Рисунок 2 — Отчеты

Заключение. Разработанный программный продукт создан в соответствии с требованиями технического задания рассматриваемой базы исследования. Выполнены все поставленные задачи, позволяющие максимально оптимизировать работу сотрудников технического отдела.

Таким образом, создание модуля автоматизации деятельности технического отдела на базе платформы 1С: Предприятие позволяет:

- повысить производительность труда работников;
- эффективно распределять рабочего времени;
- уменьшить временные и денежные затраты.

Разработанная система является достаточно эффективной, соответствует всем заявленным требованиям, не требует больших материальных затрат и глубоких познаний пользователя.

Список цитируемых источников

1. *Титоренко, Г. А.* Автоматизированные информационные технологии в экономике / Г. А. Титоренко. — М. : Компьютер, ЮНИТИ, 2006. — 400 с.

УДК 004.021

А. В. Лыско, Г. М. Раковцы
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

ПРИМЕНЕНИЕ АЛГОРИТМА ОБРАТНОГО БЭКТРЕКИНГА ДЛЯ ГЕНЕРАЦИИ ЛАБИРИНТОВ

Введение. Лабиринт — структура, обычно в двухмерном или трёхмерном пространстве, состоящая из запутанных путей, ведущих к выходу или в тупик. Существует несколько алгоритмов генерации лабиринтов:

- алгоритм двоичного дерева;
- алгоритм «Sidewinder»;
- алгоритм Эйлера;
- алгоритм Олдоса-Бродера;
- алгоритм Уилсона и другие [1][2].

Все эти алгоритмы имеют разный принцип действия и строят разные лабиринты. Их объединяет только то, что они строят ортогональные двухмерные лабиринты. В данной статье рассмотрим алгоритм обратного бэктрекинга на примере игры «Лабиринт». Для ее реализации используем среду программирования C++ Builder.

Основная часть. Игра начинается с главного меню, где есть три кнопки: «Играть», «Рейтинг» и «Выход» (рисунок 1). Нажав «Играть» мы переходим на другую форму, где можно выбрать уровень сложности лабиринта (рисунок 2). Предусмотрены три уровня сложности: Простой, Нормальный и Сложный. Они различаются размером генерируемого лабиринта.

Рисунок 1 — Главное меню

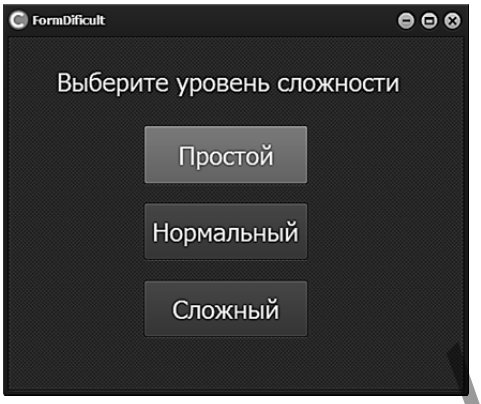

Рисунок 2 — Уровни сложности лабиринтов

После выбора уровня сложности происходит процесс генерации. Алгоритм обратного бэктрекинга, при использовании для генерации лабиринтов, строит двухмерные, односвязные, ортогональные лабиринты. Двухмерные, означат, что движение по лабиринту осуществляется по двум осям. Понятие односвязности в контексте лабиринта означает, что лабиринт не содержит замкнутых маршрутов. Ортогональным называется лабиринт, стены которого расположены строго перпендикулярно друг другу [3].

Генерация начинается с таблицы m на n пустых клеток и происходит по следующему алгоритму:

1. Сделаем начальную клетку текущей и отметим ее как посещенную.
2. Пока есть не посещенные клетки:
   2.1. Если текущая клетка имеет не посещенных соседей.
      2.1.1. Занесем текущую клетку в стек.
      2.1.2. Выберем случайную клетку из соседних.
      2.1.3. Уберем стенку между текущей клеткой в выбранной.
      2.1.4. Сделаем выбранную клетку текущей и отметим, как посещенную.
   2.2. Иначе, если стек не пуст.
      2.2.1. Извлечем клетку из стека.
      2.2.2. Сделаем ее текущей.

Недостаток данного алгоритма заключается в том, что он не предусматривает наличие входа и выхода. Поэтому, для простоты, можно сказать, что вход будет находиться в левом верхнем угле, а выход — в правом нижнем. Результат работы данного алгоритма представлен ниже на рисунке 3.

На рисунке 3 красный круг (персонаж, которым мы проходим лабиринт) обозначает начало лабиринта, а зеленый квадрат — его конец.

Также в программе реализованы следующие функции: управление движением персонажа при помощи кнопок W, A, S, D (вверх, влево, вниз, вправо), взаимодействие персонажа со стенками лабиринта, то есть он не может проходить через них. Процесс прохождения представлен на рисунке 4.

