Глава 4 АЛГОРИТМЫ И ИСПОЛНИТЕЛИ

§ 22

РАБОТАЕМ СО СЧЁТЧИКАМИ



 Я вспоминала алгоритм кипячения воды в чайнике написала его более крупными командами:

> Начало Возьми чайник. Наполни чайник. Включи чайник. Жди, пока вода закипит. Конец

Алгоритм верный, но меня вот что смущает. В чайнике уже может быть вода. Тогда зачем нам наливать воду?

Интересный вопрос. Давайте разберёмся.

ЗАДАНИЕ

Могут быть две ситуации: 1) чайник пустой; 2) чайник полный.

Тогда выполнение алгоритма может пойти разными путями, в зависимости от условия. Условие формулируется так: «чайник пустой».

Начало Возьми чайник. Если чайник пустой То наполни чайник. Включи чайник. Жди, пока вода закипит. Конец

— Я вижу в алгоритме слова «если...то». Вспомнил. Мы уже встречались с такими алгоритмами в втором классе. Это разветвлённые алгоритмы. Ещё их называют алгоритмами с условием.



Правильно!

Теперь представим алгоритм наглядно — в виде блок-схемы.





Найдите в блок-схеме условие.



Попытайтесь дополнить алгоритм в рабочей тетради. Включите в него команды заварки чая в чашке. Вам надо добавить ещё одно условие. Сформулируем его так: нужен сладкий чай.

РАБОТА НА КОМПЬЮТЕРЕ

Вы уже знакомы с цветными счётчиками. Вы могли видеть, как при их изменении цветное число перелетает от предмета или персонажа к числу в верхнем правом углу экрана и добавляется к этому числу.



Откройте любой ранее сохранённый мир, в котором есть цветные счётчики. В параметрах мира выберите «Вывод счёта» для того счётчика, который используется, и попробуйте поменять параметры вывода счётчика этого цвета.

Кроме цветных счётчиков в Лаборатории игр Коду есть счётчики, которые обозначаются латинскими буквами от А до Z



и от а до z. Счётчики от A до Z — общие для всех персонажей. Если один персонаж установит в счётчик A число 5, то другой персонаж увидит там это число. Счётчик a свой отдельный у каждого персонажа.

Такие счётчики могут управлять работой программ, а значит, поведением персонажей или предметов. Например, в зависимости от результатов проверки таких счётчиков, можно:

- 1) выполнять или не выполнять какое-то действие;
- выбирать один вариант из нескольких вариантов действия — действие с номером;
- менять свойства объекта: цвет, настроение, размер, скорость движения, скорость поворота, расстояние удерживания собранных объектов, вид неба, освещение.

Конечно, вид неба и освещение — это свойства не одного объекта, а всего мира. Например, если у нас есть счётчик времени суток, который связан с каким-то объектом этого мира, то в зависимости от того, который час, можно менять вид неба и освещение. В цветных или буквенных счётчиках могут храниться не только подсчёты чего либо: очков, времени, собранных предметов. В них можно хранить и номера: номер действия, номер фразы, номер цвета, номер размера. Поэтому часто их называют не счётчиками, а переменными. С переменными вы уже знакомы. Слово «переменная» означает, что числа в ней в ходе игры могут меняться.

Все свойства объекта, которые могут меняться в ходе игры, — это тоже переменные.

Какие переменные — свойства объектов вы можете назвать?

Не очень удобно при создании игры не видеть значения буквенных переменных (то есть не видеть тех чисел, которые в них находятся). Для удобства можно временно записывать буквенную переменную в неиспользуемую цветную и видеть, как меняется её значение.

Теперь познакомимся с несколькими примерами применения цветных и буквенных переменных.

79

Внесём изменения в игру, в которой летающая тарелка разбрасывает яблоки, а игрок может управлять байкером, который при соприкосновении с яблоком съедает его. Когда байкер съедает 5 яблок, игра заканчивается победой игрока.



Откройте мир с этой игрой, если он был сохранён, или создайте его по описанию из параграфа 19.

Изменения:

- Пусть летающая тарелка разбрасывает не только красные, но и синие яблоки.
- Пусть красный счётчик увеличивается на 1 при съедании красного яблока и уменьшается на 1 при съедании синего яблока.
- Для каждого значения красного счётчика от 1 до 4 установим новый размер байкера. Пусть байкер растёт при поедании красных яблок и уменьшается при поедании синих. Как и раньше при значении красного счётчика, равного 5, игра закончится победой игрока.

