МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Высший колледж информатики

Кафедра интеллектуальных систем теплофизики

Направление подготовки 15.03.06 Мехатроника и робототехника

Направленность (профиль): Мехатроника и робототехника

**КУРСОВАЯ РАБОТА**

**Беспалова Сергея Вячеславовича**

**Грищенко Александра Михайловича  
Солопова Ильи Руслановича**

Тема работы:

**ИГРа «RUSH Space»**

|  |  |
| --- | --- |
| **«К защите допущена»** | **Руководитель ВКР** |
| Заведующий кафедрой, | ученая степень, звание |
| ученая степень, звание | должность, место работы в НГУ |
| ………………/………….. | ………………/………... |
| (ФИО) / (подпись) | (ФИО) / (подпись) |
| «……»………………20…г. | «……»………………20…г. |
|  |  |

Новосибирск, 2022

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc103514653)

[1 ПОСТАНОВКА ЗАДАЧИ 4](#_Toc103514654)

[2 АНАЛОГИ 5](#_Toc103514655)

[3 АППАРАТНАЯ ЧАСТЬ 7](#_Toc103514656)

[4 ПРОГРАММНАЯ ЧАСТЬ 10](#_Toc103514657)

[ЗАКЛЮЧЕНИЕ 13](#_Toc103514658)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 14](#_Toc103514659)

[ПРИЛОЖЕНИЯ 15](#_Toc103514660)

ВВЕДЕНИЕ

По заданию к курсовому проекту по дисциплине «Digital platforms» нужно было спроектировать и создать игру в жанре Shoot 'em up на электронных схемах, использовав процессор CdM-8 и его ассемблерный язык.

Shoot 'em up – это вид компьютерных игр, в котором управляемый игроком персонаж чаще всего представлен в виде космического корабля или другого транспортного средства, основная цель которого – победить множество врагов, используя стрельбу. Врагами в таких играх, как правило, выступают различные инопланетяне или монстры, атакующие игрока, стреляя в него или как-то иначе. Традиционно в таких шутерах используется вид сверху или сбоку, а для успешного прохождения важна хорошая реакция, чтобы уклоняться от вражеского огня.

В соответствие с поставленной целью необходимо было решить следующие задачи:

* Изучить примеры игр жанра Shoot 'em up (аналоги);
* Изучить и проанализировать информацию о процессоре, его возможностях, командах и инструкциях;
* Определить функциональные требования.

1 ПОСТАНОВКА ЗАДАЧИ

Цель данной курсовой работы – проектирование и создание игры жанра Shoot 'em up на основе электросхемы с включенным в неё 8-битным процессором CdM-8.

Ниже представлены функциональные требования:

1. Управляемое движение игрока;
2. Неуправляемое движение противников;
3. Стрельба игрока;
4. Стрельба монстров;
5. Уничтожение монстров;
6. Управление с клавиатуры (движение игрока, стрельба игрока);
7. Возможность выигрыша;
8. Возможность проигрыша.

Методы решения поставленных задач…

2 АНАЛОГИ

В процессе создания проекта нами были изучены игры жанра Shoot 'em up. Рассмотрим некоторые из них.

1. Японская автоматная аркада «Space Invaders», выпущенная в 1978 году. Её можно назвать одной из первых в данном жанре. Данная игра имеет все заданные в предыдущем пункте функциональные требования, а также дополнительные, например, счётчик жизней и количества набранных очков, звуковые эффекты, бесконечное количество «волн» врагов, защита в виде «бункеров» от выстрелов инопланетян.

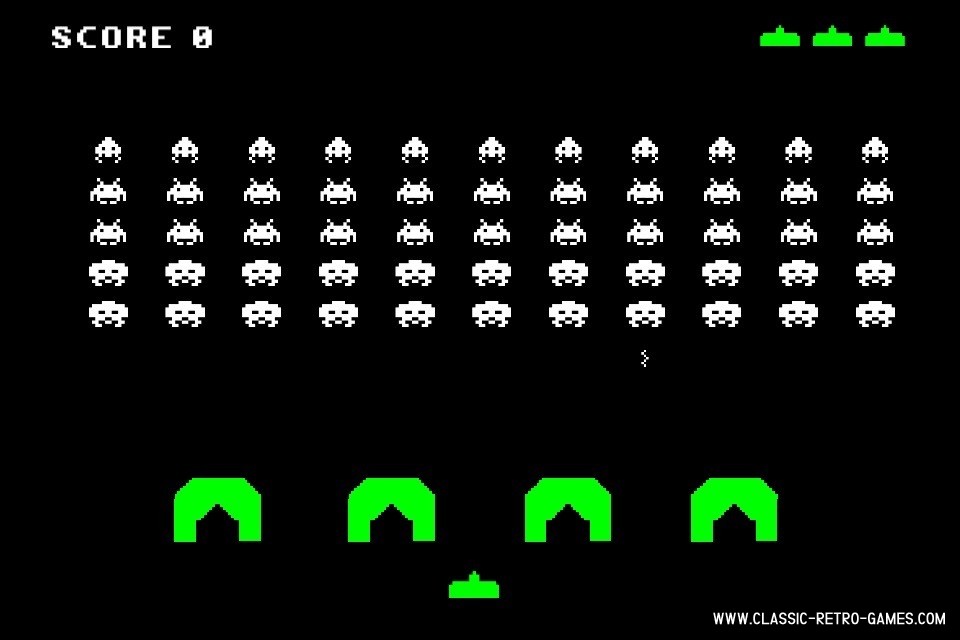


Рисунок 1 – скриншот игрового процесса «Space Invaders»

1. Еще одна японская аркада, выпущенная для игровых автоматов в 1981 году – «Galaga». Её можно назвать улучшенной и более современной версией предыдущей игры. В «Galaga» были созданы новые игровые механики, такие как притягивающий луч инопланетян, способный лишить игрока возможности управления на короткий промежуток времени и одной жизни, пикирование монстров на игрока по различным траекториям. Также это одна из первых игр с цветной RGB-графикой.

