

Научно-исследовательская работа

Информатика

**«Компьютерная игра. Разработка компьютерной игры
средствами конструктора Construct2»**

Выполнил:

Ломакин Матвей Дмитриевич

учащийся 4 класса

МБОУ СОШ пгт Вахрушев, Россия, пгт Вахрушев

Руководитель:

Гридина Ольга Михайловна

Учитель информатики,

МБОУ СОШ пгт Вахрушев, Россия, пгт Вахрушев

Содержание

Введение.....	4
1. Что такое программа.....	7
2. Виды компьютерных игр.....	7
3. Описание конструктора игр Construct 2	9
3.1. Основные части интерфейса программы	10
4. Описание созданной игры в Construct 2	14
5. Подготовка к анкетированию	21
6. Анкетирование и анализ.....	21
6.1. Исследование начального звена (2-4 классы).....	21
6.2. Исследование среднего звена (5-8 классы).....	22
6.3. Исследование старшего звена (9-11 классы)	23
Заключение	26
Список используемых источников.....	27
<i>приложение 1(подробное описание игры Labirint)</i>	<i>29</i>
<i>приложение 2 (разработанный проект игры Labirint)</i>	
<i>приложение 3 (игра Labirint (.exe))</i>	

Введение

Мы все любим играть, и в каждом возрасте свои игры и увлечения. Игра в определенном возрасте разносторонне развивает человека. Будучи малышами, нам интересны погремушки, кубики, машинки. И все это, конечно, занимательно, но как только человек знакомится с компьютером, тогда любые реальные игрушки уходят на задний план. Хорошо это или плохо, споры не утихают, но я не об этом пишу свою работу.

В три года, я не мог еще читать, но уже четко знал, как включить и выключить персональный компьютер. Клавиатура и мышь были чем-то особенным! При нажатии, клавиши издают особое «шуршание», которое мне безумно нравилось и в силу своего возраста я беспорядочно по ним любил «тыкать», что мне приносило огромное удовольствие.

Сейчас, мне уже 10 лет и обычные игрушки мне вовсе не интересны. Нет, вы не думайте, что я с головой погружен в виртуальную реальность, но становясь взрослее, компьютер все больше и больше вызывает у меня интерес. Интерес - не просто «погонять» с друзьями онлайн-игры, но и познать нечто новое.

Популярность и разнообразность компьютерных игр растет с каждым днем. Не только дети, но и взрослые проводят за этим занятием все больше времени. Бесконтрольное увлечение компьютерными играми может превратиться в игроманию, которая на сегодняшний день всемирной организацией

здравоохранения (ВОЗ) признана болезнью. В Китае же, игроманию признали профессией.

И я не являюсь исключением из огромного числа детей, которые достаточно много времени проводят за игрой в компьютерные игры и это факт, но передо мной встает задача как превратить это увлечение во что-то более продуктивное, чем простое развлечение.

Цель работы: исследовать возможности конструктора игр Construct2.

Задачи:

- выяснить какие компьютерные игры существуют;
- изучить среду визуального программирования Coundstruct2, ее возможности и характеристики;
- создать собственную развивающую игру;
- провести анализ полученных результатов с целью выявления учащихся интересующихся программированием.

Объект исследования: среда программирования Coundstruct2.

Предмет исследования: принципы создания компьютерных игр в среде Coundstruct2.

Гипотеза: можно научиться создавать компьютерные игры с помощью программы Coundstruct2 не имея даже базовых навыков программирования, достаточно проявить интерес.

Новизна: мало кто из младших школьников может создавать свои собственные игры.

Актуальность темы заключается в том, что многие дети и взрослые любят играть в компьютерные игры, которые в последствии могут послужить толчком к изучению программирования.

Методы исследования: поисковый, сравнение, сопоставление, анализ.

Ожидаемый результат:

- создание собственной компьютерной игры в Coundstruct2;

- учащиеся умеют создавать свои компьютерные игры.

1. Что такое программа

Для того, чтобы понимать, что такое компьютерная игра, считаю необходимым разобраться, что значит слово "программа".

Каждый из нас ежедневно выполняет по порядку определенные действия, например, проснувшись, встаем, идем умываться, завтракаем, идем в школу или уборка в доме, уход за домашним животным и многое другое. Запланированный порядок действий, к которому человек привыкает, можно тоже назвать программой. В зависимости от изменения обстоятельств, человек может "перестроить свою программу", так как обладает разумом.

Компьютерная игра - это сложная компьютерная программа, созданная человеком, с ограниченным количеством задач, имеющая искусственный интеллект, служащая для организации игрового процесса, связи с партнерами по игре или сама выступающая в качестве партнера. А значит, программа - это предписание, в которой заложена заданная четкая последовательность определенных действий, выполнение которых ведёт к достижению результата.

Вместо слов компьютерная игра, можно использовать слово видеоигра, то есть данные слова могут употребляться как синонимы. В компьютерных играх, как правило, игровая ситуация воспроизводится на экране дисплея или обычного телевизора.

2. Виды компьютерных игр

В настоящее время программистами разработано и разрабатывается огромное количество компьютерных игр не только на персональный компьютер, но и мобильные устройства, приставки (Nintendo switch, Sony Play Station, Xbox и др.) и все они рассчитаны на разные возрастные группы. Одни выступают как обучающие, другие для развлечения. Игры могут создаваться на основе фильмов и книг.

Компьютерные игры классифицируются по следующим тематикам:

- Приключенческие. Эти компьютерные игры оформлены как мультфильм с интерактивными функциями – возможностью управлять главным персонажем. Такие игры тренируют сообразительность и логическое мышление. Главный герой на своём пути встречает разные предметы, которые помогают ему усилить свои способности или наоборот, могут нанести вред. При этом персонаж попадает в разные приключенческие ситуации.

- Стратегии. Возможность управлять войском, энергией или иными подобными предметами. Здесь необходимо уметь планировать, рассчитывать свои силы и действовать в зависимости от ситуации. Это достаточно сложные игры, но именно они формируют усидчивость и умение планировать свои действия заранее.

- Аркады. Это многоуровневые игрушки, перейти к следующему этапу можно только выполнив конкретные задания и получив специальную награду. В таких играх участник набирает баллы, которые получает за прохождение конкретной миссии. Такая игра хорошо развивает глазомер и внимание.

Компьютерные игры делятся на несколько типов: квесты, экшен, ролевые игры (рпг), стратегии, симуляторы, логические и азартные. У каждой игры свой жанр. Существуют игры, объединяющие несколько жанров.

Меня же больше привлекают квесты, экшен/шутер.

Например, играя в квесты, мы, в лице главного героя, отправляемся в большое путешествие, преодолевая разные трудности, часто, решаем непростые задачи, тренируем сообразительность и мышление, учимся находить нестандартные решения. Одной из таких игр, является *SpongeBob SquarePants: The Movie*. Красочный и необычный платформер-квест. Сюжет построен на базе известного мультика и заключается в расследовании похищения. Играя в эту игру, мы погружаемся в атмосферу событий любимого мультика.

Brawl Stars (онлайн-игра) захватывающая игра. Она сочетает в себе несколько жанров – стратегии, экшен и др. В игре главное - это физические

возможности игрока, его координации глаз и рук и скорости реакции. Игра развивает пространственное мышление, память и реакцию. Цель игры – сбор кристаллов и охрана своих сокровищ от противников в одиночку или в команде. Во время тестовых заданий новичок учится управлять своим персонажем, остерегаться врагов и добывать драгоценные кристаллы. Задача игрока – «добивайся всего сам!»

Конечно, играя в компьютерные игры мы не только развлекаемся, но тренируем память, учимся логически мыслить.

Начиная с детского сада и начальной школы, через компьютерные игры мы можем научиться складывать слоги и читать, из одного слова составлять другие слова и не только. Использование образовательных платформ таких как Яндекс Учебник, Учи.ру, где все задания представлены в игровой форме позволяют нам погружаться в ситуации, которые встречаются в жизни, и решать их.

Засиживаясь за игрой, родители часто нас ругают - "хватит сидеть за экраном, спина станет как крючок!", "глаза скоро на лоб вылезут!". Но в современной жизни нельзя закрыть дверь между ребенком и виртуальной реальностью, да и не надо. Просто надо соблюдать правила нахождения у экрана монитора. Через каждые 15 минут делать перерыв и специальные упражнения для глаз и спины, надевать защитные очки. А родителям стоит обращать внимание на то, в какие игры играет его ребенок, чтобы они соответствовали возрасту.

Не так давно у меня появился интерес не просто играть в компьютерные игры, а познать что-то новое. Спросите у меня, что? Так вот, я хочу попытаться создать свою компьютерную игру.

3. Описание конструктора игр Construct 2

Наверное, у настоящих любителей компьютерных игр иногда возникает вопрос как создать самому компьютерную игру. На самом деле, это очень сложно и над созданием игры трудится не один программист, а целая команда, знающие

языки программирования. Но для того, чтобы создать свою игру, не всегда обязательно знать какой-либо язык программирования. В настоящее время существуют различные игровые движки, которые позволяют создавать свои компьютерные игры. Благодаря сети Интернет я и познакомился с одним из таких движков - это Construct 2. Кстати, для создания компьютерных игр в данном приложении не требуется особых знаний по математике и английскому языку, все построено на логике.

Construct 2 - это конструктор двумерных игр для Windows разработанный компанией Scirra вторая версия программы Construct Classic, вышедшей в 2007 году, который по задумке создателей движка позволяет сделать процесс создания игры доступным с помощью визуального программирования на основе событий и связанных с ним действий. Это значит, что при создании игры, мы не напишем ни одной строчки программного кода.

Визуальное программирование — способ создания программы для ЭВМ путём манипулирования графическими объектами вместо написания её текста. Некоторые авторы представляют визуальное программирование как следующий этап развития языков программирования (следующее поколение). В настоящее время визуальному программированию стали уделять больше внимания, в том числе в связи с развитием мобильных сенсорных устройств (КПК, планшеты).

Construct 2 позволяет делать игры 2D различных жанров и сложности. Созданные игры на данном движке могут быть доступны на PC, Mac, Linux, в браузерах с поддержкой HTML 5, Android и IOS и др.

Программа имеет более 20 встроенных поведений, включая Platformer, 8 Direction, Bullet, Car, bullet и т.д., которые облегчают процесс создания игры.

3.1. Основные части интерфейса программы

Запускаем Construct 2. После запуска программы мы видим перед собой несколько вкладок:

- *File* (файл) в котором можно сохранить проект, создать новый или открыть уже имеющийся;
- *View* (вид) - можно показать сетку на игровом поле и перемещать по ней объекты;
- *Home* - можно запустить отдельный уровень;
- *Events* - можно редактировать события.

Для создания нового проекта нужно нажать файл и выбрать New empty project.

Теперь рассмотрим основные окна программы. В центре экрана мы видим белый лист (пространство где будет создаваться сама игра), простыми словами это и есть наш уровень (рис. 1).

Вкладка *Layout 1* (редактор уровня - всё что находится на уровне объекты, эффекты и т.д.), *Event sheet* (лист событий - задаются события игры, например при нажатии на кнопку персонаж идёт вперед).

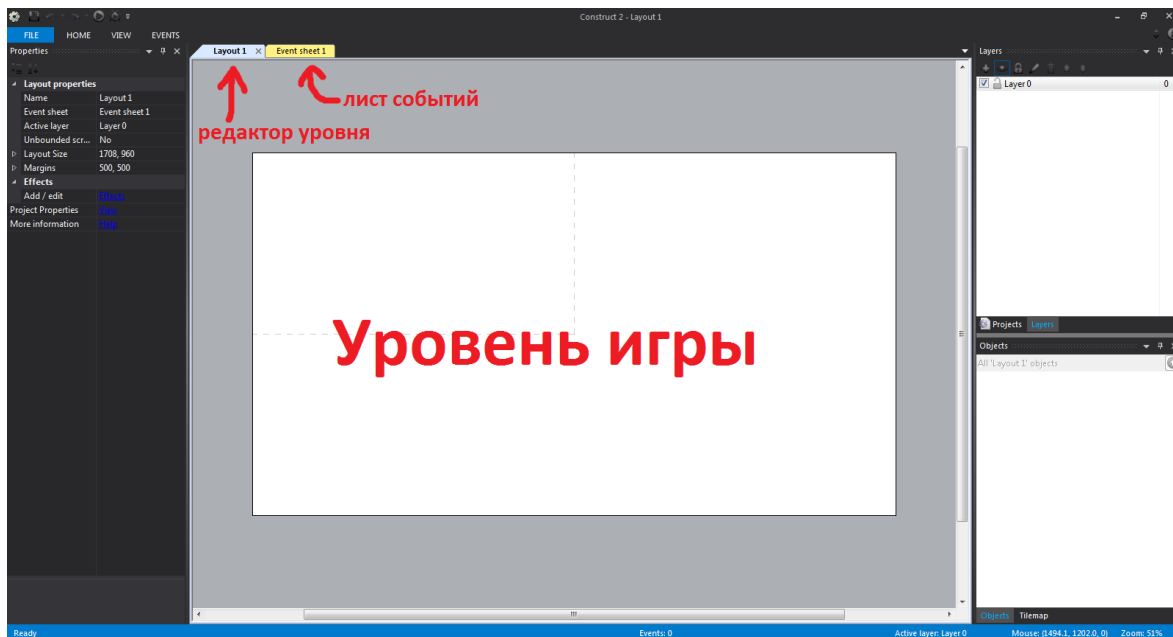


рис. 1

По бокам уровня расположены другие окна.

Начнём с окна *Project* (рис 2). В ней отображается открытый в данный момент проект в который входят - уровни игры, листы событий, звуки и т.д.

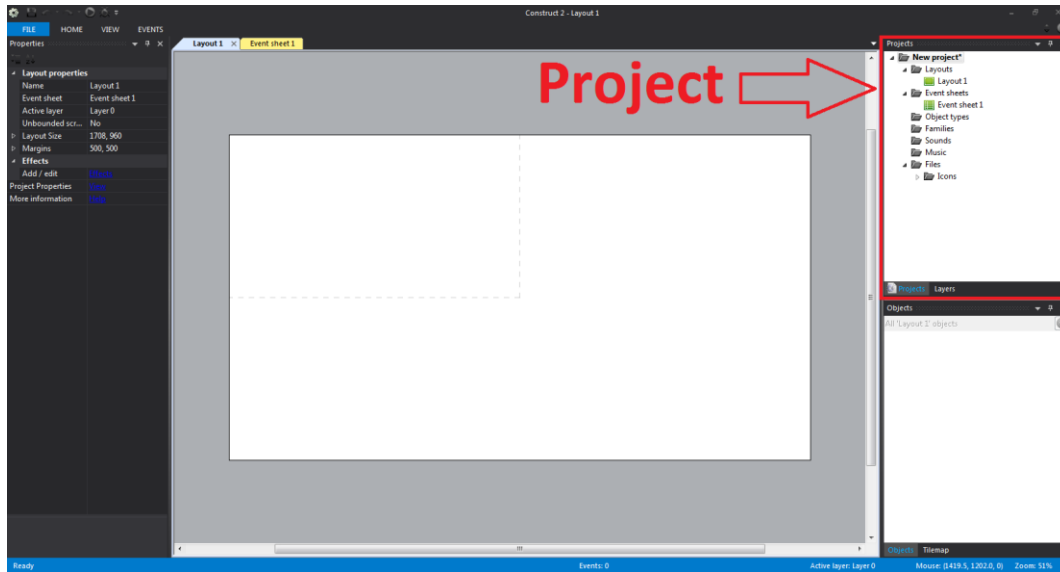


рис. 2

Вкладка *Layers* (рис. 3) отвечает за отображение всех доступных на данном уровне слоёв. То есть для каждого уровня набор слоёв может быть свой.