Красные яблоки падают каждые 5 секунд, пусть синие падают каждые 7 секунд. Для этого добавьте тарелке такой блок «Когда»:





Справа от этого блока «Когда» нужно добавить блок «Делать»:



Измените команду байкера, описывающую увеличение красного счётчика. Пусть теперь она выполняется только для красных яблок:







Скопируйте команду: щёлкните на её номере правой кнопкой мыши и выбери-«Скопировать строку». Затем, шёлкте нув на свободном месте правой кнопкой мыши, выберите «Вставить строку». Появится такая же новая команда. В ней замените карточку «красный» после карточки «яблоко» на «синий», а карточку «вычесть». Для «очки» на замены назаменяемую карточку левой жимайте на кнопкой мыши И держите кнопку нажатой. появятся варианты пока не ДЛЯ замены.

Впрочем, можно не копировать, вставлять и исправлять команды, а просто ввести новые:







Теперь добавьте команды изменения размера байкера:



Подскажем, где найти некоторые из этих команд.

При создании новой команды с полями «Когда» и «Делать» выбор поля «Когда» приводит к появлению круга выбора условий.

При выборе поля «счёт» карточка «счёт» добавляется в команду, а нажатие на этой карточке приводит к появлению круга, позволяющего выбрать карточку «нет» или любой цвет или букву счётчика. Карточка «нет» означает действия при невыполнении условия.



После указания цвета или буквы счётчика снова нажимаем в круг выбора операции сравнения. «Выше»





На круге выбора действий (смотри параграф 17) есть карточка settings 🐼. При выборе этого поля появляется круг выбора изменяемых величин — переменных.



В этом окне карточка scale означает плавное изменение размера, а карточка scale(instant) — мгновенное изменение размера.

Новое значение не видно в команде. Посмотрите на четыре расположенные подряд команды изменения размера в зависимости от значения красного счётчика (страница 83). У них на карточке «Делать» видно одно и то же действие. Но в окне выбора нового значения для первой команды выбрано число полтора (1,5), для второй два (2,0), для третьей — два с половиной (2,5) и для четвёртой — три (3,0). Считается, что вначале у всех объектов размер равен единице (1,0). Новый размер показывает, во сколько раз вырастет объект.

Сохраните этот мир под другим названием и внесите в его описание сделанные изменения.

Если вы быстро справились с работой, то поэкспериментируйте.

Сделайте так, чтобы синие яблоки падали так же часто, как и красные, и попробуйте некоторое время есть только синие яблоки. Посмотрите, что будет с красным счётчиком. А теперь поешьте красных яблок. Как меняется красный счётчик? Выйдите из игры с помощью клавиши Esc, не дожидаясь победы.



Можете попробовать самостоятельно создать другой мир, используя буквенную переменную (счётчик, обозначенный буквой).

- 1. Создайте новый мир, в котором будет летать тарелка, разбрасывая монетки.
- 2. Персонаж коду, управляемый игроком, будет собирать монетки. Как бы «съедать», ведя счётчик числа монеток.
- Если число монеток, которые собрал коду, ноль, то коду будет белого цвета. Если монеток от 1 до 4, то он будет жёлтого цвета, если от 5 до 9, то оранжевого, если 10 или больше, то красного.

подведём итоги

Вы вспомнили, что такое разветвлённый алгоритм и условие.

Вы поработали со счётчиками (переменными) в Лаборатории игр Коду.



ВОПРОСЫ И ЗАДАНИЯ

- 1. Что такое условие в алгоритме?
- 2. Как вы понимаете, что такое переменная?

§ 23 СОЗДАЁМ НОВЫЕ ОБЪЕКТЫ В ХОДЕ ИГРЫ

Я думаю, что ветвления
 бывают не только такие, как
 в алгоритме кипячения воды
 в чайнике. Там мы отступали
 в сторону от прямого порядка
 действий только по ветви «Да»



действий только по ветви «Да» (если чайник пустой). По ветви «Нет» мы не выполняли никаких дополнительных действий.

Верно, Маша. В алгоритме кипячения воды у нас было ветвление вида «если...то». А сейчас рассмотрим ветвление «если... то...иначе».