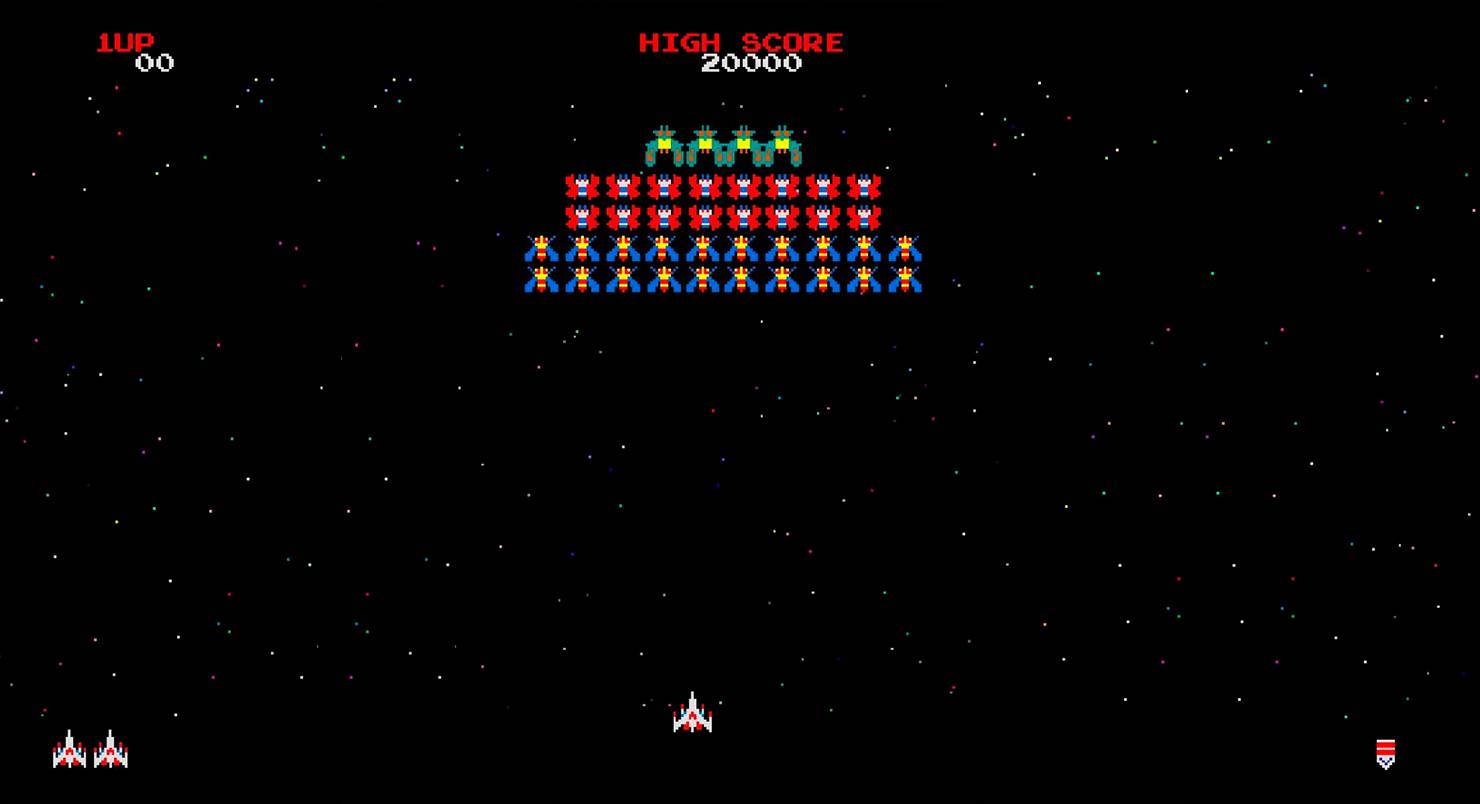


Рисунок 2 – скриншот игрового процесса «Galaga»

Изучив и проанализировав данные примеры, нашей командой было принято решение создавать проект, основываясь на дизайне и наполнении игры «Space Invaders». Из-за ограниченного количества времени, выделенного на разработку нам нужен был не очень сложный и ресурсозатратный вариант, который было бы возможно реализовать, используя CdM-8 и наши знания в области схемотехники.

3 АППАРАТНАЯ ЧАСТЬ

Аппаратная часть нашего проекта представляет из себя логические электрические схемы, созданные в программе Logisim, которая позволяет моделировать и редактировать их с помощью удобного графического интерфейса. Рассмотрим наши разработки.

4 ПРОГРАММНАЯ ЧАСТЬ

Программная часть нашего проекта была реализована с помощью ассемблерного языка процессора CdM-8 в интегрированной среде разработки CocoIDE, созданной специально для разработки кода, выполняемого этим процессором. Рассмотрим наши алгоритмы.

**Wall destruction**

Алгоритм, реализующий инициализацию и разрушение защитных стенок при попадании в них пуль монстров. Входные данные – это два четырёхбайтовых массива: строка с текущим состоянием стенок и нулевая строка с единичным битом попадания. Выходные данные – строка с состоянием стенок после попадания или промаха.