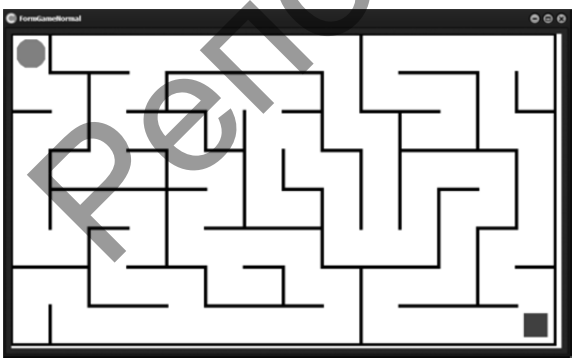

Рисунок 3 — Пример работы алгоритма

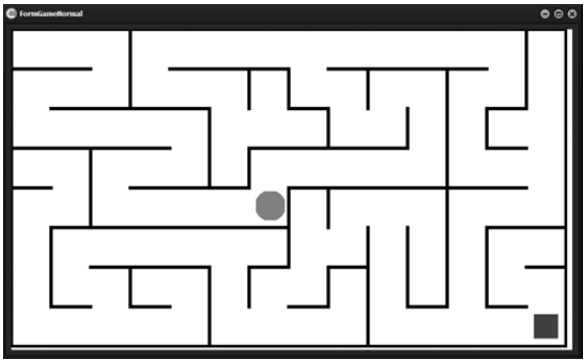

Рисунок 4 — Процесс прохождения

После прохождения появляется форма сохранения результата. Здесь показаны время прохождения, уровень сложности, и предлагается вписать имя, под которым будет сохранен данный результат. При нажатии на кнопку «Сохранить» результат запишется в файл. Если же нажать «Не сохранять», результат удалится. Окно сохранения результат представлено на рисунке 5.

Также есть таблица с рейтингом, в которую записывается время прохождения лабиринта, его сложность и имя игрока, прошедшего лабиринт (рисунок 6). Участники сортируются по времени прохождения лабиринта и их данные хранятся в файле.

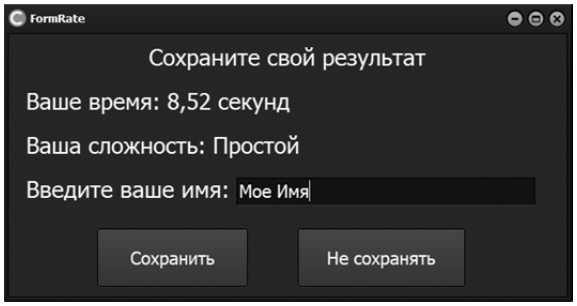

Рисунок 5 — Сохранение результата

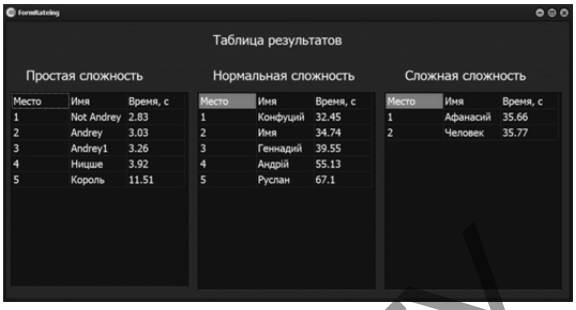

Рисунок 6 — Рейтинг прошедших лабиринт

Заключение. Алгоритм обратного бэктрекинга является оптимальным для создания лабиринтов. Данный алгоритм используется не только для генерации лабиринтов, но также распространен при решении задач, в которых требуется возвращение к предыдущему шагу и может быть реализован на разных языках программирования.

Список цитируемых источников

1. Классические алгоритмы генерации лабиринтов. Часть 1 : вступление [Электронный ресурс]. — Режим доступа : https://habr.com/ru/post/320140/. — Дата доступа : 27.04.2021.

2. Классические алгоритмы генерации лабиринтов. Часть 2: погружение в случайность [Электронный ресурс]. — Режим доступа : https://habr.com/ru/post/321210/. — Дата доступа : 27.04.2021.

3. Лабиринты: классификация, генерирование, поиск решений [Электронный ресурс]. — Режим доступа : https://habr.com /ru/post/445378/. — Дата доступа : 27.04.2021.

УДК 004.94

П. П. Люцко, О. Д. Кравчук

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

СОЗДАНИЕ МОДЕЛЕЙ С ИСПОЛЬЛЬЗОВАНИЕМ UNREAL ENGINE

Введение. Компьютерные технологии развиваются с огромной скоростью. В современном мире компьютеры помогают в работе, используются в сфере образования, экономики, игр, досуга, позволяют людям легко связываться друг с другом.

Объектно-ориентированные языки программирования пользуются в последнее время большой популярностью среди программистов, так как они позволяют использовать преимущества объектно-ориентированного подхода не только на этапах проектирования и конструирования программных систем, но и на этапах их реализации, тестирования и сопровождения.

Целью исследования является организация создания модели, импорт в Unreal Engine 4 и последующее использование ее для создания игр.

Объектом исследования является процесс создания моделей.

В качестве инструментов исследования выбран игровой движок Unreal Engine. Unreal Engine — игровой движок, инструмент, разрабатываемый и поддерживаемый компанией Epic Games. Unreal Engine — это набор инструментов для разработки игр, имеющий широкие возможности: от создания двухмерных игр на мобильные до AAA-проектов для консолей. Этот движок использовался при разработке таких игр, как S.T.A.L.K.E.R. 2 и Tekken 7. Blueprints Visual Scripting в Unreal Engine позволяет упрощенно прописывать всевозможные действия в проектах, которые создает пользователь.

Основная часть. В качестве темы разработки выбрана разработка проекта 3D-мира лесного пространства, и в качестве модели выбрана модель «кролика».

Создание модели персонажа проходило в несколько этапов:

1. Создание модели в ZBrush. ZBrush — программа для 3D моделирования, созданная компанией Pixologic. Отличительной особенностью данного ПО является имитация процесса «лепки» трёхмерной скульптуры, усиленного движком трёхмерного рендеринга в реальном времени, что существенно упрощает процедуру создания требуемого трёхмерного объекта. Модель в ZBrush представлена на рисунке 1.