ЗАДАНИЕ

Рассмотрим алгоритм.



Здесь мы видим что **если** условие выполняется (ветвь «Да»), **то** выполняется одно действие, **иначе** (ветвь «Нет») другое действие:

если условие то действие 1 иначе действие 2

Обе ветви не пустые. После прохода по одной либо по другой ветви алгоритм возвращается на прямой путь. Запишите в рабочей тетради алгоритм, представленный блок-схемой, в виде текс-товых команд.



Что делает этот алгоритм?

Выполните алгоритм при:

а) А = 78 и В = 54;

б) А = 89 и В = 98.

РАБОТА НА КОМПЬЮТЕРЕ

Часто в компьютерных играх не все персонажи сразу находятся в игре. Некоторые из них создаются уже в ходе самой игры. Иногда они могут быть созданы или не созданы в зависимости от хода игры. Кроме того, в некоторых играх может создаваться много одинаковых персонажей. Такие возможности есть и в Лаборатории игр Коду. Например, можно создать во время игры целую команду байкеров!

Поведение персонажей задано набором команд и свойствами. Чтобы создавать персонажей во время игры, надо заранее описать их поведение.

В Лаборатории игр Коду это делается так.

 Создают персонаж. С помощью написания команд и выбора свойств задают его поведение.

- Указывают в свойствах, что это у нас не персонаж, а только образец, по которому во время игры можно создавать персонажей.
- В наборы команд других персонажей и предметов добавляют команды создания нового персонажа по этому образцу.

При запуске мира не будет персонажей, для которых указано, что они образцы. Мы их видим на экране только в режиме редактирования. Чтобы персонажи появились в ходе игры, кто-то или что-то должно их создать.

Попробуйте создать мир, в котором будет башня, из которой при каждом нажатии какой-нибудь клавиши (например, цифры 1) будет выезжать один байкер, а при нажатии другой клавиши (например, цифры 2) будет вылетать одна летающая тарелка. И байкеры, и летающие тарелки будут свободно передвигаться.

В Лаборатории игр Коду башня называется «замок». Сначала создайте мир с одной башней, одним байкером и одной летающей тарелкой. Для байкера и для летающей тарелки создайте одну команду без условия: «двигаться свободно».

Щёлкните на байкере правой кнопкой мыши. В появившемся меню выберите пункт «Изменить установки». Далее найдите свойство «Родитель» и щёлкните на кнопке включения



как вы включите свойство «Родитель», внизу новое свойство. Как появится под ним вы думаете, что оно означает?



Вы установили для байкера свойство «Родитель». Теперь это уже не байкер, а только образец байкера! Как изменился его внешний вил?

Сделайте летающей то же самое с тарелкой.

Переходим к командам создания байкеров и летающих тарелок. Выберите башню и перейдите в режим создания её программы. Наберите из карточек команду:



Появится круг выбора объектов для создания. Выберите пункт «родитель», затем выберите байкера.



Наберите ещё одну команду — для клавиши с цифрой 2 и создания новой летающей тарелки (она в программе названа блюдцем).

Сохраните новый мир с новым названием.

Осторожно! Специально того не задумывая, мы сделали мир, в котором непрерывно появляются новые байкеры, пока мы держим нажатой клавишу с цифрой 1, и непрерывно появляются новые летающие тарелки, пока мы держим нажатой клавишу с цифрой 2. Если объектов будет очень много, то Лаборатория игр Коду закончит работу с ошибкой. Если при этом мир не сохранён, то его последняя версия будет утеряна.

Если вы хотите, чтобы при каждом нажатии клавиш создавался ровно один персонаж, то к команде создания нового персонажа добавляйте карточку «один раз»:



Но пока что можно воспользоваться возможностью непрерывного создания новых объектов и посмотреть, как это выглядит. Если объектов становится очень много, то Лаборатория игр Коду предупреждает об этом, показывая градусник. У этого градусника при повышении числа объектов в игре цвет переходит от бесцветного к жёлтому, потом — к оранжевому, затем — к красному.



Запускайте этот мир на выполнение и наблюдайте за появлением новых байкеров и летающих тарелок.

Теперь создадим случайных персонажей.