Первым шагом инициализируются регистры, находящиеся по адресам F0-F3, значением 3С (00111100 в двоичной системе). Эта константа – выводимая на экран стенка из 4 пикселей. На втором шаге в регистры общего назначения загружаются входные данные, инициализируется счётчик. Затем в цикле, пока счётчик больше нуля, выполняется операция исключающего «или» между соответствующими байтами входных массивов, и результат сохраняется в регистр, в который были загружены стенки. Таким образом, итог – это битовая строка с состоянием «защит» после попадания или промаха. По окончании цикла происходит переход на второй шаг алгоритма. На рисунке \_ представлена блок-схема алгоритма инициализации и разрушения стен.

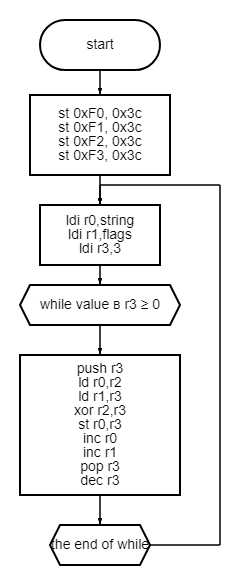


Рисунок \_ - блок-схема алгоритма инициализации и разрушения стен

**Monsters shooting**

Алгоритм, реализующий вычисление координаты (номера столбца), в которой монстр произведёт выстрел. Входные данные – два однобайтовых числа: номер пришельца, совершающего выстрел (от 0 до 3) и значение смещения монстра (-1 или 1). Выходные данные – однобайтовый номер столбца.

Первым шагом в регистры общего назначения загружаются номер пришельца, который будет использоваться в качестве счётчика, а также константы 8 и 0. Затем после вычислений в цикле получится значение, равное номеру монстра, совершающего выстрел, умноженное на 8. После этого к нему прибавляется значение смещения и константа 4, а результат (получившийся номер столбца) сохраняется в регистр. Далее происходит переход на первый шаг алгоритма. На рисунке \_ представлена блок-схема алгоритма вычисления номера столбца, в котором будет произведён выстрел.

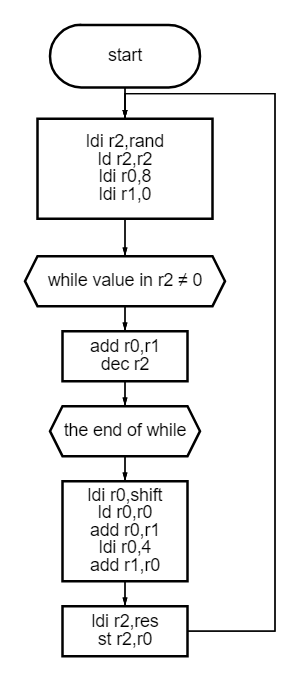


Рисунок \_ - блок-схема алгоритма вычисления номера столбца, в котором будет произведён выстрел

**Hit**

Алгоритм, реализующий инициализацию состояния пришельцев обработку попадания пули игрока в них. Входные данные – байт, хранящий состояния инопланетян (если бит равен единице, то монстр жив, если нулю, то нет) и четырёхбайтовый массив – нулевая строка с единичным битом попадания. Выходные данные – байт, хранящий новые состояния врагов.

На первом шаге инициализируется байт состояний монстров значением FF (11111111 в двоичной системе). Вторым шагом в регистры общего назначения загружаются нулевая строка с единичным битом попадания, а также константа 0, инициализируется счётчик. Затем в цикле ищется не равный нулю байт в массиве и вычисляется номер пришельца, в которого было осуществлено попадание. После этого путём побитовых сдвигов влево вычисляется однобайтовая маска, где единственный единичный бит имеет индекс, равный вычисленному ранее номеру инопланетянина, который находится в нижнем ряду. На следующем шаге происходит сравнение этой маски и байта, хранящего состояния врагов, путём выполнения операции побитового «и». Если результат равен нулю, то монстра в нижнем ряду нет, поэтому происходит еще 4 побитовых сдвига влево. Далее выполняется операция исключающего «или» между новой маской и байтом состояний, а результат сохраняется в регистр. После сохранения происходит переход на второй шаг. На рисунках \_ и \_ представлены части блок-схемы алгоритма инициализации состояния пришельцев и обработки попадания в них.

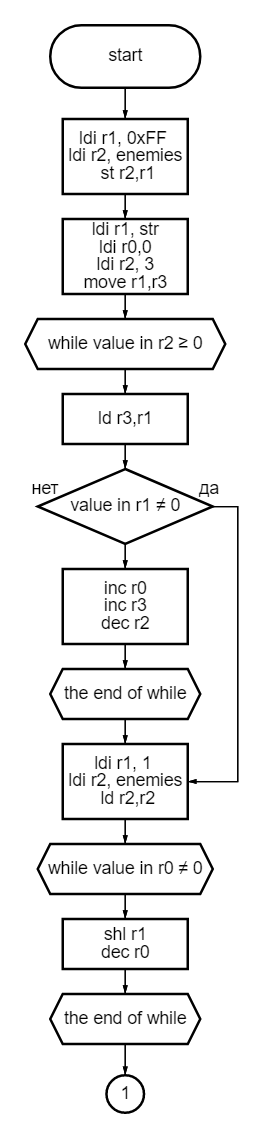


Рисунок \_ - первая часть блок-схемы алгоритма инициализации состояния пришельцев и обработки попадания в них