Что такое слой - слой это некое пространство с помощью которого можно нужным образом группировать объекты, т.е. разместив объекты на разных слоях, мы можем прятать одни объекты под другими, а так же заблокировать слой с объектами, что бы было невозможно с ними взаимодействовать.

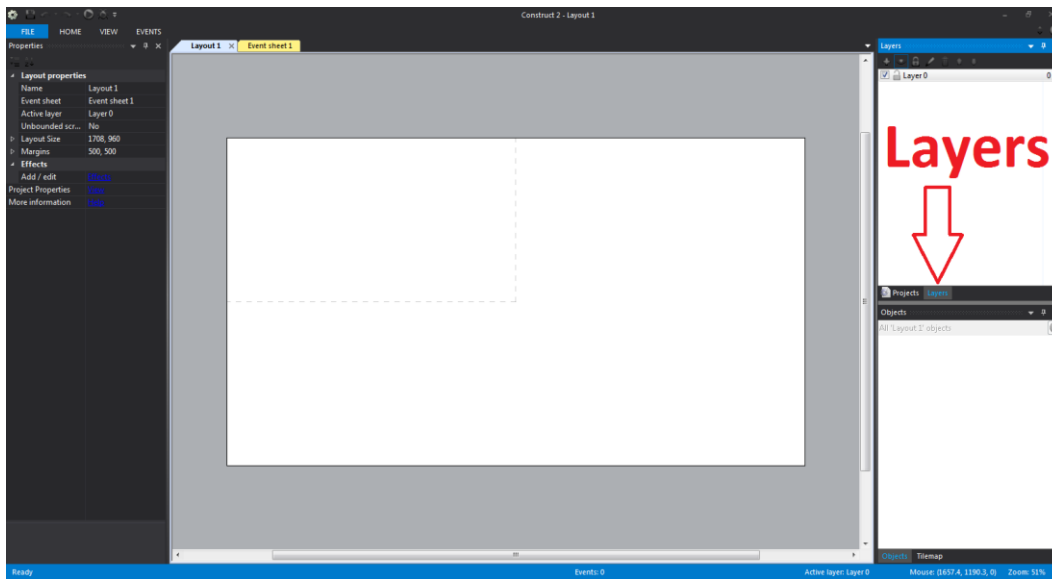


рис. 3

Окно Object (рис. 4). В ней находятся все объекты принадлежащие тому уровню на котором мы сейчас находимся.

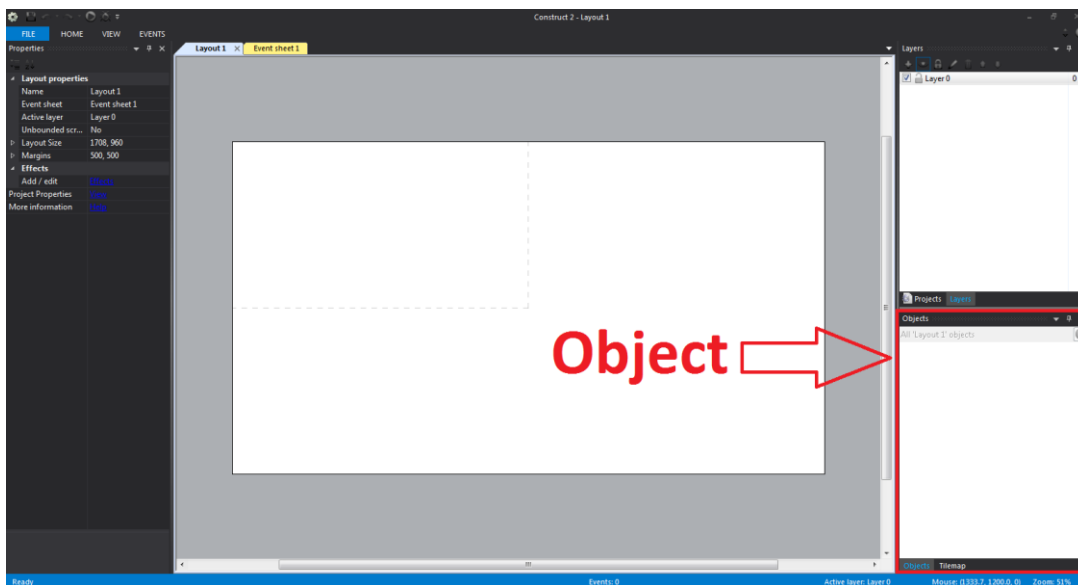


рис. 4

Вкладка *Properties* (рис. 5) отвечает за настройку всего проекта и некоторых элементов на нём. Например - названия спрайтов, название уровней, разрешение экрана всего проекта, поведения объектов и тд.

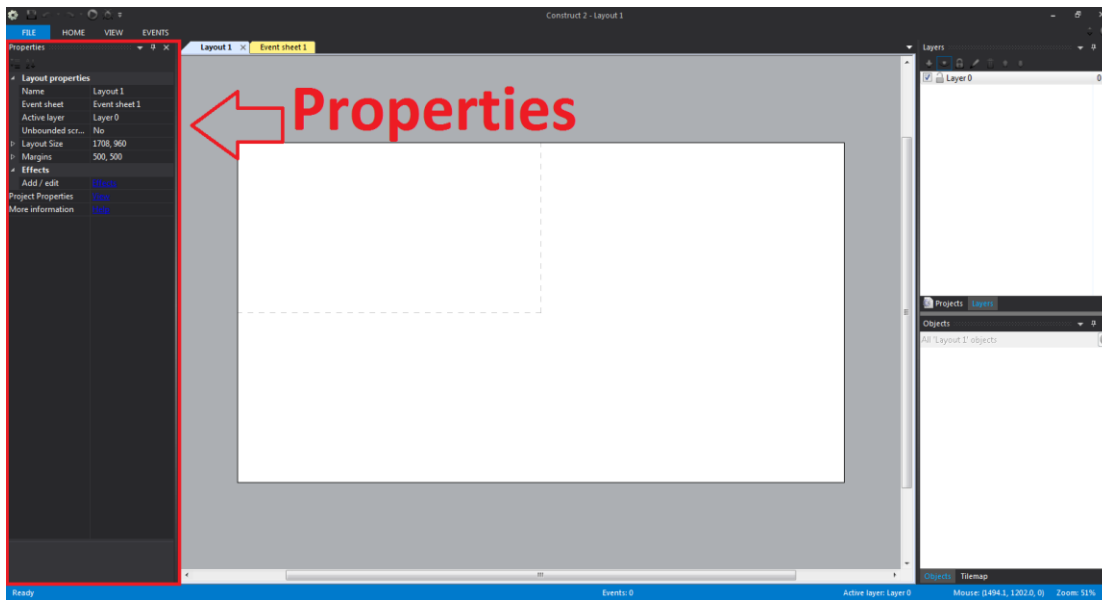


рис. 5

Сейчас мы коротко рассмотрели основные части интерфейса программы для создания компьютерных игр в Construct 2. В силу своего возраста и опыта, конечно, я еще очень мало знаю о ее возможностях, но все же попробую.

4. Описание созданной игры в Construct 2

Приступаю к созданию собственной игры.

Название: Labirint.

Жанр игры: Action.

Управление: стрелочки на клавиатуре - ходьба; пробел - поставить бомбу.

Сюжет: Вы будете играть в роли одинокого странника, искателя сокровищ и приключений по прозвищу "Бомбер".

Ходила легенда о том, что когда-то давно в старом одиноком и заброшенном замке жил один очень богатый и жадный король. Он был настолько жаден, что забирал все драгоценности у собственного народа. И дабы их не украли у него, он спрятал их глубоко в подвале своего замка в виде лабиринта, с кучей препятствий и охраной их монстрами. И вот, уже на протяжении многих веков различные

искатели сокровищ пытались их найти, но все они потерпели неудачу, поскольку препятствия оказались слишком сложными, а монстры слишком страшными и сильными. Но не смотря ни на, что, главный герой хочет отыскать украденные сокровища и раздать их бедным людям.

Игра создана с помощью программы Construct 2 в пиксельном ретро стиле и представляет собой 2D Top-Down action (вид сверху). Это приложение позволяет создавать игры без знания программирования, поскольку использует визуальное программирование.

Визуальное программирование - простыми словами, вы просто выбираете какое-то событие и выполняете определённое действие с ним, например: выбираю клавишу вперёд и выбираю действие, чтобы персонаж двигался вперёд и делал анимацию ходьбы.

В своей игре я использовал как скачанные из интернета спрайты (картинки персонажей и объектов) из реальной игры "Super Bomberman 5" которая выходила на приставку Snes (Super Nintendo Entertainment System) так и рисовал собственные спрайты (картинки: ключ, дверь, пуля, пол, пушка, облачко) в Paint для создания игры. Но сюжет я придумал свой.

Все объекты игры для удобства расположены на четырех слоях, создано семь групп с событиями, видны используемые спрайты и т.д. (рис.6, рис. 7).

имеет музыкальное сопровождение, которое погружает игрока с головой в игровой процесс.

Первый уровень выполнен очень просто и на нём вы научитесь базовым действиям, таким как ходьба и взаимодействие с ключом и дверью (рис. 8). С каждым уровнем игра становится сложнее.

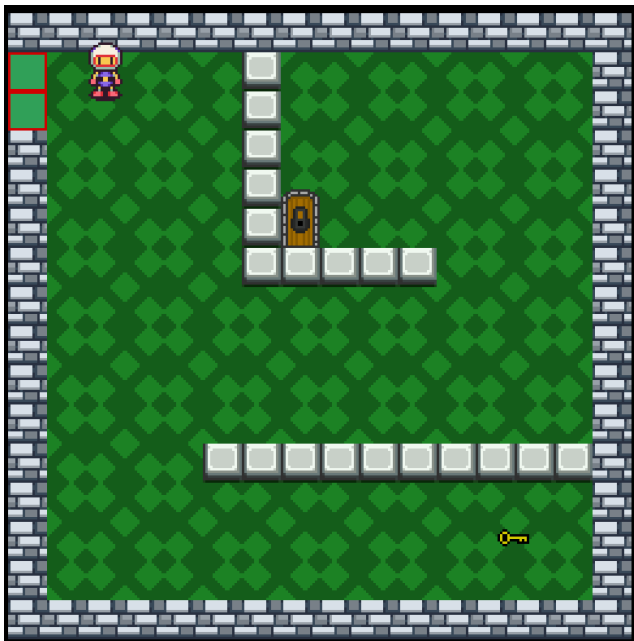
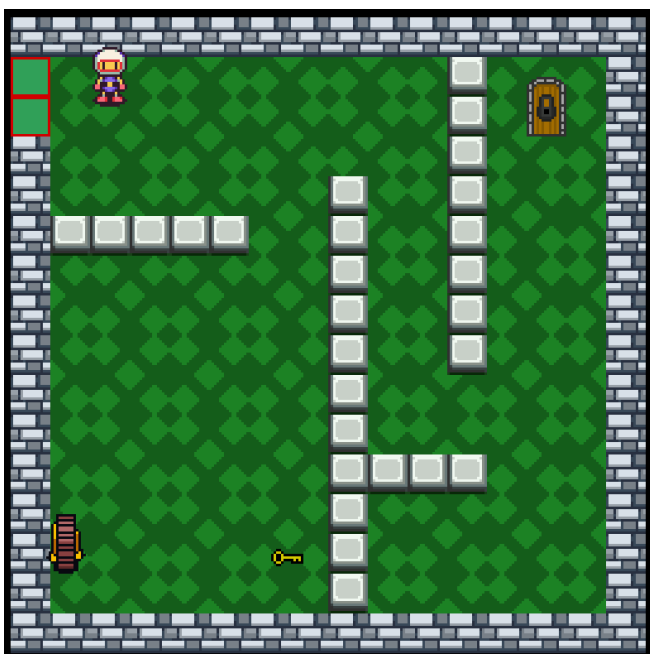


рис. 8

На втором уровне вы познакомитесь со своим первым врагом и поймёте что можете погибнуть от него (рис. 9). Поэтому лучше его избегать.



(рис. 9)

На третьем уровне вы обнаружите такой бонус как бомба (рис. 10). Это серьёзное оружие которое может как уничтожить врага, так и уничтожить вас самих.

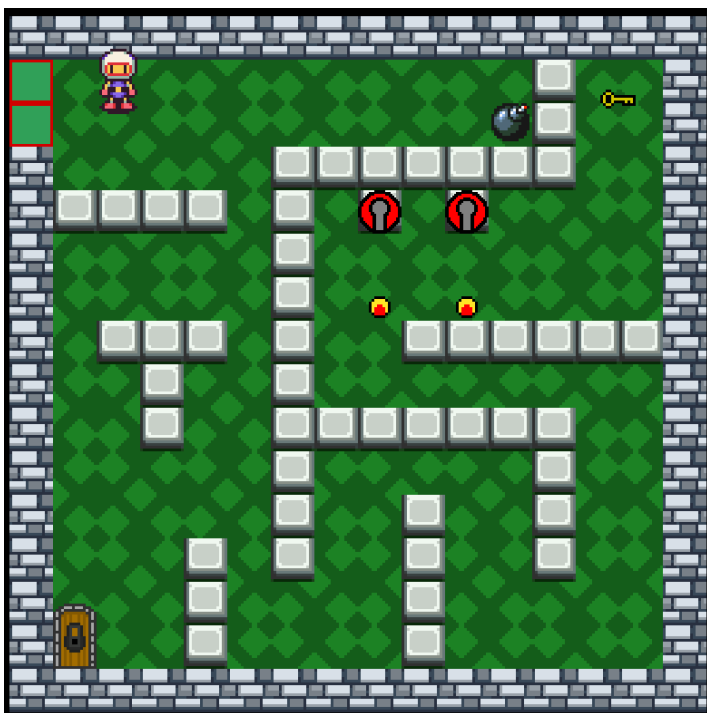


рис. 10

На четвёртом уровне вы обнаружите что не все враги погибают от бомбы (рис. 11).

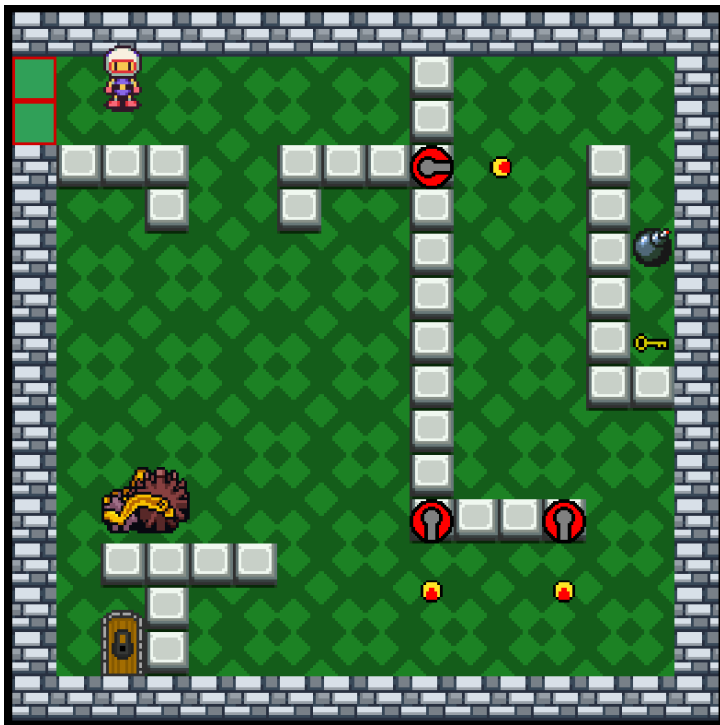


рис. 11

На пятом уровне вы должны будете пройти спиральную комнату с огромным количеством пушек (рис. 12).

