

Кросс-платформенная игра для обучения программированию Code'n'Play



Выполнил
студент группы 0826-3
Прыгин Владислав

Руководитель: доц. каф. Программной Инженерии
Борисов Н.А.

Powered by Unity

Содержание

- Цель работы
- Описание аналогов
- Преимущества данной программы перед аналогами
- Описание используемых технологий
- Демонстрация работы
- Заключение
- Дальнейшие пути развития

Цель работы

Разработать программу для интеграции в процесс обучения программированию, а также для повышения интереса к программированию у школьников средних и младших классов.

Программа должна содержать следующий функционал

- Выбор уровня для прохождения
- Написание кода для выполнения
- Компиляция кода
- Демонстрация пользователю результата выполнения его кода

Описание аналогов

<http://www.vzmakeh.ru/info/algo/modules/page4.html>

Робот исполнитель



Описание Аналогов

<https://sgimenez.github.io/laby/>

Laby

The screenshot displays the Laby programming environment. At the top, there is a menu bar with "Menu" and "Level". Below it, the title bar reads "Webs and rocks" and a subtitle says "Woh it becomes increasingly complicated!". The main area on the left shows a maze with orange walls, a red robot, a black spider, and a white web. On the right, the "Program:" window contains the following code:

```
1 open Robot
2
3 let ant =
4   right ();
5   forward ();
6   left ();
7   if look () == Web
8   then (
9     left ();
10    forward ();
11    ...
12  )
13 else ...
```

Below the code editor is an "Execute" button and a set of navigation controls: "Back", "Forward", "Rewind", "Play", and "Forward". At the bottom, the "Messages:" window shows the message "I'm ready —".

Преимущества данной программы перед аналогами

- Трёхмерная графика
- Поддержка любых платформ, в том числе мобильных
- Поддержка конкретного языка программирования, а не абстрактных языков

Описание используемых технологии

C#

В первую очередь стоял вопрос «Какой язык программирования использовать?». В качестве используемого языка был выбран C#.

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

Описание используемых технологии

Выбор движка

Далее стоял вопрос о том, какой движок использовать для создания игры на C#. Написать собственный качественный 3D движок не представлялось возможным в силу строгого ограничения во времени. Было потрачено некоторое время на движок Agile2D, с использованием C++ 14, но он оказался не подходящим, в силу отсутствия реализации работы с текстом, и общей бедности своего API. Во время поиска более подходящего инструментария был обнаружен движок Ougine. Общая сложность и времязатратность разработки на нём в итоге привела к выбору Unity3D в качестве основного инструментария.

Описание используемых технологии

Unity 3D

Unity3d это кросс-платформенный движок для разработки 3D игр. Распространяется бесплатно. Предлагает простое к использованию API написанное исходя из принципа «Больше функционала, меньше кода».

Главное преимущество Unity3d это его кросс-платформенность. Реализована она по принципу, что код который пишет разработчик един для любой платформы. Задача разработчика заключается лишь в том чтобы собирать билды под разные платформы. Так для создания игры на Android Unity3d создает apk файл, а для iOS проект для xCode. Более того нет необходимости даже менять управление мышкой на управление с использование тач скрин.

Демонстрация работы

Начало работы



```
//Чтобы найти машину введите  
var car =  
GameObject.FindObjectOfType<PlayerController>();  
/*Используйте функции:  
car.MoveDownCommand(),  
car.MoveUpCommand(),  
car.MoveRightCommand(),  
car.MoveLeftCommand, чтобы добавить  
команды движения, car.StartMove() чтобы  
указать машине когда начать двигаться и  
car.ClearCommands чтобы очистить поле  
команд*/
```

Скомпилировать

Выполнить

Информация об ошибках

or is set to 3D and does not contain packable sprites. Spent 84.80 seconds searching for sprites. Consider setting sprite packing mode to 'Disabled' or 'Enabled for Builds' under Project