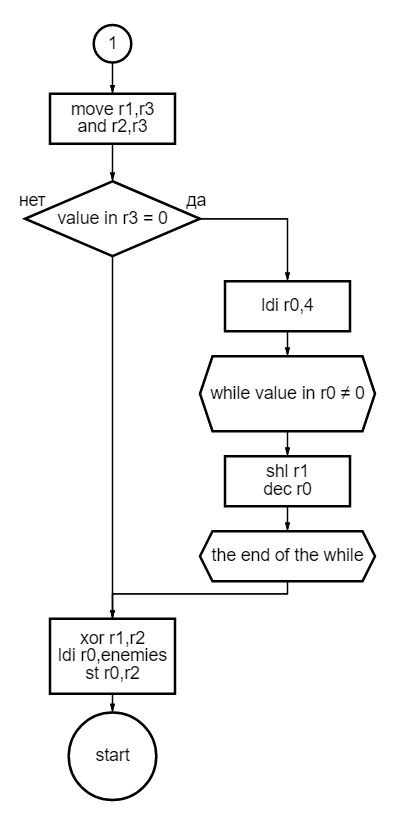


Рисунок \_ - вторая часть блок-схемы алгоритма инициализации состояния пришельцев и обработки попадания в них

Таким образом, разработанная игра – это результат взаимосвязанной работы аппаратной и программной частей проекта, описанных ранее.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы у нас получилось создать игру «Rush space» в жанре Shoot ‘em up (руководство пользователя см. приложение А). Она реализована, используя электронные схемы, в которых мы задействовали процессор CdM-8, выполняющий написанные нами коды. Задачи, поставленные перед нами в начале проекта, были успешно выполнены. Мы смогли реализовать все обозначенные функциональные требования: движения и стрельба игрока и противников, управление с кнопок клавиатуры, возможности выигрыша и проигрыша в игре. Создавая проект, мы получили знания в области создания электросхем, работы с процессором построенном на Гарвардской архитектуре и его программирования, написания проектной документации, а также опыт работы в команде.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Computing platforms: учебник / A.Shafarenko, S.P.Hunt. – 2015.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

**Руководство пользователя**

При открытии игры пользователь видит электросхему, стилизованную под аркадный автомат. На ней расположены две кнопки и клавиатура, с помощью которых игрок может осуществлять управление игрой. На рисунке А.1 представлен скриншот начального экрана игры.

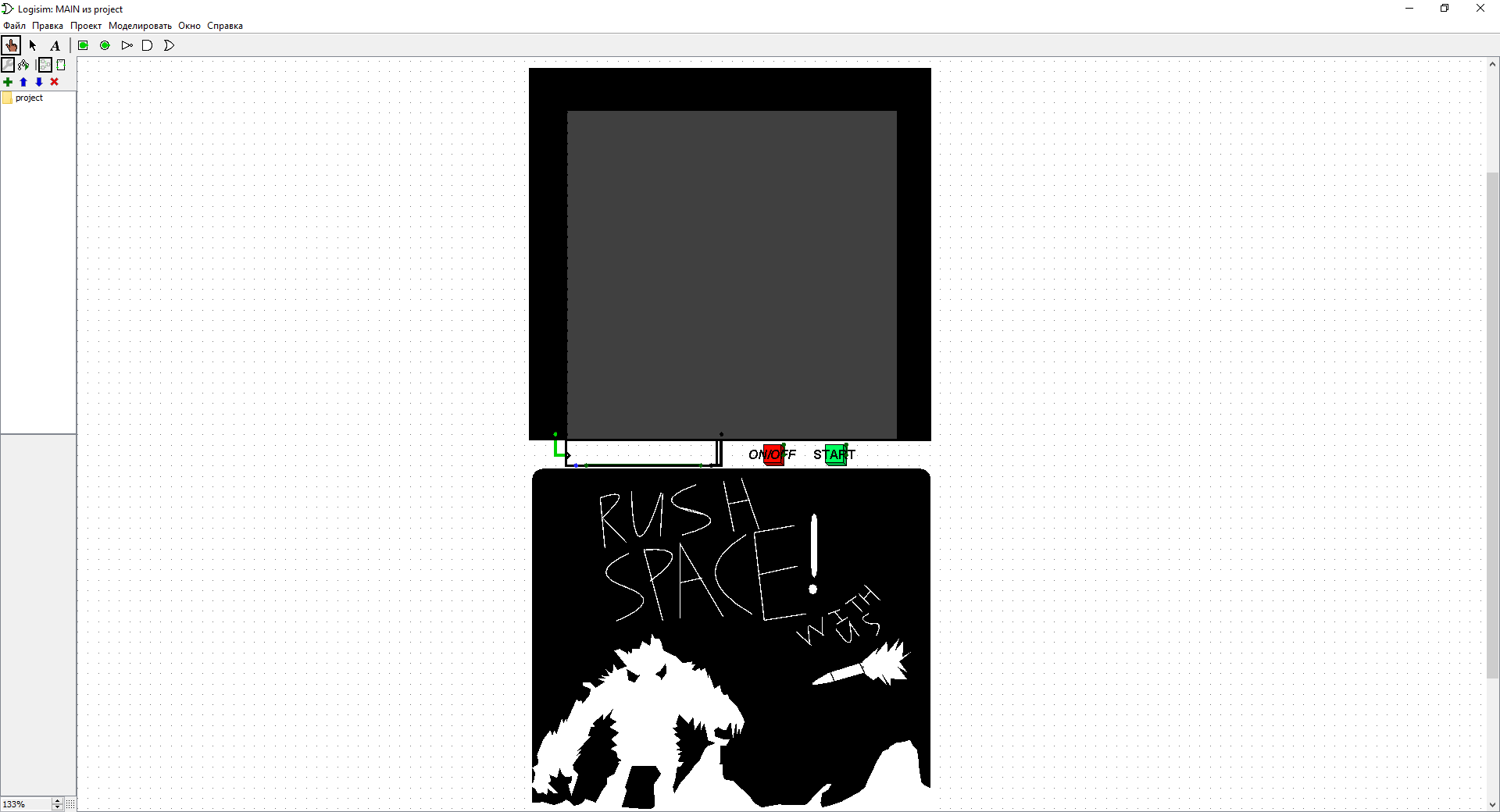


Рисунок А.1 – скриншот начального экрана игры

Для включения игры пользователю нужно нажать кнопку «ON/OFF». После этого игрок увидит на экране сообщение о том, что для начала игрового процесса ему необходимо нажать кнопку «START». На рисунке А.2 представлен скриншот экрана с этим сообщением.