Создайте мир, в котором кто-то или что-то через некоторые интервалы времени (одинаковые или случайные) порождает случайных персонажей из нескольких выбранных вами. Созданные персонажи пусть отправляются путешествовать. Подскажем: можно создать переменную, которая принимает случайное значение, например от 1 до 3. А команды создания каждого персонажа должны содержать условие, проверяющее значение этой переменной. Команда установки в переменную случайного значения, например от 1 до 3, выглядит так:



подведём итоги

Вы познакомились с формой ветвления «если...то...иначе».

Вы научились создавать множество одинаковых персонажей в Лаборатории игр Коду.

ВОПРОСЫ И ЗАДАНИЯ

- 1. Как вы поняли: чем различаются ветвления «если...то» и «если...то...иначе»?
- Опишите ситуацию в компьютерной игре, в которой потребуется создание одинаковых персонажей.





§ 24 ГРУППЫ КОМАНД



— У меня возникло затруднение. Я пытался записать B виде алгоритма поведение исполнителя «паром». Этот паром шесть раз В день перевозит людей через реку. Алгоритм получился правильный, но очень большой! Его выполнять. B нём трудно понимать и слишком много команд — это неэко-HOMHO.

Поможем Ивану разобраться.

ЗАДАНИЕ

Давайте посмотрим, что же получилось у Ивана. Перед нами только начало его алгоритма. Что мы видим на нём? Что сделает исполнитель?



Теперь давайте вспомним, что нам необходимо обеспечить работу парома в течение всего дня. Переправиться через реку паром должен 6 раз.

Оцените примерно, сколько места займёт полный алгоритм шести переправ.



Внимание! Передаю информацию.

Для того чтобы компактно записать некоторое количество повторений, используется цикл. Алгоритм с циклом называется циклическим алгоритмом.



Наш алгоритм с помощью цикла запишется так:



Заметим, что цикл содержит условие и действия, которые надо повторять.

Запишем алгоритм в виде текстовых команд:

Начало Повтори 6 раз: Пришвартуйся к пристани. Погрузи и выгрузи пассажиров. Отшвартуйся от пристани. Переправься через реку. Конец цикла Конец

Попробуйте составить в рабочей тетради алгоритм мытья посуды. Пусть вам нужно вымыть подряд 20 тарелок.



РАБОТА НА КОМПЬЮТЕРЕ

У каждого объекта в Лаборатории игр Коду есть свой набор команд.

Каждая команда состоит из условия и действия. В действие могут входить целиком другие команды, которые будут выполняться только тогда, когда выполнится общее для них действие. При записи такие команды сдвигаются вправо:



Чтобы начало выполняться действие 2, надо, чтобы были выполнены условие 1 и условие 2. Если условия нет, то сразу выполняется действие. Например, если нет условий 2 и 3, то при выполнении условия 1 выполняется действие 1, потом действие 2, а потом — действие 3.



Сколько условий на рисунке? Сколько действий? Назовите все условия и все действия.



Но группировать команды можно, не только сдвигая их вправо. В Лаборатории

игр Коду можно размещать группы команд на отдельных страницах. У каждого предмета или персонажа всего может быть 12 страниц с командами.

Когда вы собираете из карточек с условиями и действиями команду за командой, в верхней части экрана находится переключатель страниц.

Откройте любой мир или создайте новый. Выберите для любого объекта режим «Программа» и пощёлкайте левой кнопкой мыши на синих стрелочках вправо и влево рядом с буквами L и R.



Первыми всегда начинают выполняться команды на странице 1.

Есть два способа использовать команды, расположенные на других страницах: встроить их в выполняемые команды или перейти на другую страницу. Для каждого из этих способов есть своя карточка. Посмотрите на круг выбора действий и попробуйте найти на нём эти карточки.



103

При выполнении действия на карточке inline («встроить») команды с указанной страницы как будто добавляются к командам основной страницы.

Встраивание команд удобно использовать, если одни и те же действия надо выполнять в разных условиях. Если мы меняем общие действия, можно не заботиться о том, чтобы поменять их во всех местах, где они расположены. Достаточно менять их на одной странице.

Вспомните: мы говорили о том, как важно включать в игру описание цели игры и возможных действий. Этот текст должен появляться в сразу при запуске игры, а также в ходе игры при нажатии клавиши, например F1. Но ведь этот текст мы можем менять. Чтобы обеспечить его гарантированное совпадение в начале игры и по ходу игры, можно использовать встраиваемую страницу.