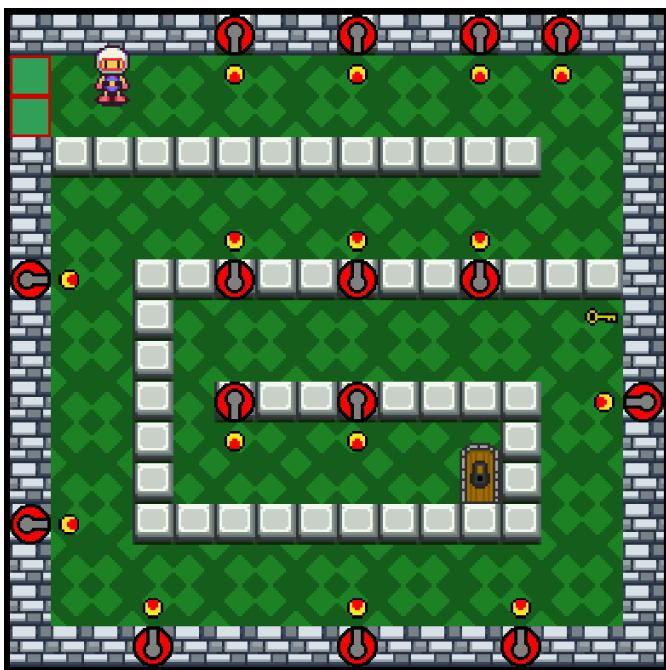


рис. 12

Ну, и наконец на шестом уровне, за все те препятствия, которые вы преодолели, получите своё вознаграждение в виде сундука с сокровищами и поздравительной надписью!

Удачи в прохождении!

5. Подготовка к анкетированию

Считаю, что тема компьютерных игр, программирования на сегодняшний день актуальна. Мне стало интересно и захотелось узнать в какие компьютерные игры играют ученики нашей школы, интересуется ли их программирование и возможно кто-то еще учится создавать свои компьютерные игры.

Для проведения исследования, для всех учащихся была подготовлена анкета с одинаковыми вопросами:

1. Вы играете в компьютерные игры? (да/нет)
2. В какие компьютерные игры предпочитаете играть? (да/нет)
3. Вас интересует программирование? (да/нет)
4. Вы пытались создать свою компьютерную игру? (если "да", то укажите приложение)

6. Анкетирование и анализ

В исследовании приняли учащиеся в количестве - 91 человека, из них:

- ученики начального звена (2-4 классы) - 30 человек;
- ученики среднего звена (5-8 классы) - 42 человека;
- ученики старшего звена (9-11 классы) - 19 человек.

6.1. Исследование начального звена (2-4 классы)

Ответ на первый вопрос анкетирования показал, что все ребята начальной школы, за исключением одного учащегося играют в компьютерные игры (рис. 13).

На второй вопрос - учащиеся разделились.

На диаграмме мы видим, что из 30 ребят, основная масса (24 чел.) играет в многопользовательские онлайн - игры с различными видами жанров - это *Minecraft*, *Roblox*, *Brawl Stars*, *Standoff 2*.

Гораздо меньшее количество (5 чел.) играют в однопользовательские компьютерные игры, такие как - *Minecraft* (песочница, сочетающая различные виды жанров), *Slime Rancher* (симулятор жизни/приключенческий боевик), *Red*

Ball (платформер), *Фиксики Мастера* и *Веселая математика* (обучающие квесты).

Как показывает вторая диаграмма (рис.14) - ответ учащихся на третий и четвертый вопросы показал, что не смотря на свой маленький возраст, ребята интересуются, что такое программирование, а один создает свою компьютерную игру в *Construct2*.



рис.13

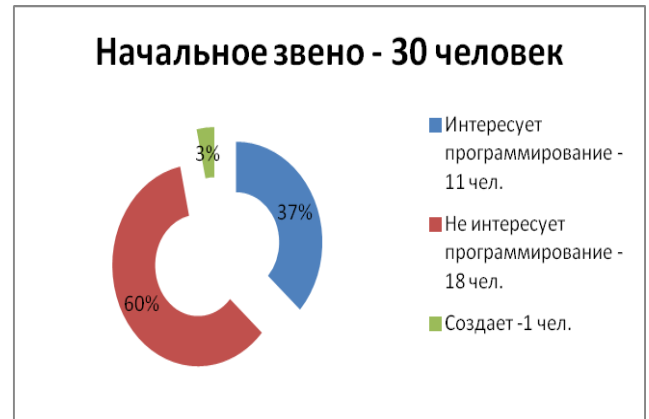


рис.14

6.2. Исследование среднего звена (5-8 классы)

Ответ на первый вопрос анкетирования показал, что ребята средней школы так же играют в компьютерные игры, но количество не играющих увеличилось до 5 человек (рис. 15).

На второй вопрос, как и учащиеся начального звена - разделились.

На диаграмме мы видим, что из 42 ребят, основная масса (34 чел.) играет в многопользовательские онлайн - игры с различными видами жанров, при этом разнообразие игр увеличивается, основными из которых являются *Minecraft*, *Roblox*, *Brawl Stars*, *Standoff 2*, *Pubg mobile*, *GTA5*, *Warface*, *World of Tanks*, *Couner Strike*, *Clash Royale*, *Far Cry*, *Star Wars Battlefront*, *Fortnite* и др., и всего 3 человека играют в однопользовательские компьютерные игры, такие как - *Как достать соседа* (аркада), *Stalker2* (шутер от первого лица), *Subway Surfers* (платформер), *People Playground* (симулятор) и др.

Следующая диаграмма (рис.16) - ответ учащихся на третий и четвертый вопросы показал, что ребята так же интересуются программированием (17 чел.), при этом, один учащийся создает 2D анимации в *Aseprite*¹. Еще один создает 3D анимации в среде *Roblox Studio*², скрипт пишется на языке программирования Lua, а это уже текстовое программирование.

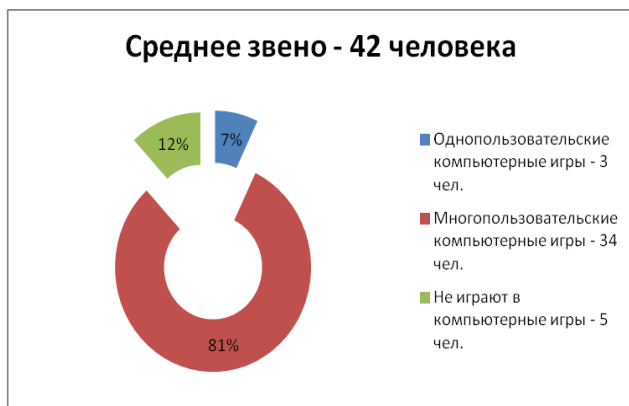


рис. 15

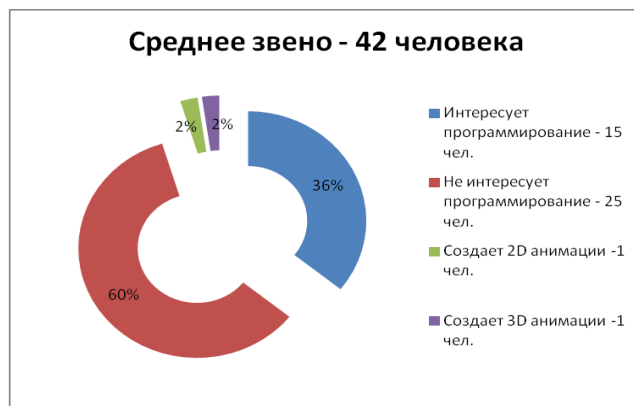


рис. 16

6.3. Исследование старшего звена (9-11 классы)

Анкетирование показало, что ребята старшей школы играют в компьютерные игры (рис. 17), но к моему удивлению, количество играющих уменьшилось практически на половину, интересующихся программированием в начальном и среднем звеньях было на уровне, но в старшем звене снизилось, а занимающихся текстовым программированием увеличилось до трех человек (рис.

¹ Благодаря **Aseprite** пользователь имеет возможность создавать анимацию в формате 2D, статичные двухмерные картинки, любые типы графики и спрайты. Программа представляет собой инструмент для создания графики пиксельного типа. Благодаря ей пользователи имеют возможность проводить эксперименты с тем, что они создали. Имеется возможность выбирать самые разные логотипы, текстуру, заставки. Можно изменять фон, узоры, движения. При необходимости возможно комбинирование большого количества цветовых палитр. Данное программное обеспечение подходит как для профессионалов, которые разрабатывают элементы для видеоигр, так и для тех пользователей, кто просто любит пиксельные изображения.

² **Roblox** — это популярная онлайн-платформа, где пользователи могут создавать свои 3D-игры и размещать их.

18) из которых - двое работают на языке *Python*³, создавая свои компьютерные игры, третий работает в *Java*⁴ - занимается написанием веб-программ.

На диаграмме мы видим, что из 19 ребят, а это девять человек, играют в многопользовательские онлайн - игры с различными видами жанров. Примечательным оказался тот факт, если в начальном звене ребята указали всего одну игру в которую играют, то в среднем звене количество многопользовательских игр на одного ребенка доходило четырех, а в старшем не более двух и только один играет в шесть. Основными из которых являются *Minecraft, Brawl Stars, Standoff 2, Pubg mobile, GTA5, Warface, Rust, Valorant, Dota u Couner Strike*. Один человек играет в однопользовательскую компьютерную игру *Subway Surfers (платформер)*.



рис. 17



рис. 18

³ **Python** - популярный язык программирования высокого уровня, который предназначен для создания приложений различных типов: веб-приложения, игры, настольные программы.

⁴ **Java** — объектно-ориентированный язык программирования, используется для написания веб-программ, приложений и серверного программного обеспечения.

Заключение

Заканчивая писать свою работу, считаю, что моя гипотеза подтвердилась, поставленные задачи выполнены. Компьютерные игры действительно можно научиться создавать в Counstruct 2 не имея базовых навыков программирования и созданная мною игра является тому доказательством.

Казалось бы, на первый взгляд все очень сложно и запутано, но достаточно иметь желание, проявлять интерес, усидчивость и конечно же первым на помощь приходит Интернет. Он открывает перед нами безграничные возможности - от огромного множества разнообразных видео уроков в YouTube, до форума на котором всегда можно задать вопрос и тебе с радостью ответят, помогут, подскажут.

С уверенностью могу сказать, что для меня, компьютерные игры послужили толчком для начала изучения программирования. Возможно в будущем, визуальное программирование перерастет во, что-то более серьезное и я начну изучать текстовое программирование.

Удивил и тот момент, что при проведении анкетирования и его анализа, достаточное количество ребят интересуются программированием, некоторые учатся создавать 2D и 3D анимации, а кто-то уже создает свои компьютерные игры. Думаю, что и для других ребят игра в компьютерные игры также как и для меня послужили толчком в изучении программирования.

Очень хочется пожелать тем ребятам, которые засиживаются за продолжительной игрой в компьютерные игры и зачастую не соответствующие возрасту, направить и попробовать свои силы в более увлекательном занятии, как изучение программирования. Пускай сегодня это будет создание 2D и 3D анимации, визуальное программирование, а завтра гораздо большее.

Программирование - это настоящее и будущее.

В заключении хочется сказать, что Construct 2 прекратил свою официальную поддержку от разработчика 1 июля 2021г., поскольку вышла новая версия - Construct 3 (поддерживает разные языки, в том числе и русский). Но не смотря на это, считаю, что приложение прошлой версии всё ещё остаётся актуальным, поскольку проекты созданные в Construct 2 можно открывать и продолжать разрабатывать в Construct 3, чтобы игру выпустить официально.

Список используемых источников

<https://www.youtube.com/c/BarmaleyPlay>

<http://c2community.ru/forum/>

<https://opengameart.org>

<https://www.sprisers-resource.com/snes/sbomber5/>

http://school81.tgl.ru/sp/pic/File/2018-2019/dekabr/Prilojenie_Opisanie_servisa.pdf

<https://www.alawar.ru/logicheskie/detskie/>

<https://infourok.ru/referat-kompyuternye-igrы-vidy-plyusy-minusy-rekomendacii-roditelyam-po-vyboru-i-organizacii-4953552.html>

<https://wobla.ru/infomat/games.aspx>

<https://www.i-igrushki.ru/igrushkapedia/kompyuternye-igrы.html>

0 https://www.igromania.ru/news/83114/VOZ_oficialno_priznala_igrovuyu_zavisimost_boleznyu.html

https://ru.wikipedia.org/wiki/Визуальное_программирование

1

<https://bytorrent.games/23663-aseprite.html>

2

<https://zen.yandex.ru/media/id/5f7b5efe4c2a9a1f2b8a0085/sozdanie-igr-v->

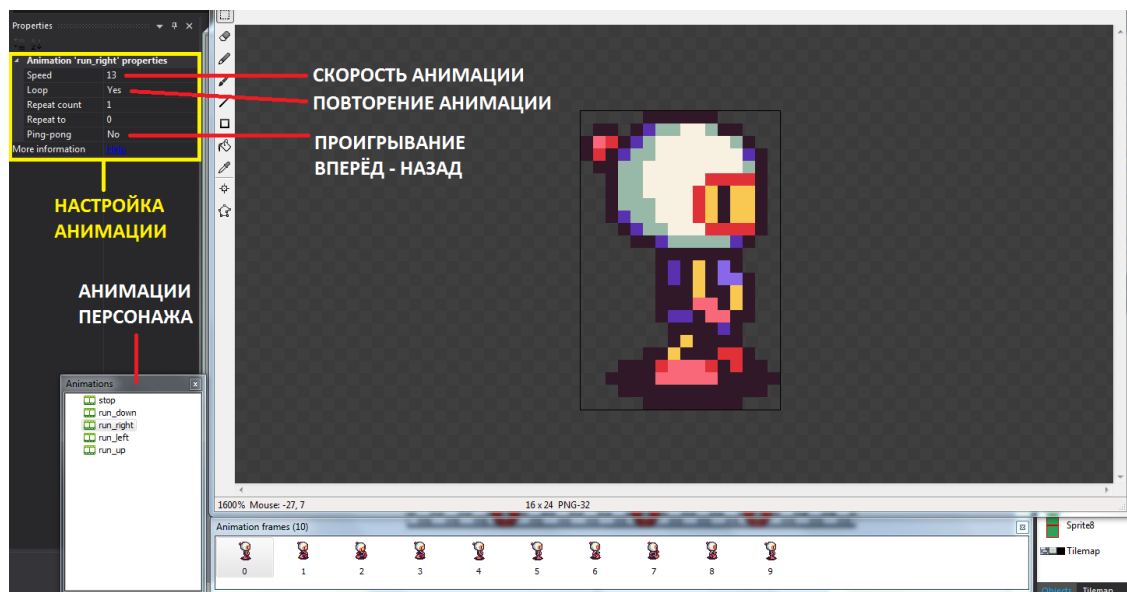
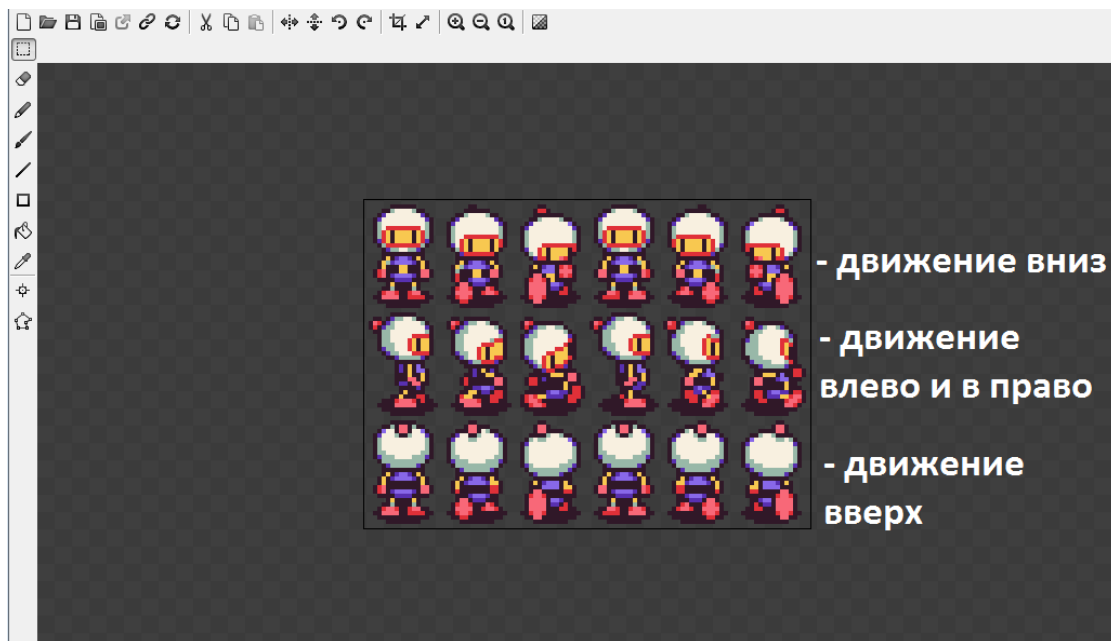
3 roblox-studio-dlia-detei-ot-9-let-besplatnye-videouroki-
604b17f76257184ecb99ba1a