Изображение выглядит как текст, часы

Автоматически созданное описание

Рисунок А.2 – скриншот экрана с просьбой нажать кнопку «START»

При нажатии на кнопку, запускающую игру, на экран выводится поле и начинается игровой процесс. Игрок управляет пушкой, находящейся в нижней части экрана, в горизонтальной плоскости с помощью клавиатуры, а именно клавиш “w”, “a”, “s”, “d”. Также на поле пользователь видит 4 статичные стенки, и 8 монстров, движущихся неуправляемо по горизонтали и вертикали. На рисунке А.3 представлен скриншот игрового поля в момент начала игры.

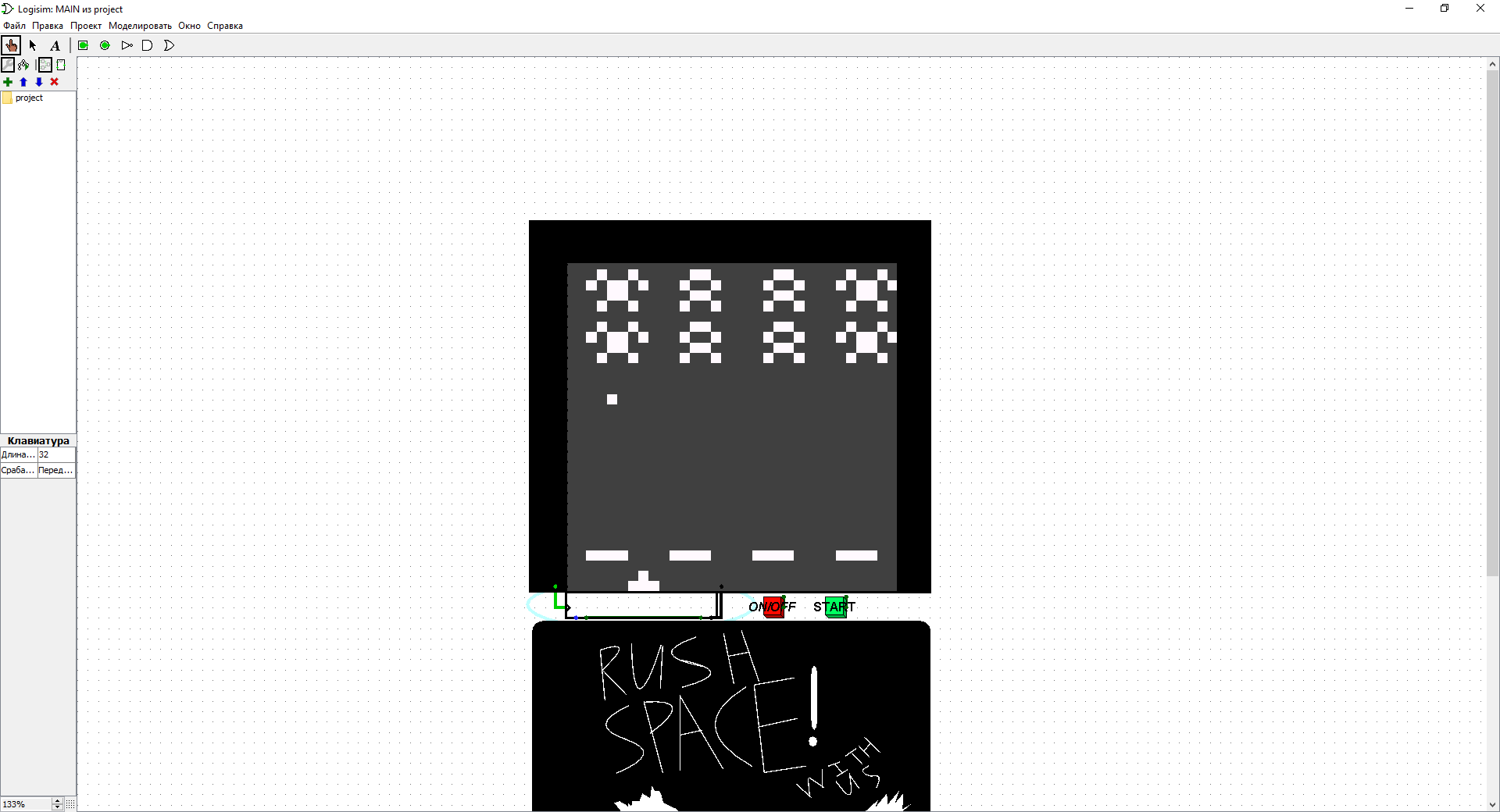


Рисунок А.3 – скриншот игрового поля в момент начала игры

Также пользователь может стрелять, используя клавишу "SPACE” на клавиатуре. При нажатии из пушки вылетает пуля, летящая вертикально вверх до того момента, пока не попадет в пришельца, либо до того, как выйдет за пределы игрового поля. На рисунке А.4 показан скриншот игрового поля в момент выстрела.