Откройте любой мир. Подготовьте небольшой текст с описанием этого мира. Выберите любой предмет, который не исчезает во время игры. Выберите любую свободную страницу, например страницу 2. На этой странице разместите кобез условий с карточками манду «ска-«один раз». Карточка «сказать» зать» И «Полный быть параметром должна С

экран» и текстом, который будет появляться на экране. На странице 1 разместите команду без условий с действием «встроить» с указанием страницы с командой «сказать». Ещё там разместите команду с условием «если нажата клавиша F1» и действием «встроить» с указанием той же страницы. Запустите игру на выполнение и проверьте её в работе.

При выполнении команды «переход на другую страницу» происходит смена страниц без возврата к начальной странице. Переход на начальную страницу произойдёт только в том случае, если будет выполнена команда «переход на страницу» с номером этой страницы.

Очень часто в игре происходят какието события, после чего у персонажей временно или постоянно меняется поведение. Можно сказать, что они переходят в другое состояние.

Поведение в разных состояниях описывается командами на разных страницах, а переход в другое состояние делается командой «перейти на страницу».

Создайте мир, в котором персонаж, например коду, догоняет мяч и бьёт по нему, а мяч отлетает. Но когда по вашей команде меняется состояние коду, он



по-прежнему догоняет мяч, но при касании с мячом уже сам отлетает от мяча. Запустите игру на выполнение и проверьте её в работе.

Помните, как нужно записать, что коду догоняет мяч? «Когда видишь мяч, двигаться к нему». А как записать, что при касании с мячом мяч отлетает? «Когда коснулся мяча — запуск».

Команду на смену состояния можно дать нажатием клавиш «1» и «2». Не забывайте, что своё поведение должны изменить и коду, и мяч. Это значит, что у каждого должно быть две страницы с описанием поведения в разных состояниях.

Обратите внимание: если в разных состояниях у персонажа есть что-то общее в поведении, то это общее можно описать командами на отдельной странице.

подведём итоги

Вы узнали, что такое цикл и циклический алгоритм.

Вы научились работать с группами команд в Коду.

?

вопросы и задания

- 1. Как вы понимаете, что такое цикл?
- 2. Какие способы группировки команд в Коду вы знаете?



§ 25 ЗВУКИ И МУЗЫКА

— А может цикл длиться бесконечно?



Иван, вспомни свойства алгоритма. Одно из его свойств говорит о том, что алгоритм обязательно должен привести к результату.

Если цикл повторяется бесконечное число раз и условие цикла никогда не выполнится, то результат мы не получим.

Значит, алгоритм составлен с ошибкой!

ЗАДАНИЕ

Таксист работает в течение смены, пока его выручка не составит установленной суммы. Посмотрите на алгоритм его работы.

Начало Ввести А и С. Повторять, пока А < С: Ввести В. А = А + В. Конец цикла Ехать домой. Конец

Здесь А — выручка таксиста, С установленная сумма денег, которую таксист должен набрать за смену. В — столько денег получает таксист с пассажира.



— Непонятно, что означает запись A = A + B.

Она означает, что, получив от пассажира количество денег В, таксист докладывает В к накопившейся до этого выручке А. А увеличивается на величину В. Заметим, что в этом цикле количество его повторений неизвестно. Таксист будет ездить, пока не соберёт требуемую сумму. Поэтому такой цикл называют: цикл «пока».

Запишите в рабочей тетради алгоритм работы таксиста в виде блок-схемы. Выполните алгоритм, используя следующие значения переменных:



a) A = 0, B = 400, C = 2000;

б) A = 300, B = 600, C = 3000.

РАБОТА НА КОМПЬЮТЕРЕ

Сегодня у нас будет урок-исследование. То, чему вы сегодня научитесь, сложно нарисовать или описать словами, потому что речь пойдёт о звуке и музыке. В Лаборатории игр Коду мало простых команд для работы со звуком и музыкой, но много разных звуков и мелодий.

Для того чтобы научиться добавлять в игры звуки и музыку, советуем создавать миры с простыми действиями, но с большим числом разных звуков, музыки, с их сочетанием, чередованием, одновременным звучанием. Мы только подскажем вам, какие команды запускают и останавливают звуки и музыку. Пробовать, что получается при их использовании, вы будете сами.