Подробное описание созданной игры Labirint

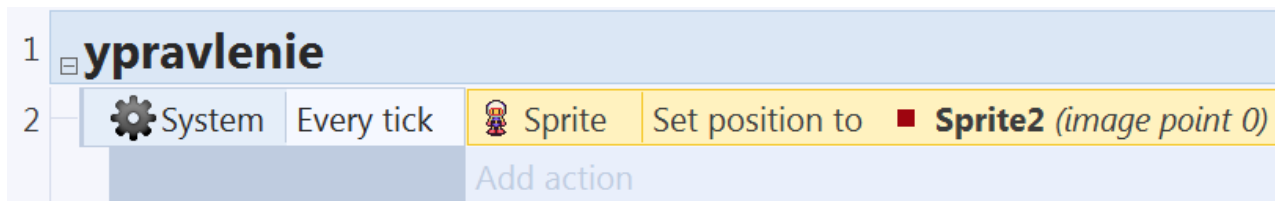
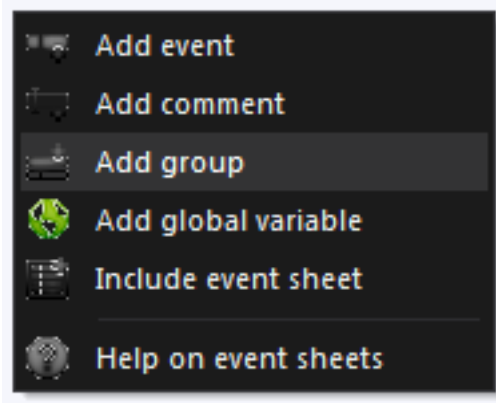
Запускаем Construct 2, создаем новый проект File/New empty project, задаем имя нашего проекта (название будущей игры) - Labirint.

Начнем создание главного героя.

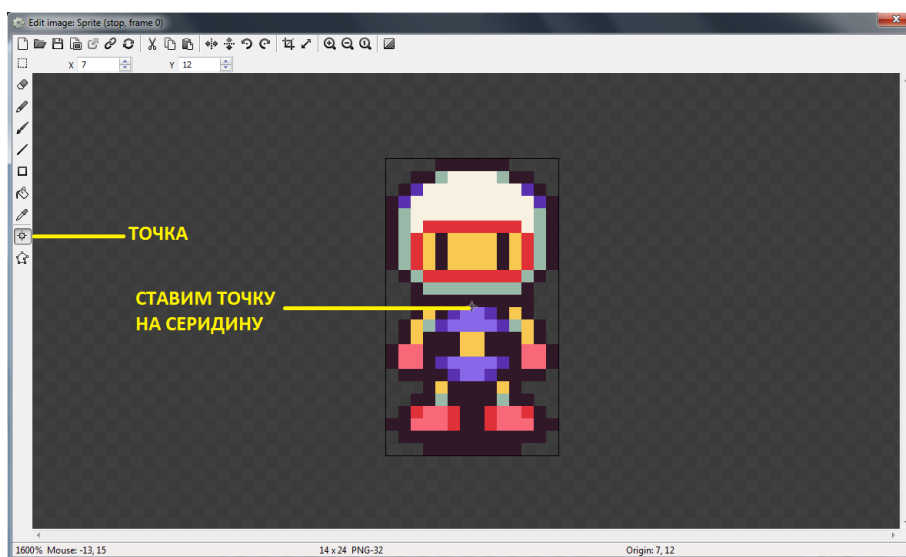
Для его создания я скачал готовые спрайты Бомбера из Интернета, открыл скачанные спрайты в конструкторе, вырезал нужные спрайты в каждый кадр анимации, настроил им нужную скорость в окне настройки анимации.

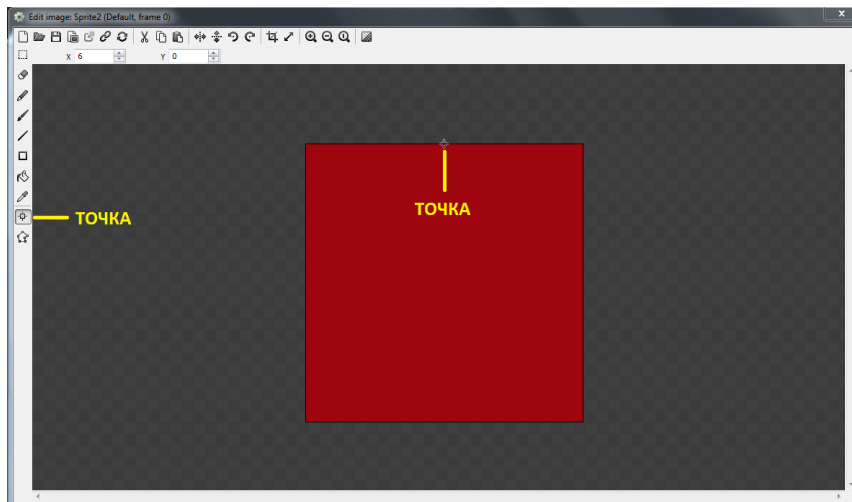


Создаем кубик, который будет выступать в роли главного персонажа (хит-бокс). Это необходимо для того, чтобы наш персонаж пикселями не цеплялся за твердые объекты, в нашем случае это стены (у кубика все стороны равные). Далее в листе событий, создаем событие - всегда спрайт "бомбера" стоит на позиции объекта "кубик". Далее создаем папку "управление" и перетаскиваем созданное событие в созданную папку.



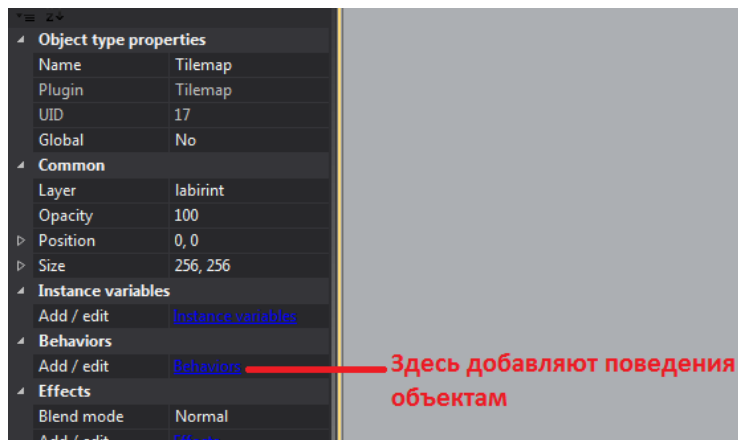
Теперь настроим точки у кубика и "Бомбера" за которые они будут прикрепляться друг к другу.



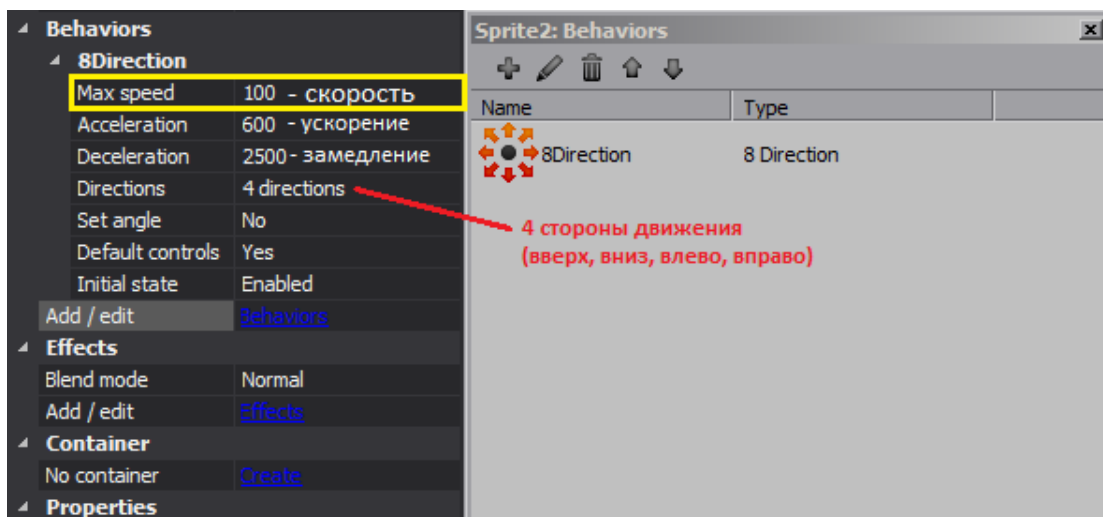


Задаём поведение кубику - 8direction.

Это поведение отвечает за передвижение персонажа с видом сверху.

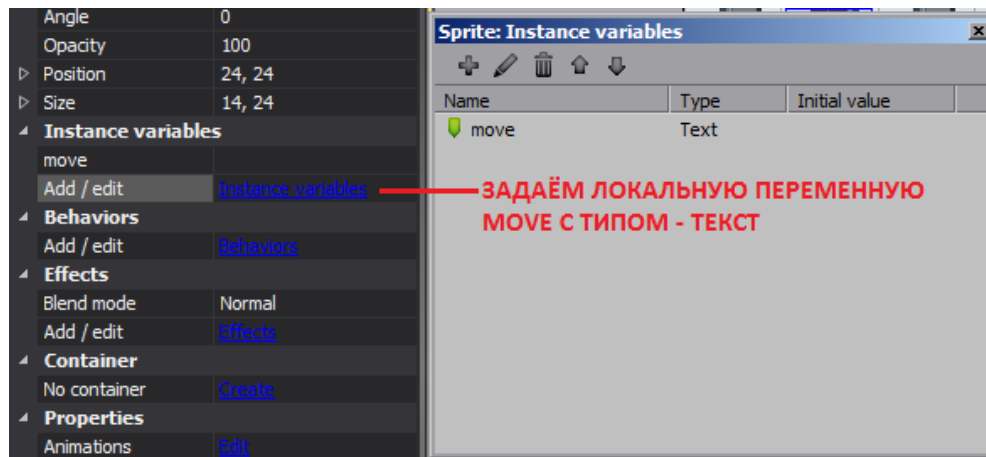


Настраиваем скорость передвижения 100 и замедление 2500. (чем выше тем быстрее останавливается).



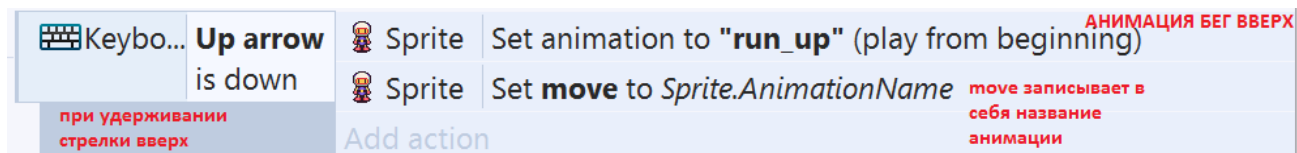
Задаю "Бомберу" локальную переменную с названием *move* с типом - *текст*.

Она будет отвечать за правильную сторону анимации при остановке персонажа.

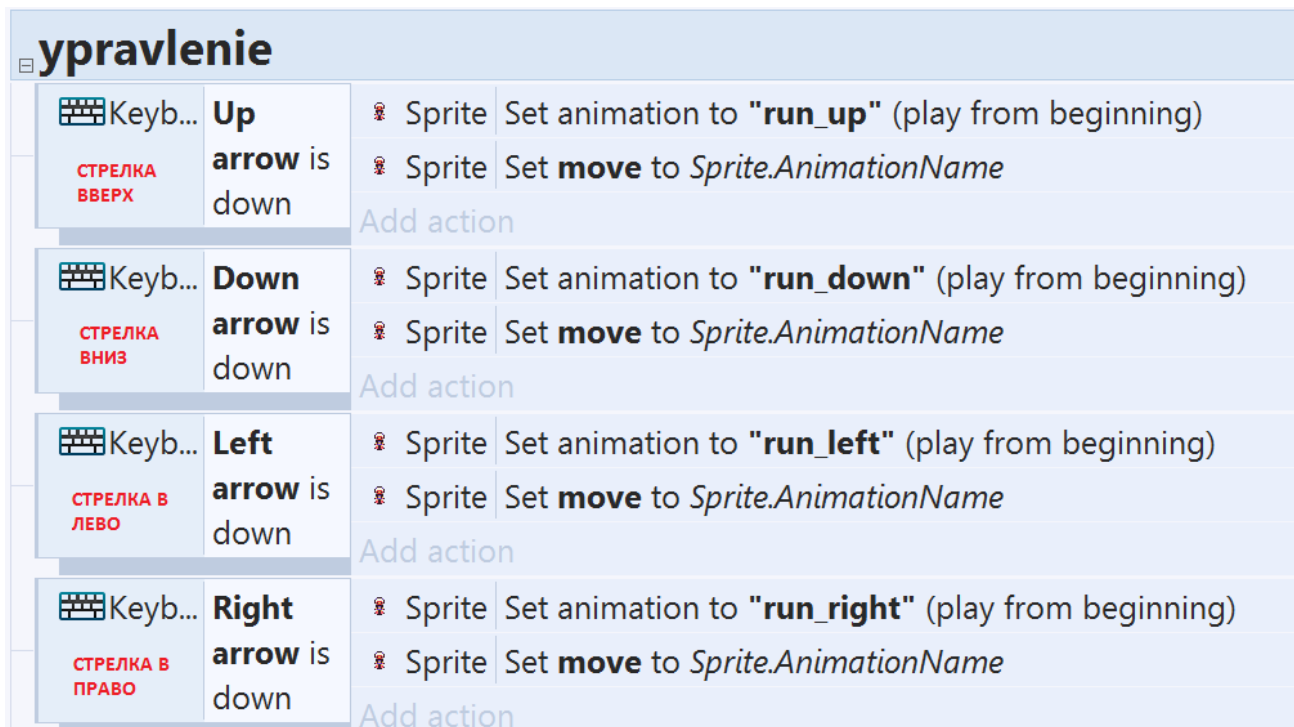


Создаю событие в листе событий.