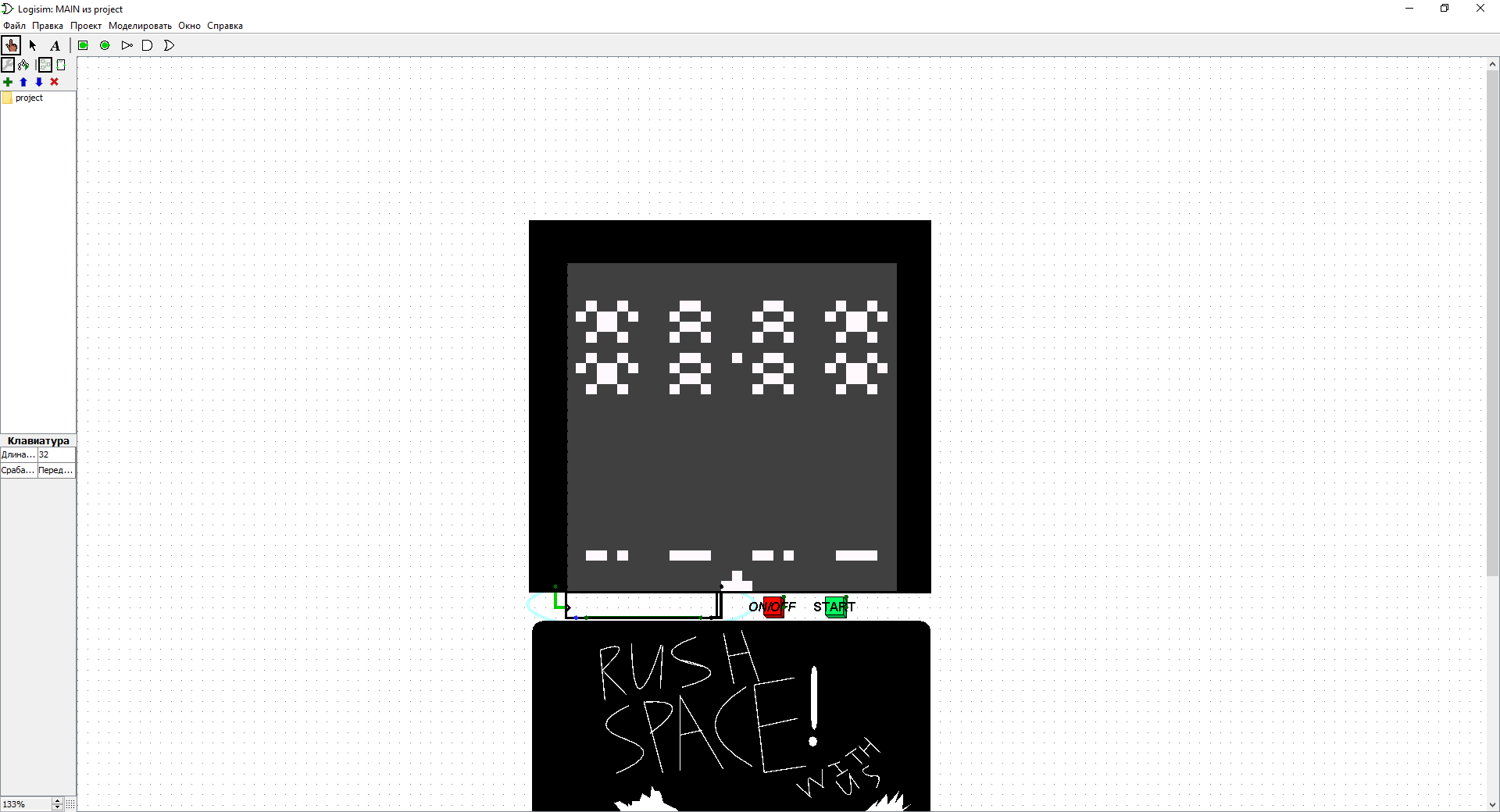


Рисунок А.4 – скриншот игрового поля в момент выстрела

Монстры тоже осуществляют стрельбу. Каждые несколько секунд случайный пришелец выпускает пулю, летящую вертикально вниз до того момента, пока не попадет в игрока, либо в стенку, либо до того, как выйдет за пределы игрового поля. Стенка, защищающая игрока, при попадании в неё пули частично разрушается. Каждая «защита» выдерживает 4 попадания, прежде чем полностью разрушится. Важно заметить, стенка может быть атакована не только инопланетянами, но и самим игроком. На рисунке А.5 представлен скриншот игрового поля с частично разрушенными стенками.

Изображение выглядит как текст, часы, снимок экрана

Автоматически созданное описание

Рисунок А.5 – скриншот игрового поля с частично разрушенными стенками

При попадании пули пришельца в игрока он проигрывает, и на экране появляется сообщение о конце игры. На рисунке А.6 представлен скриншот сообщения о конце игры.

Изображение выглядит как текст, часы, снимок экрана

Автоматически созданное описание

Рисунок А.6 – скриншот экрана с сообщением о конце игры

Игрок может проиграть и при другом условии. Каждые несколько секунд все монстры перемещаются ближе к пушке. Если хотя бы один пришелец сможет выжить и добраться до стенок, то игра будет закончена и на экране появится сообщение о её конце. На рисунке А.7 представлен скриншот игрового поля в состоянии близком к проигрышу.