Ещё вам предстоит задуматься, как подбирать музыку к играм. Подумайте, какая музыка подходит для гонок, какая для поиска и собирания предметов. Подумайте, какое настроение вы хотите создать у игрока с помощью выбранной музыки.

Затем можно взять любой ранее созданный мир и добавить в него звуки и музыку.

Две основные карточки при работе со звуками и музыкой:





Они находятся в круге выбора действий: лепесток «действия», затем — «ещё».



Теперь можно щёлкать на разных лепестках, разглядывая, какая есть музыка, какие звуки и как они сгруппированы, что находится в круге «среда», а что — в круге «события».



Приступайте к исследованиям звуковых возможностей Лаборатории игр Коду. Можете делать это в паре или в небольшой группе с одноклассниками.

Удачи в исследованиях!



подведём итоги

Вы поработали с циклом в алгоритме. Вы творили, создавая музыку.



вопросы и задания

- 1. Как вы понимаете, почему бесконечный цикл не приводит к результату?
- Какие способности вы открыли в себе, создавая музыку в Лаборатории игр Коду?

§ 26 ГОНКИ

— Мне очень понравилось, что мы можем записывать повторяющиеся действия в виде цикла. Это компактно и наглядно. Можно делать красивые программы.



Потренируемся составлять алгоритмы с циклами.

ЗАДАНИЕ

Возьмите лист в клетку и отступите от верхнего левого угла две клетки вниз и две вправо. Поставьте там точку.

Исполните следующий алгоритм.

```
Начало
Опустить ручку.
Идти на 2 клетки вниз.
Идти на 2 клетки вправо.
Идти на 2 клетки вверх.
Идти на 2 клетки влево.
Поднять ручку.
Конец
```



Что у вас получилось?

Теперь составьте в рабочей тетради циклический алгоритм, который позволит вам получить такое изображение:



РАБОТА НА КОМПЬЮТЕРЕ

Познакомимся с созданием некоторых видов популярных игр в Лаборатории игр Коду. Один из таких видов игр — гонки.

Гонки могут быть по заданному маршруту. Есть также свободные гонки по бездорожью. Можно сочетать дороги и их отсутствие. Если гонки идут по указанному пути, то его надо прокладывать так, чтобы игрок не мог нарушить правила гонок и пройти короткой запрещённой дорогой. Помните про разновидности пути: дороги и стены, дороги с растениями. Их тоже можно использовать в гонках.

Для гонок лучше отвести побольше места. Если гонки проходят по дороге, то поверхность с холмами, растениями, озёрами, реками, мостами через реки — это просто красивое окружение дороги. Мост через реку — это просто дорога, у которой увеличили высоту узла, попавшего в реку. На узле щёлкают правой кнопкой мыши и выбирают «изменить высоту».



Дорога может не сразу принять правильный вид. Можно на миг запустить мир, затем вернуться в его редактирование. Некоторые персонажи могут проезжать и по дороге, которая проходит по дну реки.

В Лаборатории игр Коду вся поверхность может состоять из одной дороги. Попробуйте удалить всю поверхность, щёлкая правой кнопкой мыши при создании поверхности, а потом создать дорогу. Получится дорога, лежащая в небе.

На дороге можно устраивать разные препятствия. Можно проложить дорогу через земли, где падает скорость, можно установить мешающие предметы и выпустить мешающих персонажей. Мешающие предметы могут просто лежать на дороге, не давая проехать, пока их не столкнут. Особый мешающий предмет, действующий на расстоянии, — это вентилятор:



Впрочем, в зависимости от направления потока воздуха вентилятор может или мешать, или помогать. Воздух может дуть или от вентилятора, или к вентилятору. После размещения вентилятора важно правильно его повернуть. Для этого надо щёлкнуть на нём правой кнопкой мыши и выбрать «повернуть».

Вентилятор — это пример объекта с особенностями действий в круге выбора действий. У него есть действия, которых нет у других объектов: «втягивать» (pull) и «отталкивать» (push):





Мешающие персонажи в гонках могут гулять поперёк гоночной трассы. При столкновении с ними гонщик может выбывать из соревнований. Вряд ли в гонках допускаются персонажи, которые едут одновременно с гонщиками, толкают их и мешают соревнованиям. Впрочем, это ваша игра, можете в ней делать, что угодно.