Если удерживать клавишу стрелка вверх, то "Бомбер" делает анимацию - бег вверх и локальная переменная бомбера *move* записывает в себя название проигрываемой анимации в данный момент.



И так создаю события для всех клавиш со стрелками.



Теперь создам событие для правильной анимации остановки в нужном направлении.

Keybo... <small>если отпустил любую клавишу</small>	On any key released	Sprite Set animation to "stop" (play from beginning)	Add action <small>БОМБЕР ДЕЛАЕТ АНИМАЦИЮ СТОП</small>
S...	move = "run_up"	Sprite Set animation frame to 3	И ТАК КАК В MOVE ЗАПИСАНА АНИМАЦИЯ, ТО БОМБЕР ВЫБИРАЕТ НУЖНЫЙ КАДР АНИМАЦИИ "STOP" ИСХОДЯ ИЗ НАЗВАНИЯ ПРОИГРЫВАЕМОЙ АНИМАЦИИ В ДАННЫЙ МОМЕНТ
S...	move = "run_left"	Sprite Set animation frame to 1	
S...	move = "run_right"	Sprite Set animation frame to 2	
S...	move = "run_down"	Sprite Set animation frame to 0	

Теперь создам событие для того, чтобы персонаж смог остановиться если мы нажимаем две кнопки одновременно.

Keybo... <small>если нажата стрелка в верх</small>	Up arrow is down	Sprite2 Start ignoring 8Direction user input
Keybo... <small>если нажата стрелка в лево</small>	Left arrow is down	

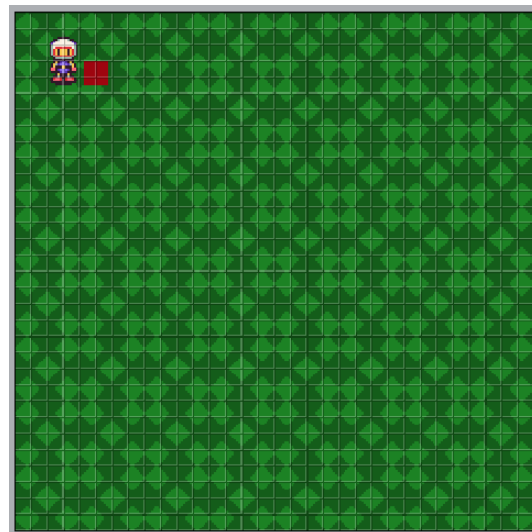
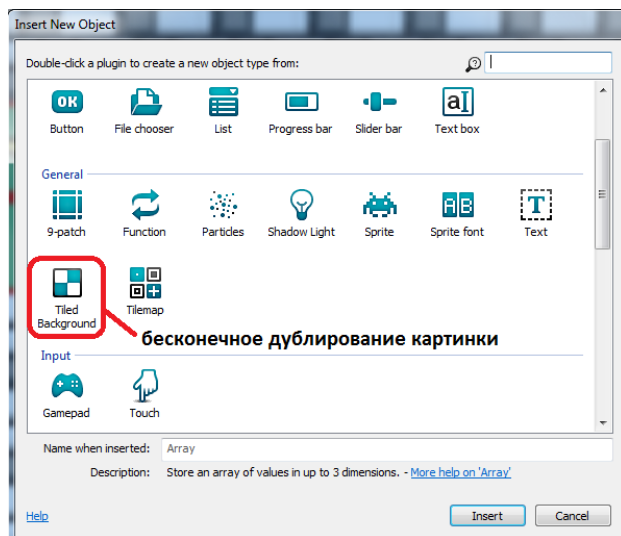
включается игнорирование управления кубика (игрока)

И так делаю для всех 4 направлений.

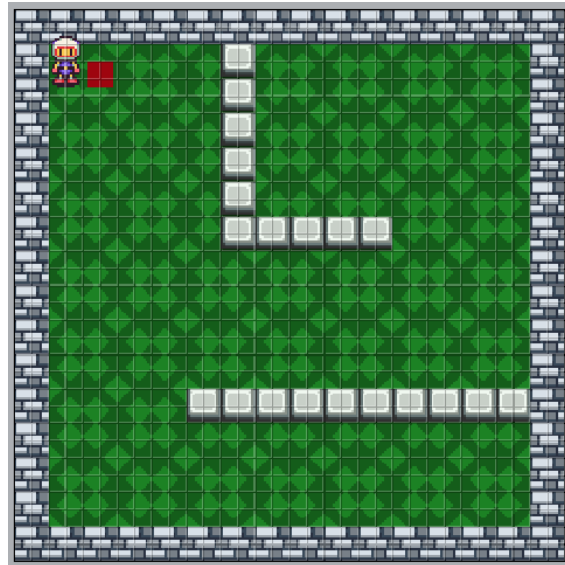
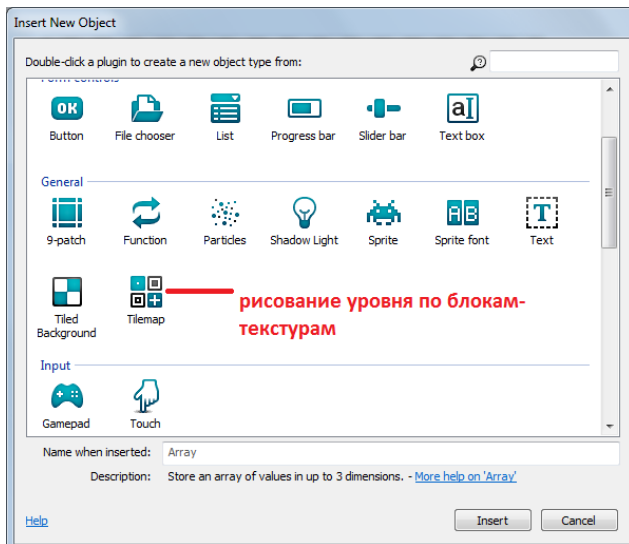
Keyboa... Up arrow is down	Sprite2	Start ignoring 8Direction user input
Keyboa... Left arrow is down		Add action
Keyboa... Up arrow is down	Sprite2	Start ignoring 8Direction user input
Keyboa... Right arrow is down		Add action
Keyboa... Down arrow is down	Sprite2	Start ignoring 8Direction user input
Keyboa... Left arrow is down		Add action
Keyboa... Down arrow is down	Sprite2	Start ignoring 8Direction user input
Keyboa... Right arrow is down		Add action

В Paint нарисовал текстуру поля 32 x 32 пикселя и сохранил на компьютере.

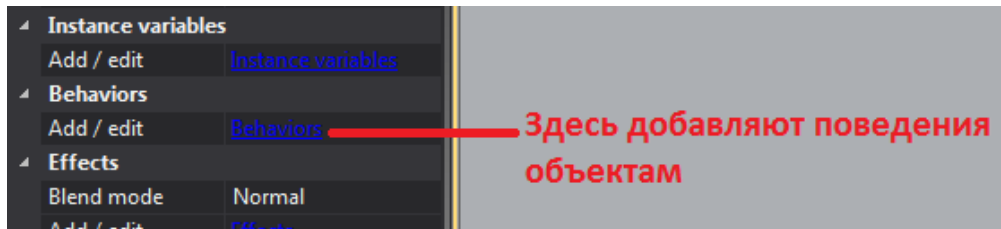
Создал "бесконечное дублирование картинки при растягивании" - *tiled background* и открыл в нем мой нарисованный пол.



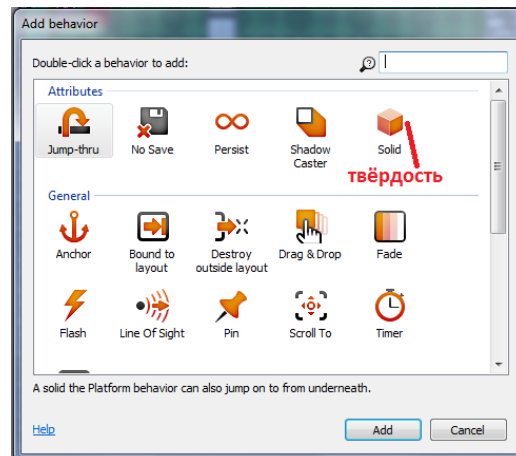
Далее я скачал спрайты стен и создал Tilemap, чтобы нарисовать (добавить) уровень блоками стен.



Добавляем поведение твёрдость нашему Tilemap.

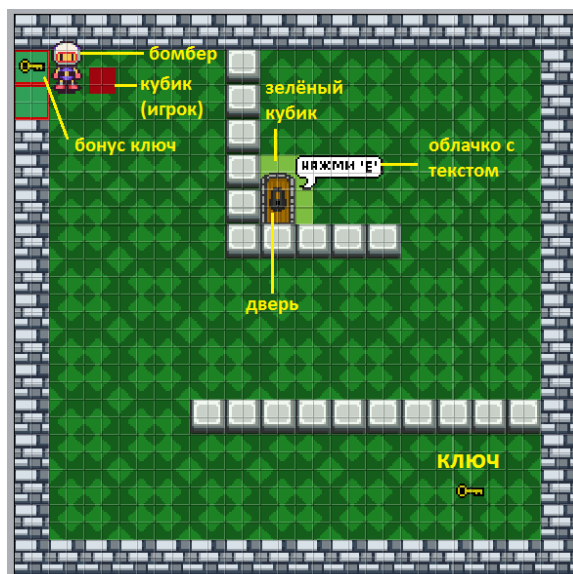


Выбираем твёрдость (Solid)

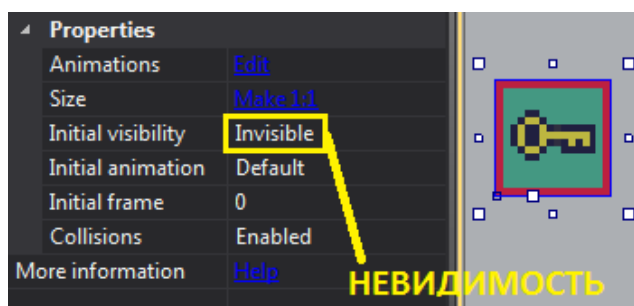


Теперь стены твердые и наш персонаж сквозь них пройти не сможет.

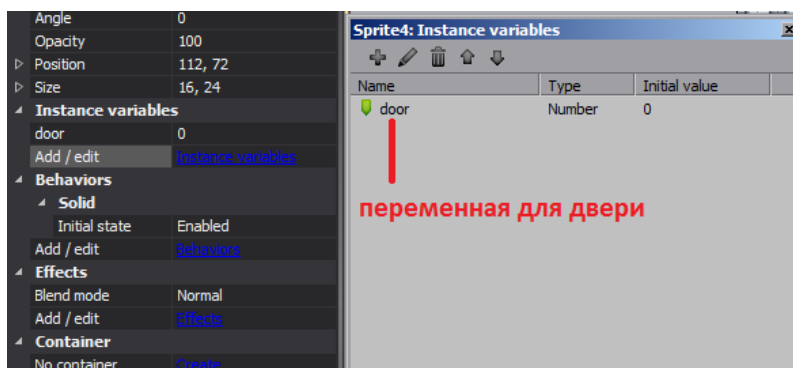
Далее в Paint нарисовал дверь, ключик, блок бонуса с ключом, зелёный кубик (некая зелёная зона, которая будет стоять возле дверей для отображения облачка с текстом, который будет подсказывать игроку что нужно сделать), облачко с анимациями текста. Создал спрайты, и добавил в них свои рисунки, и поставил в нужное место на уровне.



У двери два кадра анимации (открытая/закрытая). В настройках объекта бонус "ключ" включил ему невидимость. Чтобы бонус становился видимым когда "Бомбер" прикоснется к ключу на уровне. Но это мы сделаем в листе событий.



Добавил двери локальную переменную (это такая переменная, которая есть у конкретного объекта и работает на конкретном уровне где есть этот объект, у переменной изменяется значение).



Для удобства создал группу "Key" (ключ).

Добавил событие, если "бомбер" дотрагивается до ключа, то ключ уничтожается и бонус "ключ" становится видимым и переменная двери *door* ставит себе значение 1. (ключ взят).

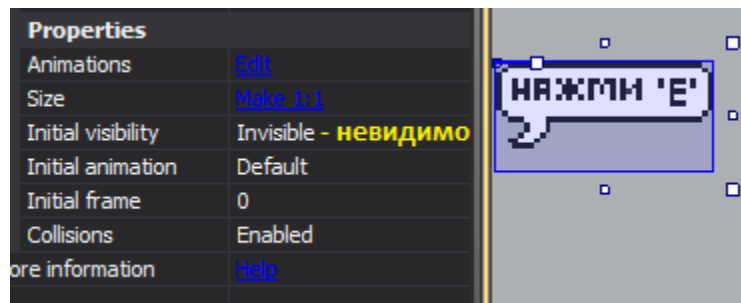
Sprite бомбер дотронулся до ключа	On collision with Sprite3	Sprite3	Destroy - ключ уничтожается
		Sprite7	Set Visible - бонус ключ видим
		Sprite4	Set door to 1 - переменная двери door ставит значение 1
Add action			

Если бонус видим и блок персонажа дотрагивается до зелёного блока, и если мы нажимаем на клавиатуре клавишу "E", то тогда локальная переменная у двери ставит себе число 2 (ключ использован) и дверь меняет кадр анимации на 1 (открыто).

У двери 2 кадра анимации 0-закрыто, 1-открыто

бонус ключ видим Sprite7	Is visible	Sprite4	Set door to 2 - переменная двери door ставит значение 2
Sprite2 кубик (игрок) дотронулся до зелёного кубика	Is overlapping Sprite6	Sprite4	Set animation frame to 1 - анимация двери меняет кадр на 1 (открыто)
при нажатии на клавишу "E" Keyboa...	E is down	Add action	

Для облачка с текстом ставлю в настройках объекта изначальную невидимость.




Создаю группу "tablo" в которой создаю события:



Если переменная двери door равна 0 (ключ не взят) и кубик игрока (красный) дотрагивается до зелёного кубика, тогда облачко с текстом становится видимым и делает анимацию "close" (в которой написано "Где ключ?").

tablo			
Sprite4	door = 0	Sprite5	Set Visible
Sprite2	Is overlapping Sprite6	Sprite5	Set animation to "close" (play from beginning)
Add action			

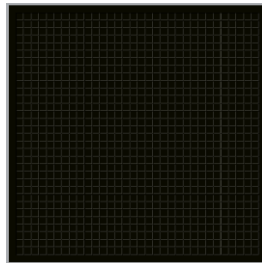
Если переменная двери равна 1 (ключ взят) то облачко видимое и делает анимацию *default* (в которой написано нажми "E").

Sprite4	door = 1	Sprite5	Set Visible
Sprite2	Is overlapping	Sprite5	Set animation to "default" (play from beginning)
			Add action
	Sprite6		

Если не дотрагивается до зелёного кубика то облачко невидимо.

Sprite2	 Is overlapping	Sprite5	Set Invisible
			Add action
	Sprite6		

Добавим на слой выше два чёрных спрайта на весь уровень и даём им поведения *Fade* (плавное затухание).