Демонстрация работы

Ввод кода и компиляция



The image shows a game engine interface. On the left is a 3D view of a maze environment with a car, a fire, and various objects. On the right is a code editor with the following code:

```
var car =
GameObject.FindObjectOfType<PlayerController>();

car.ClearCommands();

for(int i = 0;i<7;i++){
    car.MoveDownCommand();
}

car.MoveRightCommand();
car.MoveRightCommand();
car.MoveRightCommand();

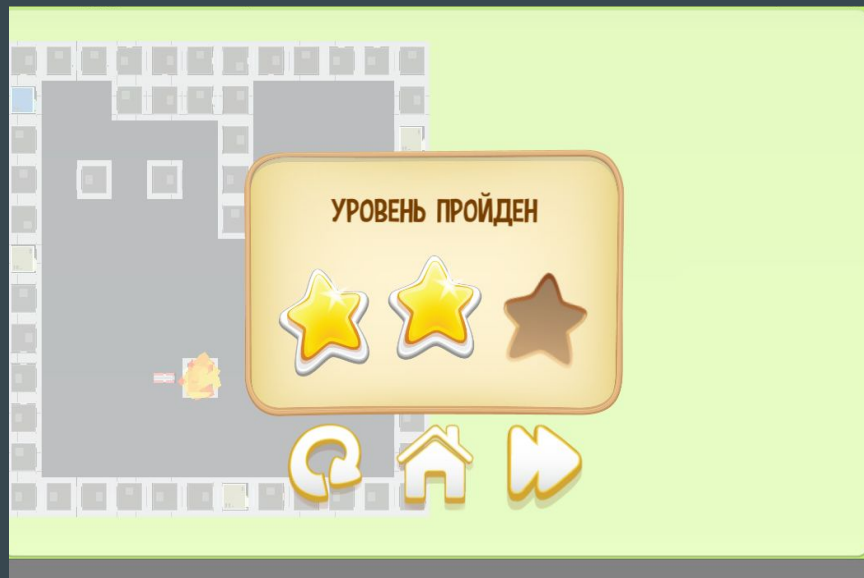
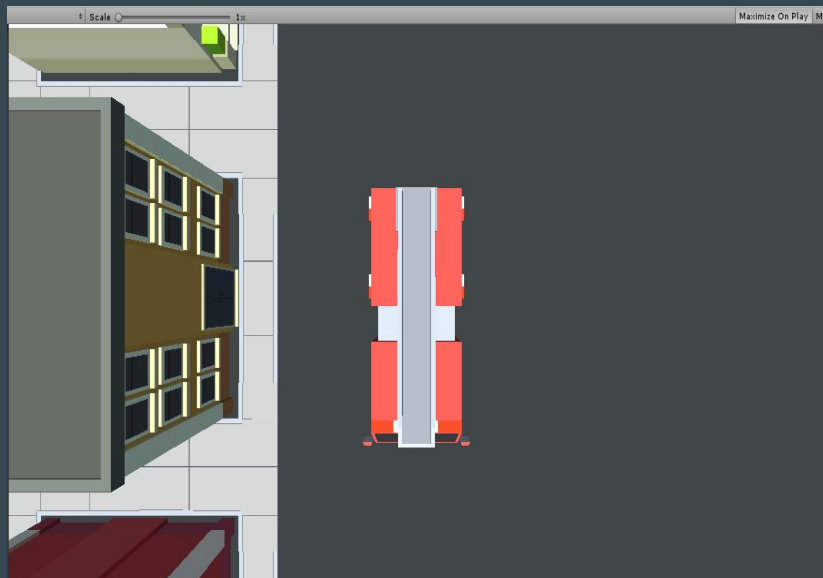
car.StartMove();
```

Below the code editor are two buttons: "Скомпилировать" (Compile) and "Выполнить" (Execute).

ior is set to 3D and does not contain packable sprites. Spent 84.80 seconds searching for sprites. Consider setting sprite packing mode to 'Disabled' or 'Enabled for Builds' under Project

Демонстрация работы

Просмотр выполнения написанного кода и вывод результата



Заключение

- Была написана программа для обучения программированию

Главные особенности

- Кросс-платформенность
- Трёхмерная графика
- Существующий язык программирования
- Популярный и приятный глазу стиль графики: Lowpoly + Material Design

Дальнейшие пути развития

- Загрузка уровней из текстового файла
- Поддержка большего количества языков программирования
- Добавление новых команд для персонажа
- Внутриигровой редактор уровней

Контактная информация

Прыгин Владислав Алексеевич

группа 0826-3

email: prygin.vladislav@gmail.com