Если вы расставляете препятствия и мешающих персонажей, то надо научить автоматических персонажей это всё преодолевать. И не забудьте указать все опасности и способы их преодоления в инструкции и в вызываемой по клавише F1 справке.

Там, где гонки должны завершиться, установите какой-нибудь предмет. Кто первым его коснётся, тот и будет победителем гонок.

Попробуйте создать свою игру — гонки с препятствиями.

подведём итоги

Вы научились создавать популярную игру «Гонки».

ВОПРОСЫ И ЗАДАНИЯ

- 1. Какие ещё игры вы хотели бы создать?
- 2. Что в создании игры в Лаборатории игр Коду вам удалось? Что не удалось?

§ 27 ИГРА ОТ ПЕРВОГО ЛИЦА



— Хотелось бы познакомиться с разными исполнителями.

Хорошо, Иван. Представим тебе исполнителя Чертёжник.

ЗАДАНИЕ 1

Команды исполнителя Чертёжник:

Идти вперёд ... шагов

Повернуть направо

Повернуть налево

Поднять ручку

Опустить ручку

Повторить ... раз

Вспомните: такие команды мы с вами уже записывали в параграфе 26.

Проанализируйте алгоритм для Чертёжника.

Начало Повернуть направо. Повторить 5 раз Опустить ручку. Повторить 4 раза Идти вперёд 3 шага. Повернуть налево. Поднять ручку. Идти вперёд 6 шагов. Конец

Что получится в результате выполнения алгоритма? Можете ли вы записать его в виде блок-схемы? Попробуйте сделать это в рабочей тетради.



РАБОТА НА КОМПЬЮТЕРЕ

Особенность игр от первого лица В том, что игрок видит мир глазами своего героя. Часто такие игры проходят в разлабиринтах, комнатах или личных когда игрок переходит из одного помещения В другое, где его ждут новые приключения. При ЭТОМ разные помещения позволяют спрятать игрока от взглядов всех мешающих ему персонажей. Игрок каждый раз сталкивается только с небольшой группой. Не забывайте при использовании условий «если вижу» добавлять карточку «на линии видимости» (LOS).

Для создания игры от первого лица надо построить лабиринт или помещения. Это удобно сделать с помощью создания стен (режим «Добавление и редактирование пути»). Созданный лабиринт надо наполнить препятствиями и мешающими персонажами. Надо дать возможность главному персонажу избегать их и поставить перед ним цель: куда-то дойти, что-то собрать. Конечно, игрок может встречаться со своими союзниками, беседовать с персонажами и получать от них задания.

Чтобы видеть мир глазами персонажа, надо научиться устанавливать камеру. У каждого персонажа на круге выбора действий можно выбрать действие «видеть»:



На появившемся круге можно выбрать один из трёх вариантов:

«1 игрок» — смотреть глазами объекта; «следить» — камера будет следить за этим объектом:

«игнор» — отменяет два других режима.



В разных ситуациях удобны разные режимы наблюдения.

Например, при создании игры в футбол хорошо смотреть за мячом, но смотреть на мир глазами мяча неинтересно: всё перед глазами мелькает и скачет.

Возможно, когда вы посмотрите на мир глазами персонажа, вы захотите чтото добавить в этот мир, чтобы он выглядел более привлекательно.

Переключать камеру можно автоматически, не давая выбора игроку, как видеть поле действий. А можно дополнить игру возможностью переключать камеру нажатием клавиш по усмотрению игрока. Такая возможность упрощает игру, но лишает её ощущения полного погружения в созданный вами мир. Возможно, переключение камеры клавишами будет полезным на этапе отладки программы.

Не забывайте подробно описывать правила игры в инструкции и в вызываемой по клавише F1 справке.

Попробуйте создать свою игру от первого лица.

подведём итоги

Вы познакомились с исполнителем Чертёжник.

Вы узнали, что такое игра, от первого лица в Лаборатории игр Коду.



вопросы и задания

- 1. Что может делать исполнитель Чертёжник?
- 2. Как вы думаете, какие возможности дают игры от первого лица?

§ 28 КВЕСТЫ

— А можем мы поработать с циклическими алгоритмами из окружающей нас жизни? Такими, где исполнители — мы?

ЗАДАНИЕ 1

Вспомните алгоритм перехода через дорогу, который вы составляли во втором классе.

