. end	

Flag 1. @onCollide = (obj) => 2. if obj.name == "Jason" 3. this.destroy() 4. end 5. end	
timer 1. this.set(20) 2. this.start()	

Вопрос 2. Исправь ошибки в коде уровня. Активируй видимость робота. Чтобы он появился на локации, Прокси должна собирать энергию, набрать больше 20. Каждая энергия дает +=5. Время на прохождение – 20 секунд.



1. @onKey
2. if key == arrowRight
3. this.move(20)
4. end
5. if key == arrowLeft
6. this.glide(-2)
7. end
8. if key == arrowUp
9. this.lift()
10. end
11.end

ally

1. @onUpdate = () =>
2. if counter.value = 20
3. this.setVisible(false)
4. else
5. this.setVisible(true)
6. end
7. end

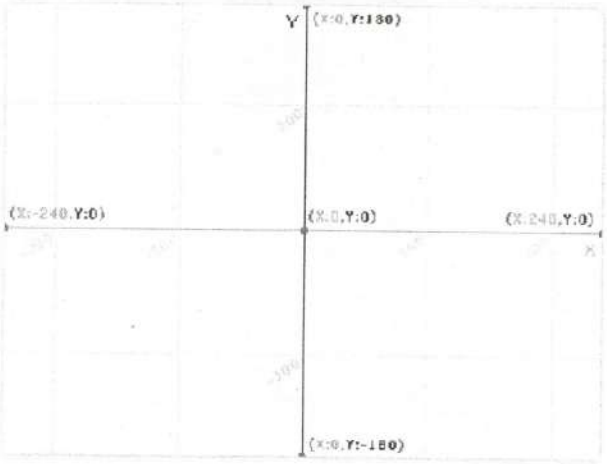
timer

1. this.set(20)
2. this.stop()

powerUp1/2/3/4/5

1. @onCollide = (obj) =>
2. if obj.name == "Proxy"
3. this.setVisible(true)
4. counter.value -= 5
5. end
6. end

Вопрос 3. Для чего нужна система координат?