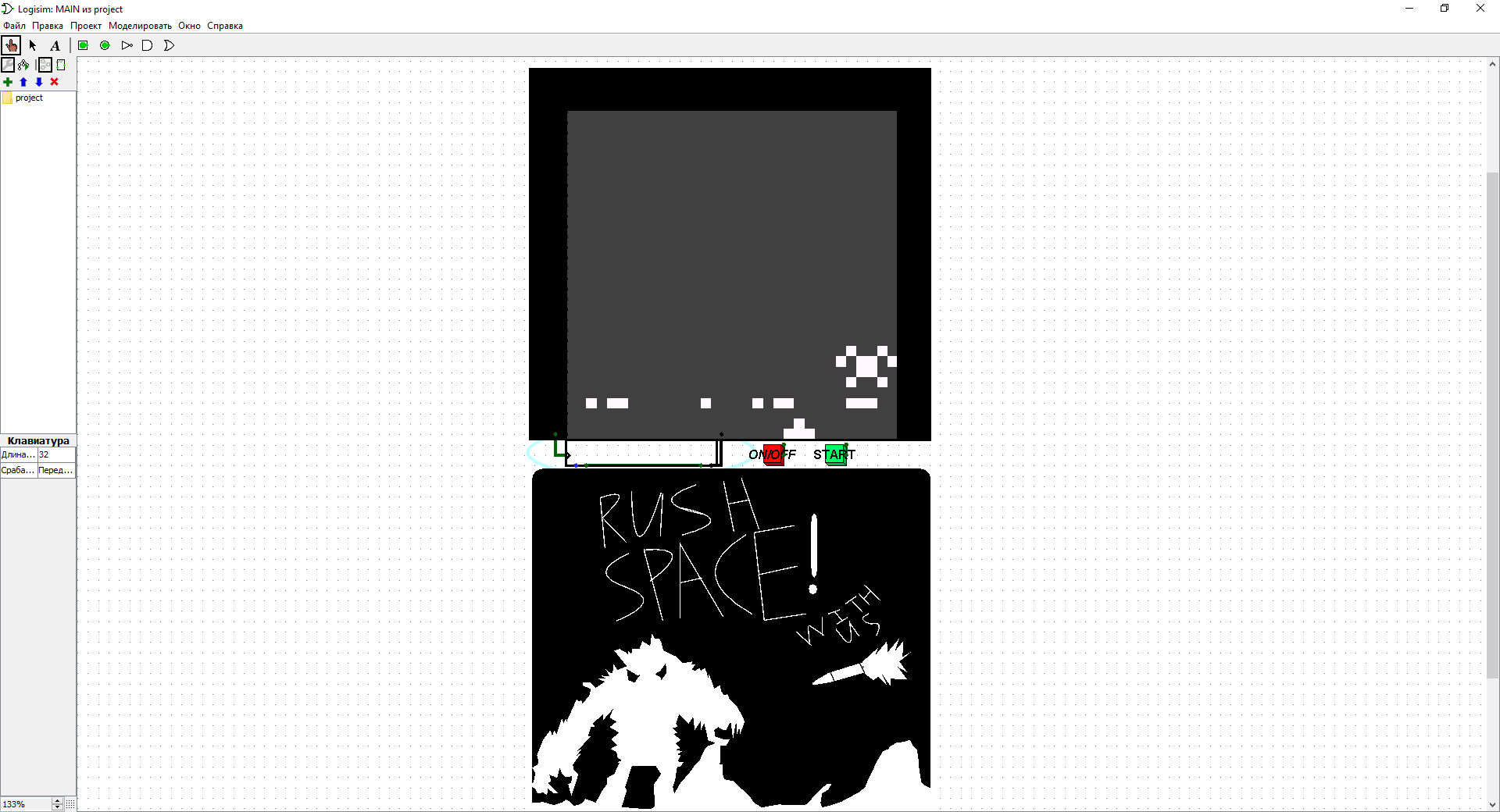


Рисунок А.7 – скриншот игрового поля в состоянии близком к проигрышу

Для победы пользователь должен уничтожить всех монстров. На рисунке А.8 представлен скриншот игрового поля после попадания в большинство инопланетян.