Какие в нём были условия?

Можно ли записать такой алгоритм с помощью цикла?

Попробуйте составить в рабочей тетради алгоритм действий для перехода через дорогу по нерегулируемому перекрёстку. Для записи используйте язык блок-схем.

РАБОТА НА КОМПЬЮТЕРЕ

Квест — это игровое приключение, в котором нужно пройти разные препятствия для достижения какой-либо цели. В квестах много загадок и головоломок.





Для квеста в Лаборатории игр Коду характерно большое поле, возможно с сушей, морями и островами. Могут даже быть предложены разные территории. Поодной на другую нельзя С пасть иначе, при помощи телепорта (мгновенного как переноса). Если обычный телепорт переносит персонажа при соприкосновении с некоторым предметом, то телепорт в квесте сработает только при выполнении каких-то условий. При создании игры-квеста активно используются разные переменные, описывающие текущее состояние разгаданных загадок и выполненных заданий. Чтобы в них не запутаться, советуем выписать на бумаге, какая переменная что будет обозначать.

В квестах, создаваемых в Лаборатории игр Коду, нет возможности проверять ответы на загадки. Поэтому ответ на загадку должен использоваться в поиске очередного места на территории или в выполнении условий для того, чтобы сработал телепорт или выполнилось какое-то иное действие.

Для загадывания загадок и выдачи головоломок используйте действие «сказать».

Необязательно переносить с собой собираемые предметы, достаточно их «съесть» и увеличить счётчик их количества. Если вы хотите всё время видеть, сколько каких предметов собрано или сколько заданий выполнено, то можно использовать для этого цветные переменные, указав в установках создаваемого мира их названия.

Условием победы в квесте может быть касание чего-то или кого-то при выполнении условий по имеющимся к этому моменту предметам и по отгаданным загадкам.

Есть секрет, как сделать в Лаборатории игр Коду телепорт. Для этого надо выполнить следующие действия.

- Сделать основного персонажа порождаемым (свойство «родитель») с числом одновременно существующих объектов равным 1 (свойство «сколько можно создать»). В самом начале игры пусть любой объект породит одного основного игрока.
- При переносе на другую поверхность должно быть проверено условие телепортации. Затем персонаж должен быть уничтожен. После этого в переменной, которую вы выберете для телепортации, должен быть установлен номер территории, на которую вы

хотите перенестись. В Лаборатории игр Коду территории не нумеруются. Вы просто запомните, у какой территории какой номер. Равная нулю переменная телепортации означает, что телепортации нет.

- 3. На каждой территории должен быть какой-то неподвижный предмет для «ворота». В командах телепортации, этого предмета должна быть проверка значения переменной телепортации. Если значение этой переменной равно номеру территории, то «ворота» создают главного персонажа и тут же устанавливают переменную телепортации равной нулю.
- 4. Строго говоря, уничтожение ОДНОГО персонажа и создание такого же нового — это не телепортация, но в игре ЭТО очень похоже на телепортацию. Если какие-то свойства главного персонажа меняются в ходе игры, то, чтобы они не изменились при телепортации, надо сохранять их значение в переменных.

В долгой игре в квест игрок может забывать, что он уже сделал и что надо

делать на следующем шаге. Поэтому вызываемая нажатием F1 подсказка должна меняться по ходу игры. Можно выделить одну переменную, где хранить номер выдаваемой подсказки. А в команде выдачи подсказки выдавать текст в зависимости от номера подсказки.

Попробуйте создать свой квест. Самый простой квест может быть таким: на разнообразной территории разложите группу монет. Когда главный персонаж соберёт их и коснётся какого-то предмета, игра закончится.



ПОДВЕДЁМ ИТОГИ

Вы познакомились с созданием игр-квестов.

ВОПРОСЫ И ЗАДАНИЯ



 Что вам больше всего понравилось при создании квестов в Лаборатории игр Коду?

ПРОВЕРОЧНАЯ РАБОТА № 4



Создайте свою игру в Лаборатории игр Коду.

Выберите вид игры, придумайте интересную игру и создайте её. Обязательно сохраняйте всевозможные наброски и вспомогательные записи о будущей игре. Они помогут вам продолжить создание игры, если вы вернётесь к этому после перерыва, когда что-то уже забудете.

Успехов!