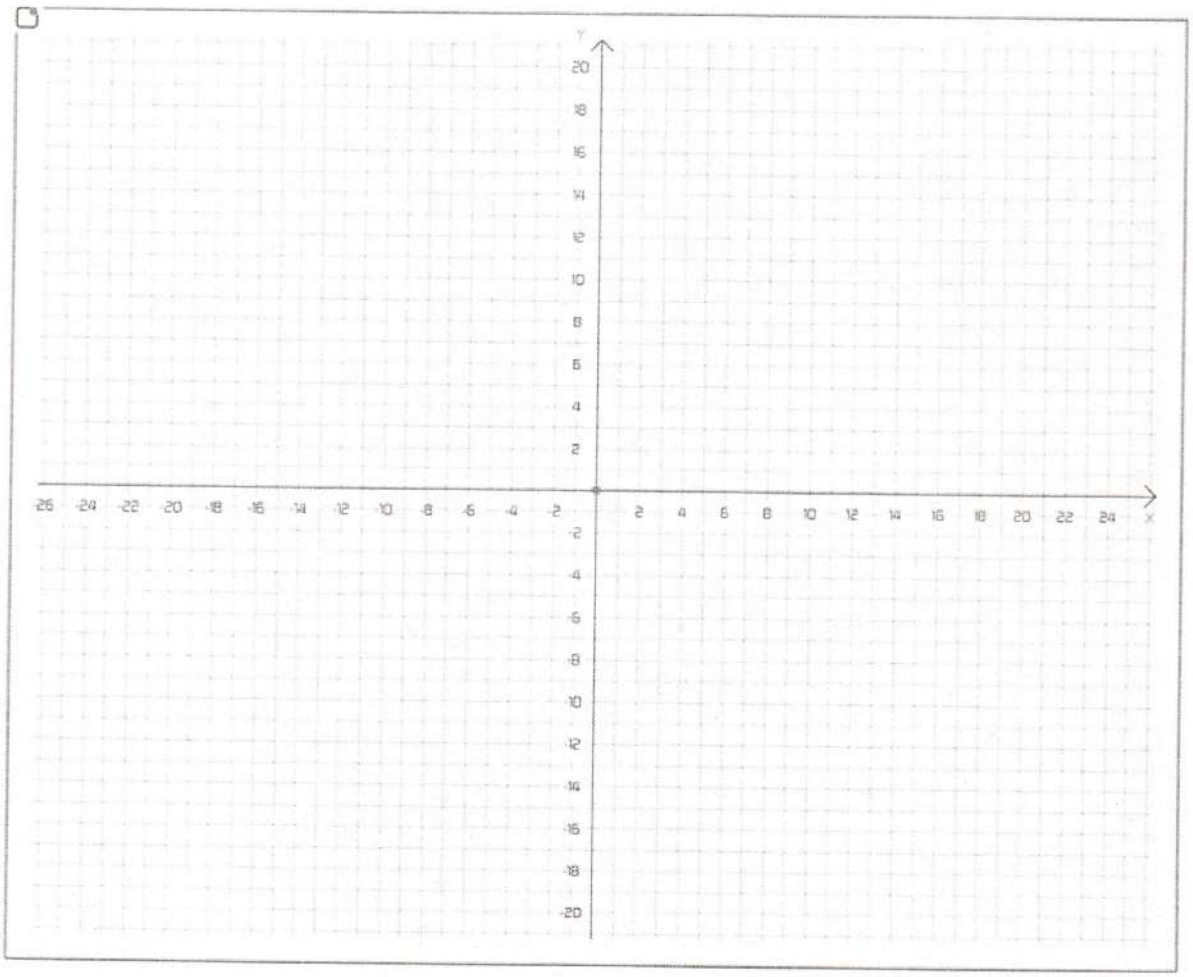


1. Чтобы поставить спрайт в точку

2. Чтобы найти точку

3. Чтобы определить положение спрайта на сцене

Практическое задание (на листе): отметь на координатной плоскости точки и получи изображение **одного** (на свой выбор) животного, соединив эти точки.



Верблюд

1) (- 9; 6), (- 5; 9), (- 5; 10), (- 4; 10), (- 4; 4), (- 3; 4), (0; 7), (2; 4), (4; 7), (7; 4), (9; 3), (9; 1), (8; - 1), (8; 1), (7; 1), (7; - 7), (6; - 7), (6; - 2), (4; - 1), (- 5; - 1), (- 5; - 7), (- 6; - 7), (- 6; 5), (- 7;5), (- 8; 4), (- 9; 4), (- 9; 6).
2) Глаз: (- 6; 7).

Конь

1) (14; - 3), (6,5; 0), (4; 7), (2; 9), (3; 11), (3; 13), (0; 10), (- 2; 10), (- 8; 5,5), (- 8; 3), (- 7; 2), (- 5; 3), (- 5; 4,5), (0; 4), (- 2; 0), (- 2; - 3), (- 5; - 1), (- 7; - 2), (- 5; - 10), (- 2; - 11), (- 2; - 8,5), (- 4; - 8), (- 4; - 4), (0; - 7,5), (3; - 5).
2) Глаз: (- 2; 7).

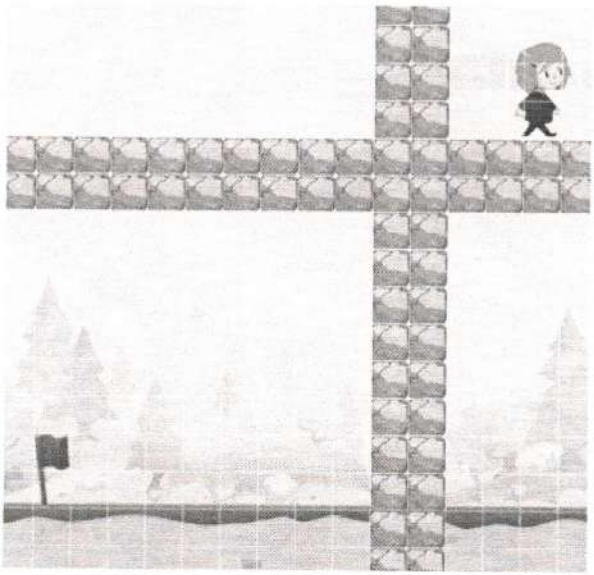
Слоник

1) (2; - 3), (2; - 2), (4; - 2), (4; - 1), (3; 1), (2; 1), (1; 2), (0; 0), (- 3; 2), (- 4; 5), (0; 8), (2; 7), (6; 7), (8; 8), (10; 6), (10; 2), (7; 0), (6; 2), (6; - 2), (5; - 3), (2; - 3).

2) (4; - 3), (4; - 5), (3; - 9), (0; - 8), (1; - 5), (1; - 4), (0; - 4), (0; - 9), (- 3; - 9), (- 3; - 3), (- 7; - 3), (- 7; - 7), (- 8; - 7), (- 8; - 8), (- 11; - 8), (- 10; - 4), (- 11; - 1), (- 14; - 3), (- 12; - 1), (- 11;2), (- 8;4), (- 4;5).

3) Глаза: (2; 4), (6; 4).

Практическое задание (на платформе Кодвардс): запрограммируй кнопки так, чтобы можно было довести Прокси до флага. Напиши код самостоятельно.



Приложение №7

Текущий контроль по разделу «Коллизии объектов, первые игровые миры»

Теоретическое тестирование:

Вопрос 1. Напиши, что обозначают программные коды:

@onKey _____

@onCollide_____

```
if obj.name == "Jason"
      this.setVisible(false)
end
```

_____

glide_____

glideX_____

_

glideY_____

de-
stroy_____

@onTimerComplete_____

reset_____

start_____

setText_____

popUp_____

attachTo_____

@onUpdate_____

tween_____

tweenCircle_____

@onClick_____