И в настройках поведения объекта у первого черного спрайта включаю активность при старте. (Active at start - *Yes*)

Fade	
Active at start	Yes
Fade in time	0 время появления
Wait time	0
Fade out time	1 время исчезания
Destroy	No
Add / edit	Behaviors

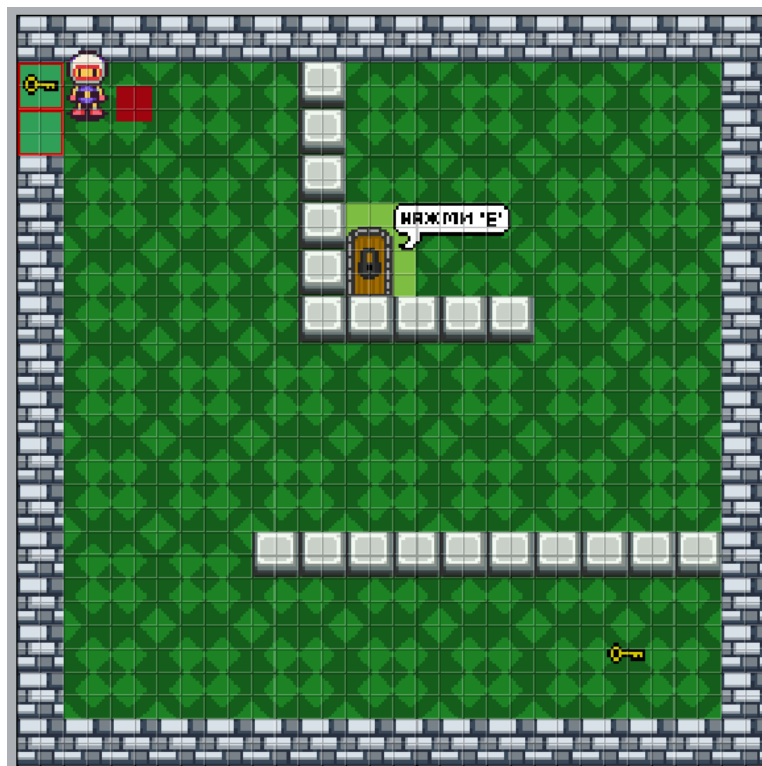
А в настройках поведения объекта у второго черного спрайта выключаю активность при старте. (Active at start - *No*) и ставлю цифры Fade in time 1

Fade	
Active at start	No
Fade in time	1 время появления
Wait time	0
Fade out time	0 время исчезания
Destroy	No
Add / edit	Behaviors

Если переменная двери door равна 2 (ключ использован), тогда облачко делает анимацию (в которой написано "Открыто!") и облачко становится видимым и черный спрайт 2 становится видимым - активируется *Fade* поведение. Далее игра ждёт 1 секунду (чтобы проиграло затухание) и переходит на следующий уровень.

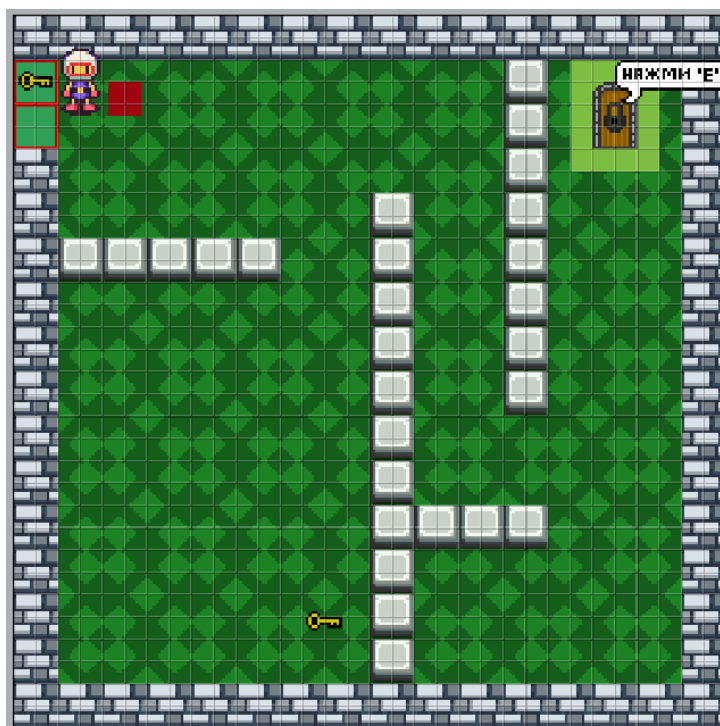
Sprite4	door = 2	Sprite5	Set animation to "open" (play from beginning)
Sprite2	Is overlapping	Sprite5	Set Visible
Sprite6		black2	Set Visible
		black2	Fade: start fade
		System	Wait 1.0 seconds
		System	Go to next layout
Add action			

Готовый первый уровень!



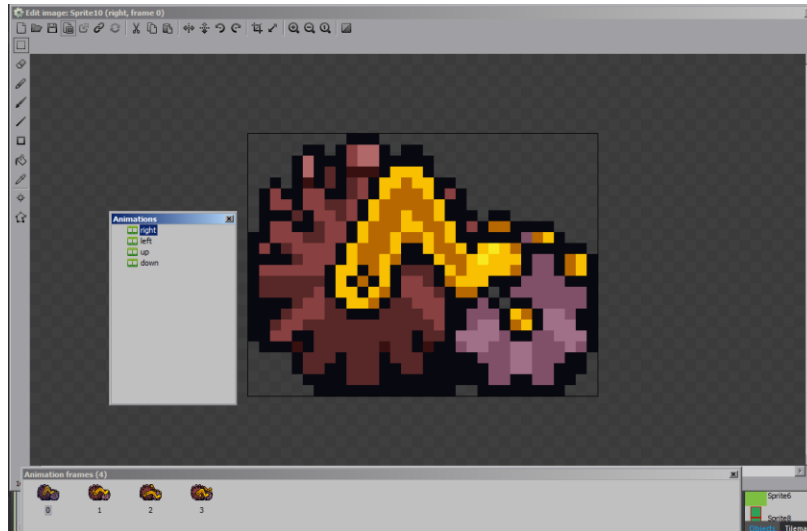
Переходим к созданию второго уровня.

Переносим все объекты на второй уровень и меняем местоположение стен.



Теперь добавим врага.

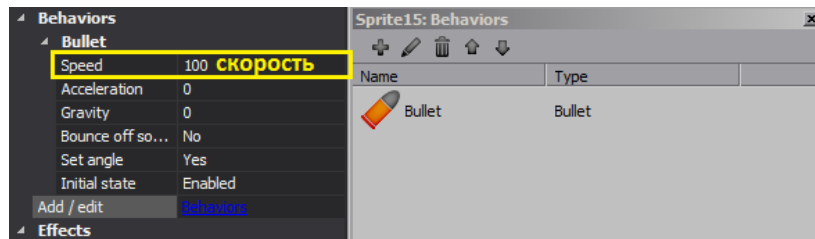
Создаём новый спрайт и вставляем туда скачанного с интернета врага и его анимации в разных направлениях. (Вверх, низ, лево, право.)



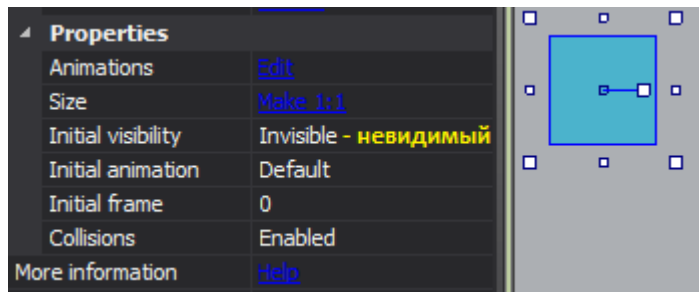
Добавим спрайт - синий кубик, чтобы к нему присоединить в будущем нашего врага. А так же добавляем спрайты со стрелочками, чтобы потом в листе событий сделать так: чтобы когда синий кубик дотронулся до стрелочек, то он менял своё направление, а враг менял анимацию в зависимости от направления, но пока, добавим просто спрайты.



Добавляю поведение кубику врага пуля. И ставим скорость на 100.

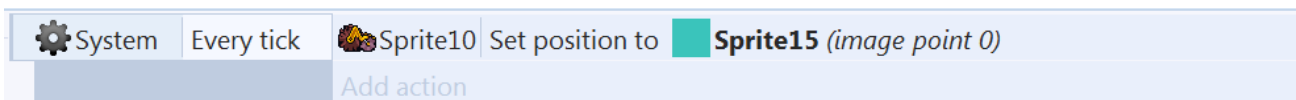


Делаю кубик невидимым в настройках объекта.

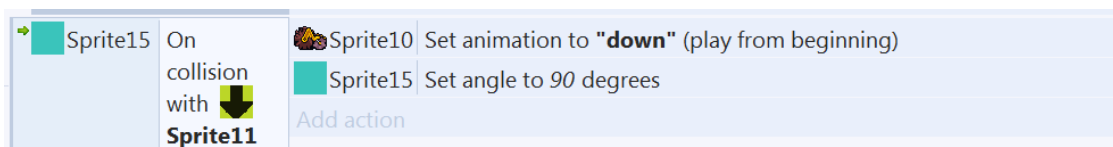


Теперь создаём события.

Враг всегда стоит на позиции кубика врага. (присоединён к кубику).



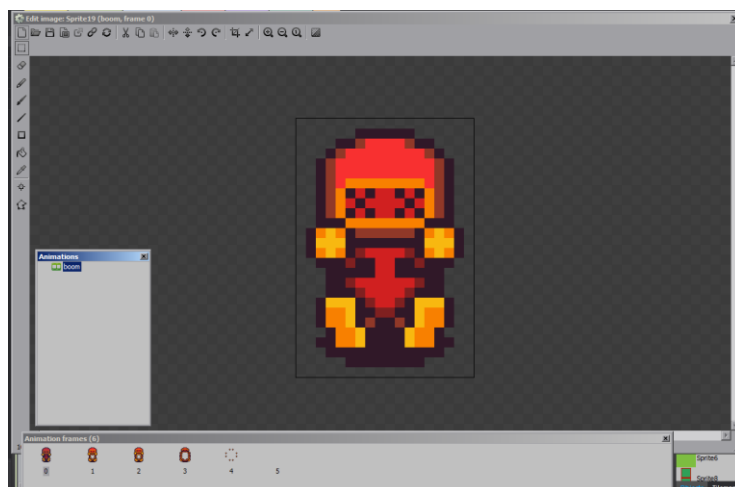
Если кубик врага дотрагивается до стрелки вниз, то враг делает анимацию вниз. И кубик врага разворачивается на 90 градусов. Тем самым кубик летит вниз благодаря поведению пули.



Так делаем со всеми четырьмя стрелками.



Sprite15	On collision with  Sprite11	Sprite10 Set animation to "down" (play from beginning)	Sprite15 Set angle to 90 degrees
Sprite15	On collision with  Sprite12	Sprite10 Set animation to "up" (play from beginning)	Sprite15 Set angle to 270 degrees
Sprite15	On collision with  Sprite13	Sprite10 Set animation to "left" (play from beginning)	Sprite15 Set angle to 180 degrees
Sprite15	On collision with  Sprite14	Sprite10 Set animation to "right" (play from beginning)	Sprite15 Set angle to 0 degrees

Добавим смерть персонажу. Создам спрайт и открою скачанную с интернета анимацию смерти персонажа.

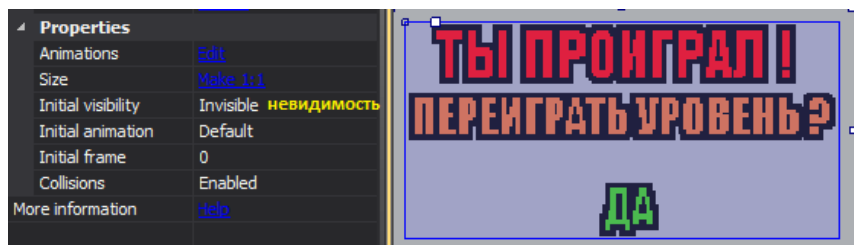


Создаю для удобства группу в листе событий GAME OVER.

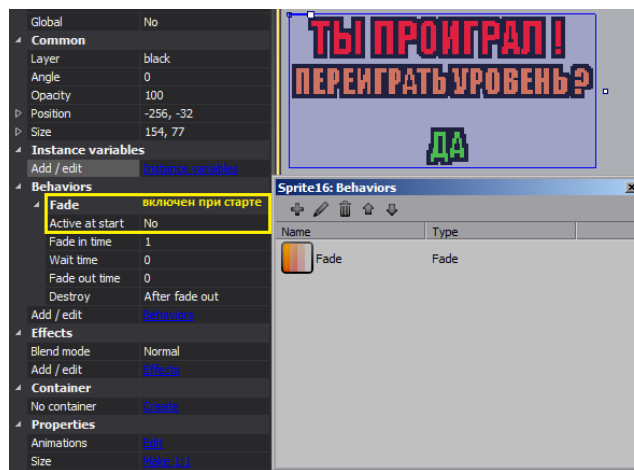
В ней создаю событие: если "Бомбер" дотрагивается до врага, то "Бомбер" создает спрайт с анимацией смерти "Бомбера", и сам "Бомбер" уничтожается вместе с кубиком.

GAME OVER			
Sprite	On collision with  Sprite10	Sprite	Spawn  Sprite19 on layer "player" (<i>image point 0</i>)
		Sprite2	Destroy
		Sprite	Destroy

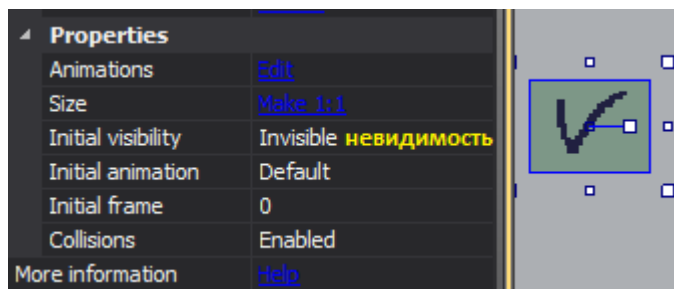
Добавляю спрайт с надписью "ТЫ ПРОИГРАЛ! ПЕРЕИГРАТЬ УРОВЕНЬ? ДА" и сделаю его невидимым в настройках объекта.



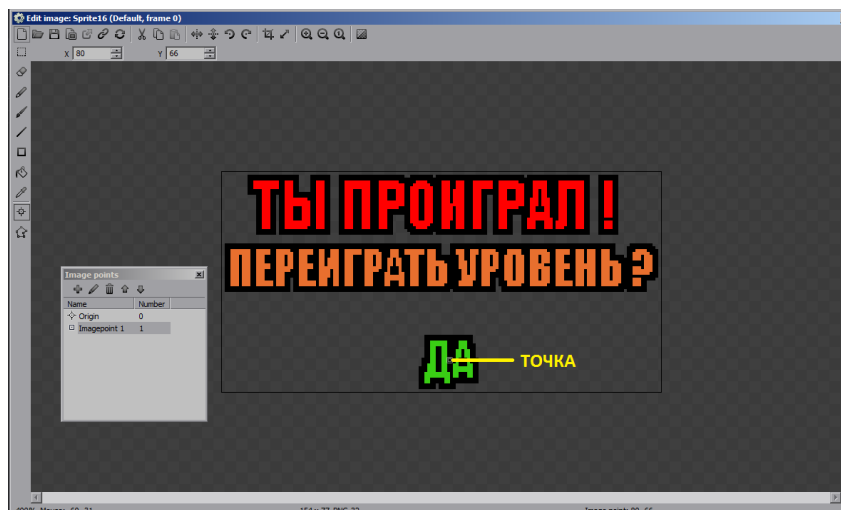
Дам поведение *Fade* спрайту с надписью, чтобы потом он постепенно появлялся и выключу поведение при старте уровня. (Active at start - *NO*)



Добавляю спрайт с подтверждением и сделаю его невидимым в настройках объекта.