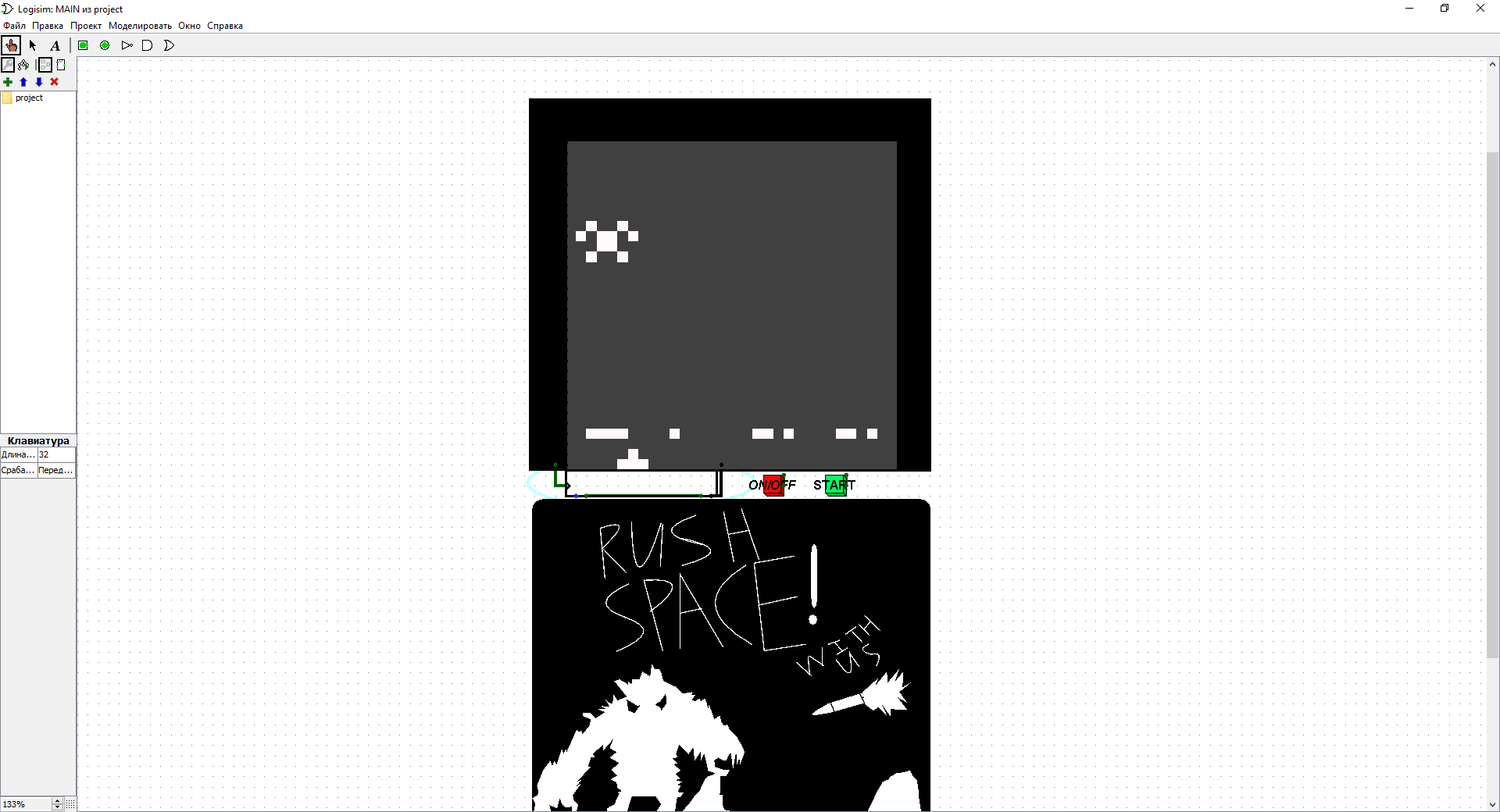


Рисунок А.8 – скриншот игрового поля после уничтожения большинства инопланетян

После уничтожения всех пришельцев на экране появится сообщение о победе игрока. На рисунке А.9 представлен скриншот сообщения о победе.

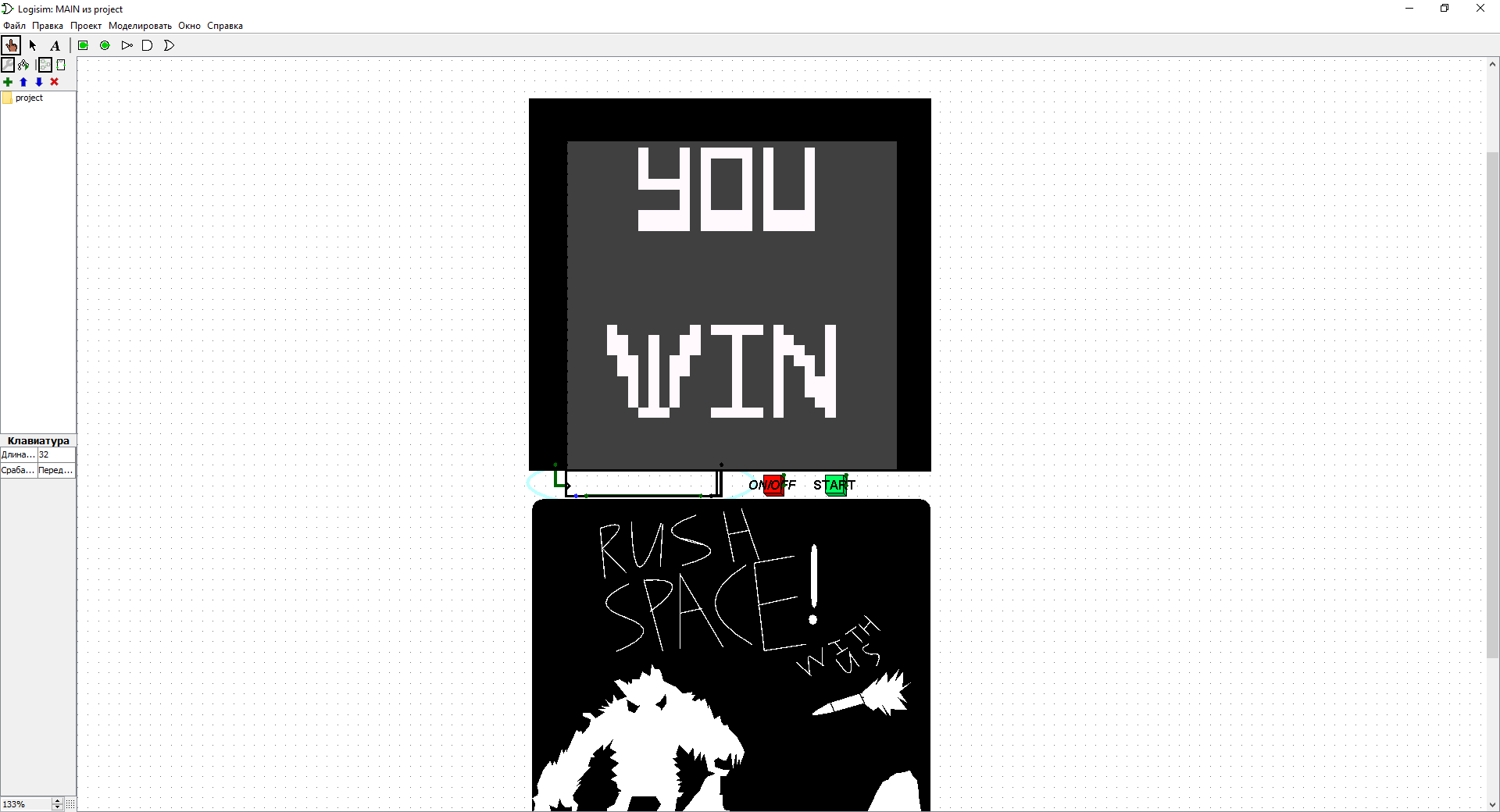


Рисунок А.9 – скриншот сообщения о победе