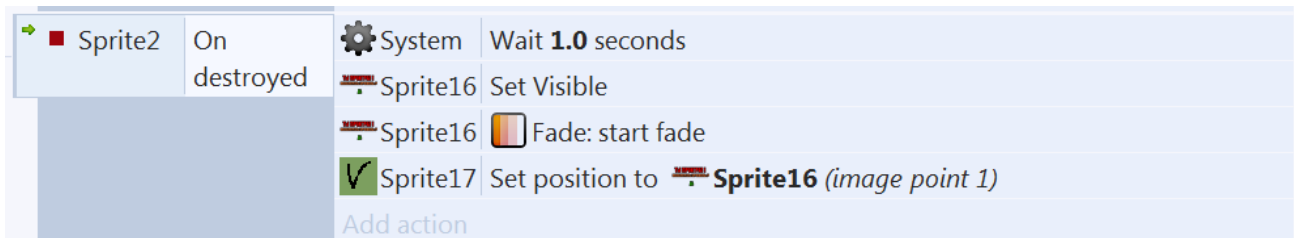


Теперь добавляю точку для сцепления объектов к спрайту надписи на слово "ДА"



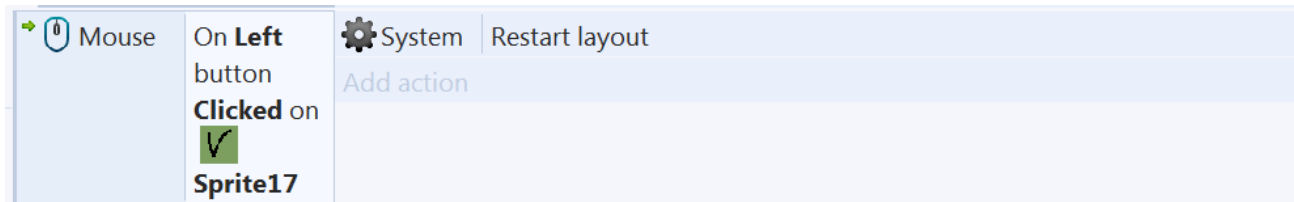
Создаю ещё одно событие:

Если блок игрока уничтожен, то система игры ждёт 1 секунду, для того, чтобы мы увидели как проигралась анимация смерти персонажа. После того, как прошла 1 секунда спрайт с надписью "ТЫ ПРОИГРАЛ! ПЕРЕИГРАТЬ УРОВЕНЬ? ДА" становится видимым и у него включается поведение *Fade* (это поведение отвечает за плавное появление или затухание объекта), и невидимый спрайт с галочкой становится на позицию точки спрайта с надписью, а именно, становится на точку где написано слово "ДА". Тем самым невидимая кнопка стоит на слове "ДА".

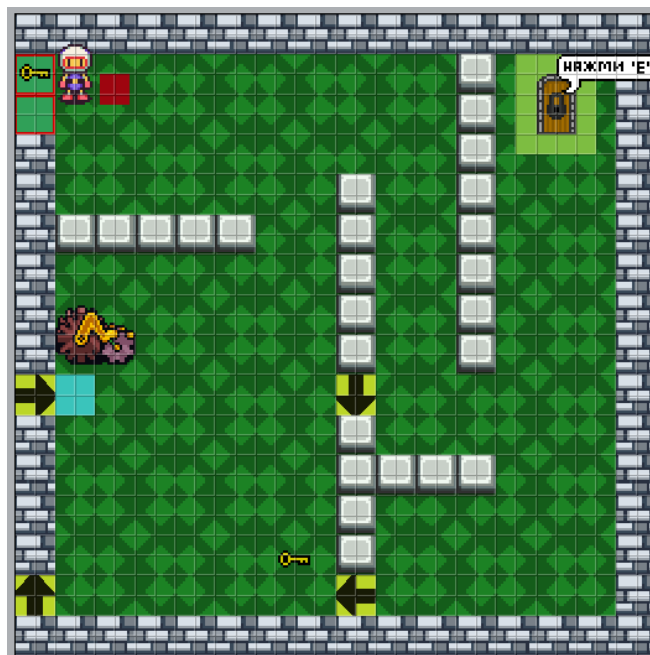


Создаю событие:

Если левой кнопкой мыши кликнуть на невидимый спрайт. То система игры перезапустит уровень.



Второй готовый уровень!

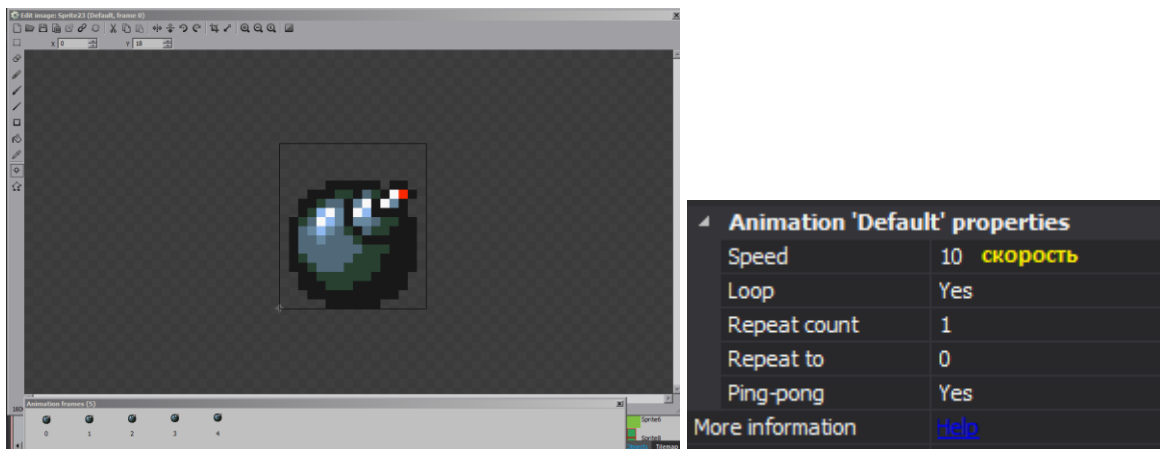


Переходим к созданию третьего уровня.

Переносим все нужные объекты на третий уровень и меняем местоположение стен.

Создадим бомбу!

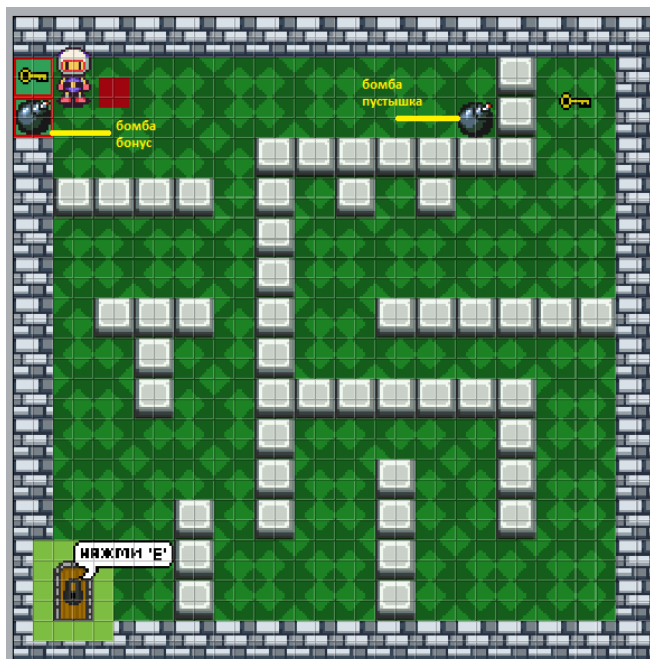
Создам спрайт и открою в нём скачанный с интернета спрайт бомбы. И сделаю ему по кадрам анимацию движения вверх и вниз. Поставлю скорость анимации 10. Это будет просто бомба, которую можно будет подобрать - пустышка.



Теперь добавлю спрайт бонуса бомбы



Поставлю спрайты в нужное место на уровне.

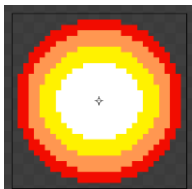


Далее добавлю ещё один спрайт бомбы.

Эта бомба уже будет настоящей, и она будет взрываться! Её "Бомбер" сможет ставить на уровне. У неё есть анимация тряски. Что бы она тряслась когда "Бомбер" её поставит.

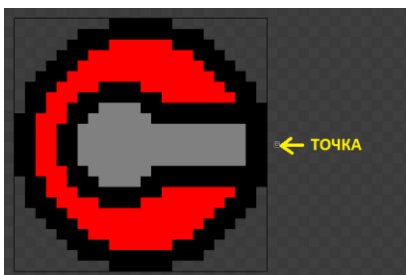


Добавлю спрайт взрыва с анимацией взрыва! Он понадобится для бомбы.

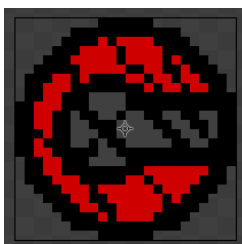


Добавлю спрайт пушки которая в будущем будет стрелять!

У неё есть анимация взрыва и поставлена точка на кончик пушки, там будет появляться пуля и лететь.

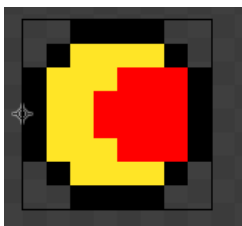


Добавлю спрайт взорванной пушки!

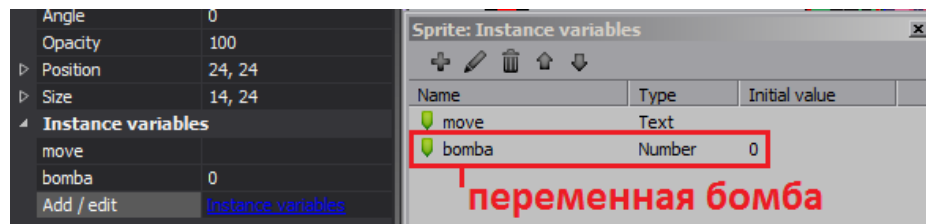


Добавлю спрайт пули которой будет стрелять пушка!

У неё стоит точка в начале пули, это то место откуда она будет появляться.

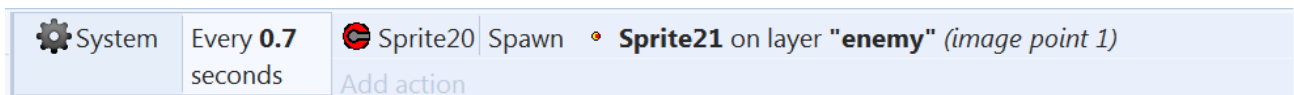


Добавлю локальную переменную "Бомберу" (это такая переменная, которая есть у конкретного объекта и работает на конкретном уровне где есть этот объект) и назову её bomba. Она будет хранить количество бомб у "Бомбера".

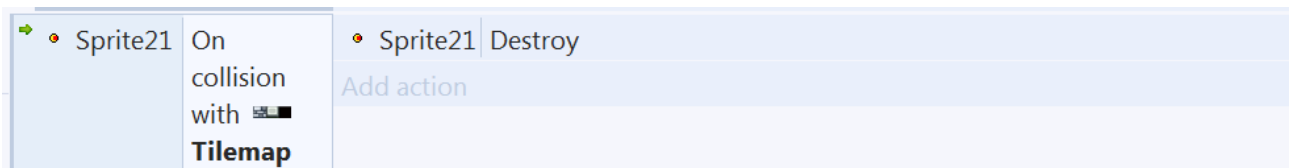


Создаю события:

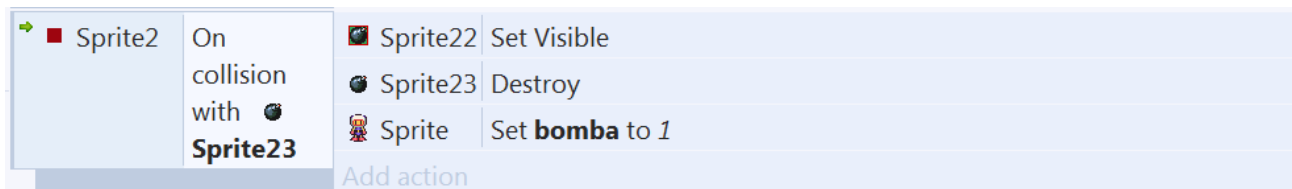
Всегда каждую 0.7 секунду пушка создаёт пулю.



Если пуля дотронется до *TileMap* (стена), то она уничтожится.



Если кубик игрока дотронется до бомбы-пустышки, то бомба бонус станет видимой, и бомба пустышка уничтожится. Локальная переменная "Бомбера" bomba ставит значение 1. (подобрал бомбу)



Если нажать на клавишу пробел - бомба бонус видима и переменная "Бомбера" bomba равна 1, тогда переменная bomba ставит значение 0 и кубик игрока создаёт настоящую бомбу, тогда система игры ждёт 3 секунды, бомба создаёт взрыв и уничтожается. Система игры ждёт 0.6 секунд для того, чтобы проигралась анимация взрыва и переменная "Бомбера" bomber ставит значение 1 (бомба перезарядилась у игрока и её снова можно поставить).

Keyboa...	On Space pressed	Sprite	Set bomba to 0
Sprite22	Is visible	Sprite2	Spawn Sprite24 on layer " player " (<i>image point 0</i>)
Sprite	bomba = 1	System	Wait 3.0 seconds
		Sprite24	Spawn Sprite25 on layer " player " (<i>image point 0</i>)
		Sprite24	Destroy
		System	Wait 0.6 seconds
		Sprite	Set bomba to 1
		Add action	

Если пушка дотрагивается до взрыва бомбы, тогда пушка делает анимацию взрыва и создает сломанную пушку.

Sprite20	On collision with Sprite25	Sprite20	Set animation to " boom " (play from beginning)
		Sprite20	Spawn Sprite26 on layer " labirint " (<i>image point 0</i>)
		Add action	

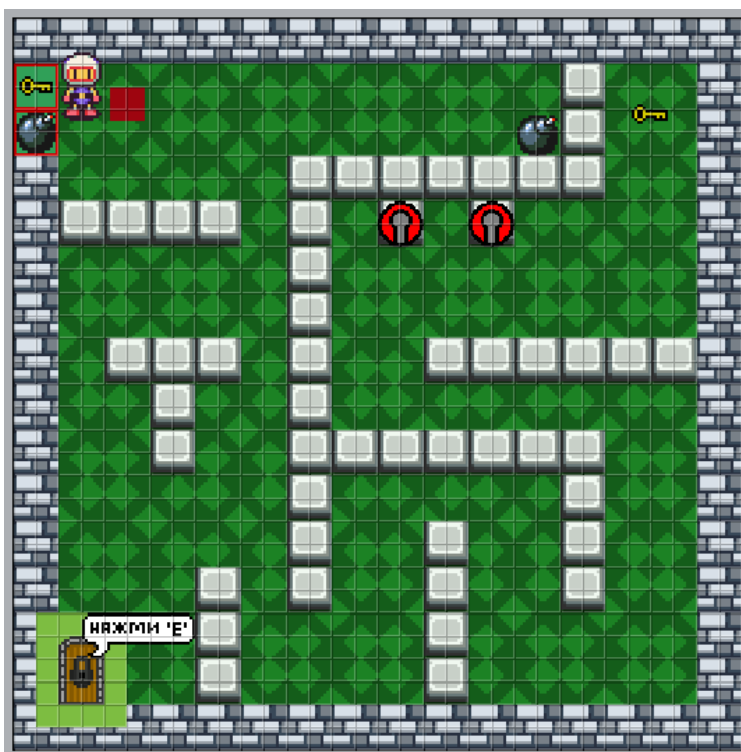
Если пушка закончила анимацию взрыва, то она уничтожается. (В итоге на её месте остается только взорванная пушка)

Sprite20	On animation " boom " finished	Sprite20	Destroy
		Add action	

Если анимация взрыва бомбы закончилась, то анимация уничтожается.

Sprite25	On animation " boom " finished	Sprite25	Destroy
		Add action	

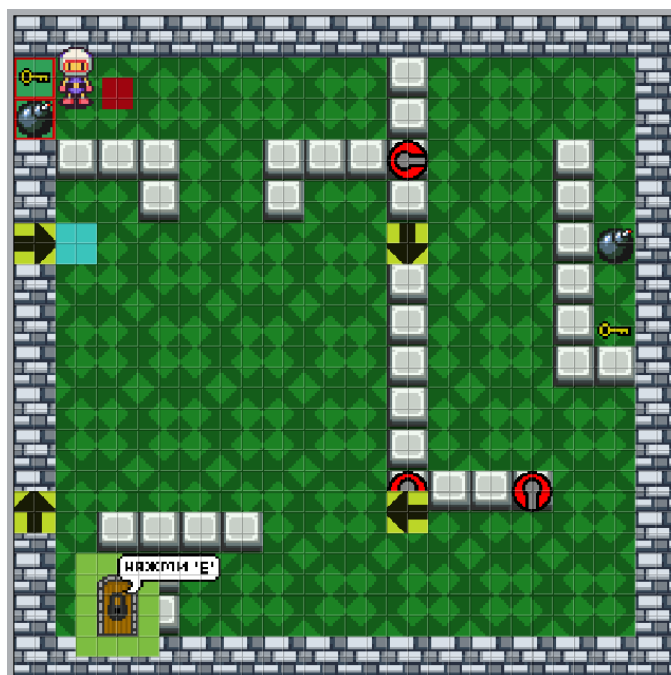
Готовый третий уровень!



Создаю четвёртый уровень.

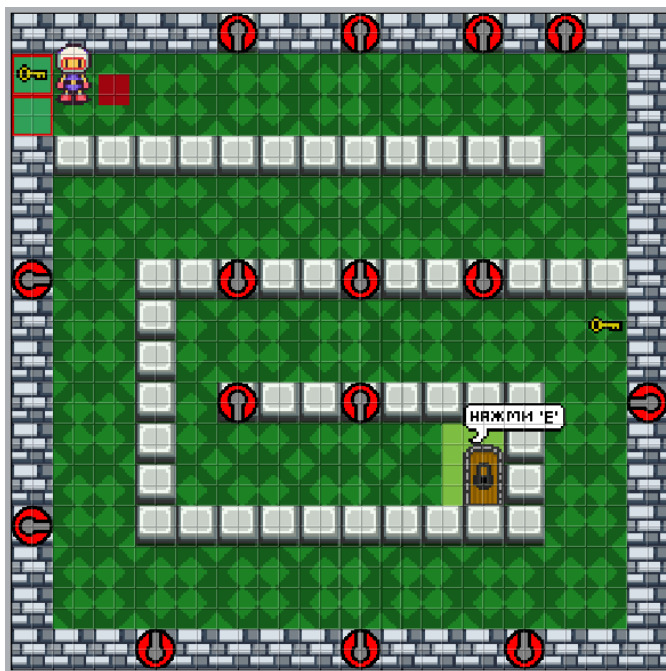
Переношу туда все нужные объекты, меняю расположение стен.

Там всё само работает. Ничего нового не добавляю, просто немного усложняю уровень расположением врагов.



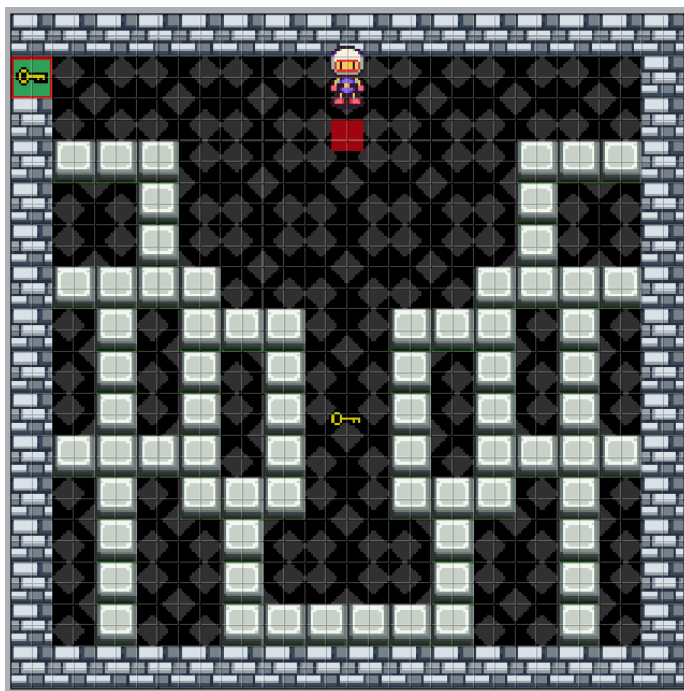
Создаю пятый уровень.

Переносу туда все нужные объекты, меняю расположение стен. В нем, так же всё работает само. Ничего нового не добавляю. Просто немного усложняю уровень расположением врагов.



Создаю шестой - заключительный уровень.

Переносу туда все нужные объекты, меняю цвет пола и меняю расположение стен. Уровень стал похож на подземелье.



Далее создаю спрайт сундука скачанного с интернета.

У него есть кадр анимации где сундук открыт.



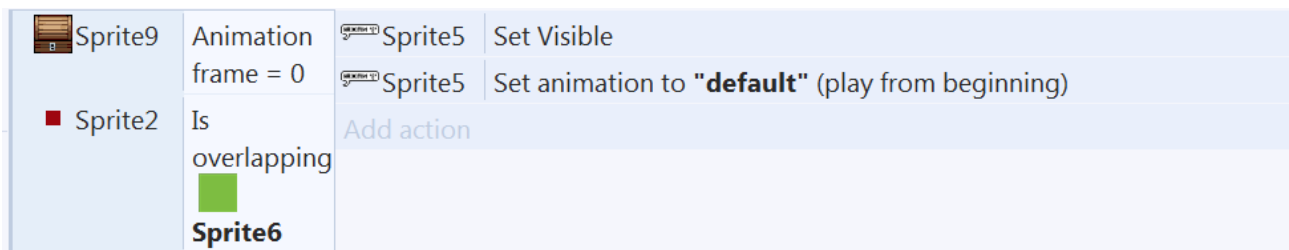
Ставлю его в нужное место на уровне.



Создаю событие в группе tablo.

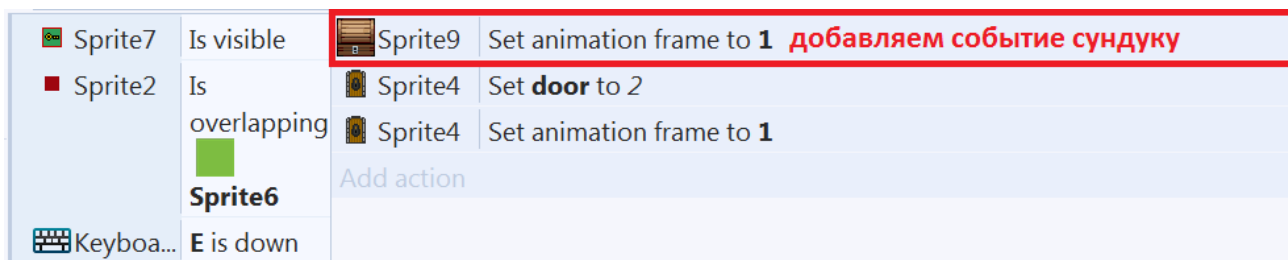
Сейчас настроим облачко с текстом для сундучка.

Если кадр анимации сундука равен 0 (закрытый) то облачко с текстом становится видимым и делает анимацию с текстом - "Нажми E".



Добавляем в группе key событие:

Если бонус ключ видим и кубик игрока дотрагивается до зелёного кубика и нажимаем клавишу "E", тогда сундук ставит кадр анимации 1. (Открытый)

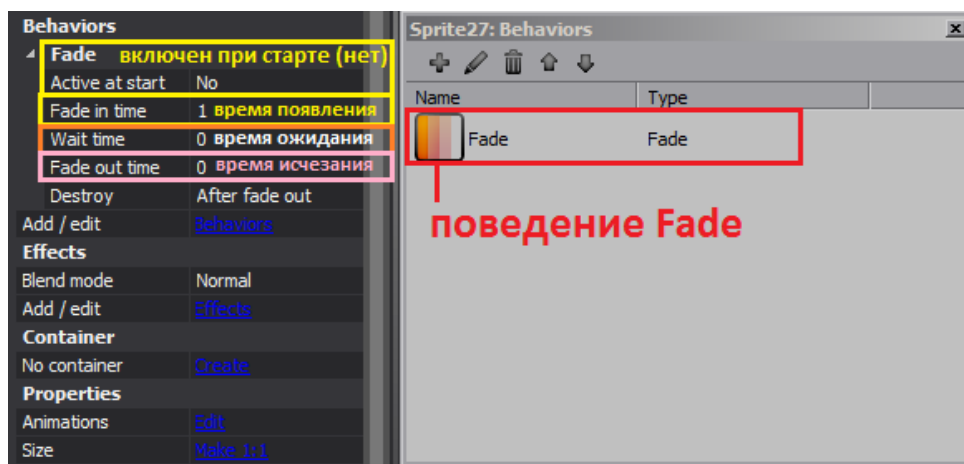


Добавляю спрайт с текстом "СОКРОВИЩЕ НАЙДЕНО! МОЛОДЕЦ!".

Делаю его невидимым в настройках объекта.



Даю ему поведение *Fade*, и настраиваю поведение на появление - 1 сек. Отключаю активацию поведения при старте уровня.



Ставлю в нужное место на уровне.

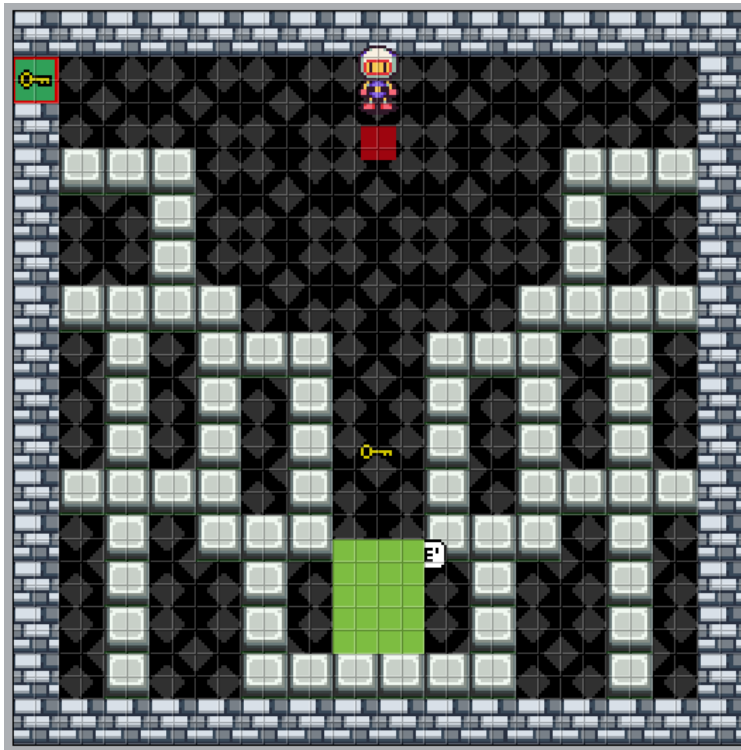


Создаю событие в группе key:

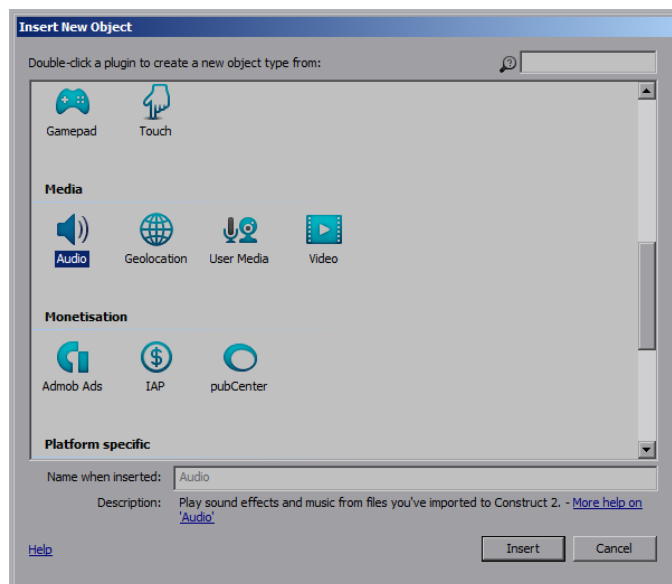
Если кадр анимации сундука равен - 1 (открытый) и если кубик игрока дотрагивается до зелёного кубика (это некая зелёная зона, которая стоит возле сундука и в прошлых уровнях возле дверей, которая отвечает за появление облачка с текстом - оно подсказывает что нужно сделать), тогда облачко с текстом становится невидимым и спрайт с текстом "СОКРОВИЩЕ НАЙДЕНО! МОЛОДЕЦ!" становится видимым и у него включается поведение *Fade*. (плавное появление/затухание).

Sprite9	Animation frame = 1	Sprite5	Set Invisible
Sprite2	Is overlapping Sprite6	Sprite27	Set Visible
		Sprite27	Fade: start fade
Add action			

Готовый последний шестой уровень!

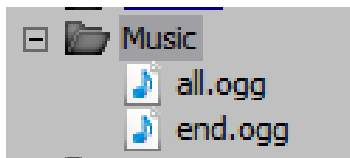


Ну и напоследок, давайте добавим музыку, что бы скучно не было. Добавляю плагин аудио.



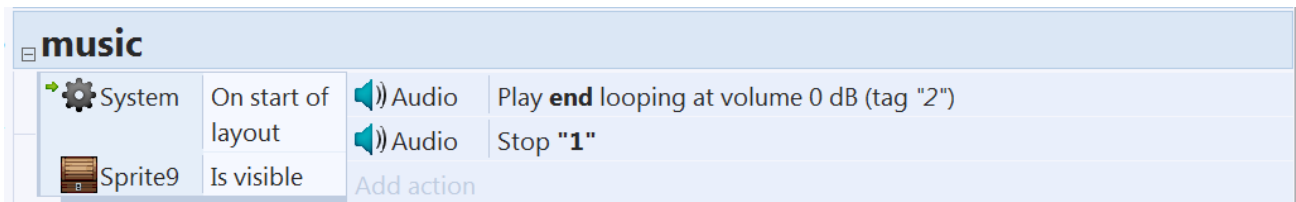
Теперь добавляю музыку в проект в папку "music".

all - будет играть на всех уровнях, end - будет играть на последнем уровне с сокровищем.



Создаю группу "Music" и добавляю в неё 2 события:

При старте уровня если сундук видимый, то плагин аудио включает музыку end, и останавливает музыку all.



При старте уровня если дверь видимая, тогда плагин аудио включает музыку all.

