Список цитируемых источников

1. *Ричардсон, К.* Микросервисы. Паттерны разработки и рефакторинга / К. Ричардсон. — СПб. : Питер, 2019. — 544 с.
2. *Симан, М.* Внедрение зависимостей на платформе .NET / М. Симан, Стивен Ван Дерсен. — СПб. : Питер, 2021. — 608 с.

УДК 004.9

Р. Д. Дедулько, О. Д. Кравчук
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

РАЗРАБОТКА ИГРЫ «СОКОБАН» С ПРИМЕНЕНИЕМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

Введение. Игровая индустрия играет важную роль в различных аспектах жизни и общества, таких как:

1. Развлечение и отдых. Самая очевидная роль игровой индустрии — предоставление различных видов игр для развлечения и отдыха. Игры могут быть использованы для расслабления, увлечения, улучшения моторики и координации движений, получения культурных знаний и т. д.

2. Развитие технологий. Игровые разработки стоят у истоков развития многих новых технологий. Например, в процессе создания игр были разработаны уникальные алгоритмы, графические движки, системы управления и распределения ресурсов.

3. Обучение. Различные жанры игр также могут быть использованы как эффективное средство обучения. Обучающие игры предоставляют пользователям целый ряд преимуществ, таких как облегчение запоминания информации, улучшение реакции, увеличение мотивации и интереса к учебному процессу.

4. Создание новых профессий. Игровая индустрия создала множество новых профессий, таких как разработчик игр, дизайнер персонажей, гейм-мейкер, игровой архитектор, тестировщик игр и т. д.

5. Экономическое развитие. Игровая индустрия является одной из самых быстрорастущих в мире. Она создает рабочие места, привлекает инвестиции, вносит вклад в общее экономическое развитие и содействует улучшению условий жизни многих людей.

Таким образом, игровая индустрия играет важную роль в различных сферах жизни и общества и продолжает расти и развиваться, предоставляя новые возможности и выгоды, как для потребителей, так и для разработчиков и инвесторов.

Основная часть. Цель исследования: разработка игры «Сокобан» с использованием современных технологий программирования. Объект исследования: игровой процесс игры «Сокобан».

Сокобан — это головоломка, в которой надо переставлять ящики на определенные места на игровом поле. Эта игра очень популярна во всем мире, и ее играют люди всех возрастов.

Процесс игры начинается с того, что игрок появляется на игровом поле вместе с тремя или четырьмя ящиками. Игрок может передвигаться только вверх, вниз, вправо и влево, и каждый раз, когда игрок двигается, все ящики на игровом поле тоже двигаются вместе с ним.

Цель игры состоит в том, чтобы передвинуть все ящики на определенные места на игровом поле. Эти места обычно помечены различными символами или цветами, чтобы обозначить, куда нужно поместить каждый ящик. Игрок должен продвигать ящики по-игровому полю так, чтобы они не заблокировались в углах или у краев поля [1].

Каждый уровень игры имеет свой уникальный дизайн, что делает игру более интересной и вызывающей. Уровни становятся все сложнее и сложнее по мере продвижения игрока в игре, и требуют более тщательного планирования и размышления, чтобы достичь цели.

В целом, процесс игры «Сокобан» представляет собой игру рефлексов и логического мышления, требующую от игрока многократных попыток и ошибок, чтобы достичь цели и пройти все уровни игры.

Диаграмма классов является одним из главных инструментов для моделирования различных аспектов объектно-ориентированной системы. Ее назначение заключается в создании графического представления классов и их связей в системе, что позволяет лучше понимать структуру и архитектуру системы. На рисунке 1 представлена диаграмма классов разрабатываемого приложения.

В ClassLevelMatrix имеются значения ячеек матриц уровня игрок, ящик, стена, целевая точка, ящик на целевой точке, пустая клетка, игрок на цели. Так же имеются методы loadfromfile и savetofile наследуемые из интерфейса InterfaceWorkingWithFile для работы с файлами.

В интерфейсе InterfaceMoving описаны прототипы функций для перемещения по лабиринту moveUp(), moveDown(), moveLeft(), moveRight().

В интерфейсе InterfaceGameVictory описаны прототипы функций VictoryValidate() для проверки условий победы и Victory() для обработки победы.

ClassScoreRecord содержит три поля результат, имя игрока при входе, пройденный уровень.

ClassHichStoreTable содержит поле «таблица с результатами» и методы LoadData() для отображения

таблицы рекордов DataGridView и SaveData() для сохранения результата в таблицу рекордов. Класс ClassCoordinates содержит координаты игрока в матрице.

ClassGameController множественно наследует ClassLevelMatrix, InterfaceMoving и InterfaceGameVictory. Имеется функция поиска координат игрока GetPlayerCoordinate() в котором объекту класса ClassCoordinates присваиваются координаты игрока. В классе описываются методы moveUp(), moveDown(), moveLeft(), moveRight() для перемещения по лабиринту.

Игра содержит 15 уровней. Время прохождения каждого уровня фиксируется и сохраняется в таблицу рекордов.

На рисунке 2 представлен интерфейс приложения.

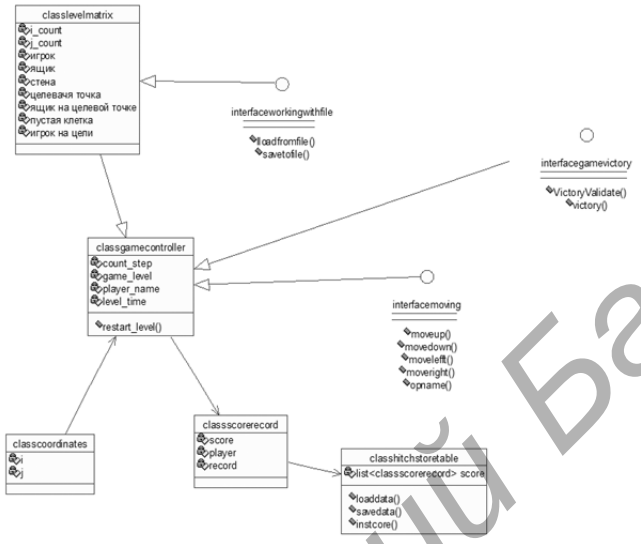

Рисунок 1 — Диаграмма классов разрабатываемого приложения

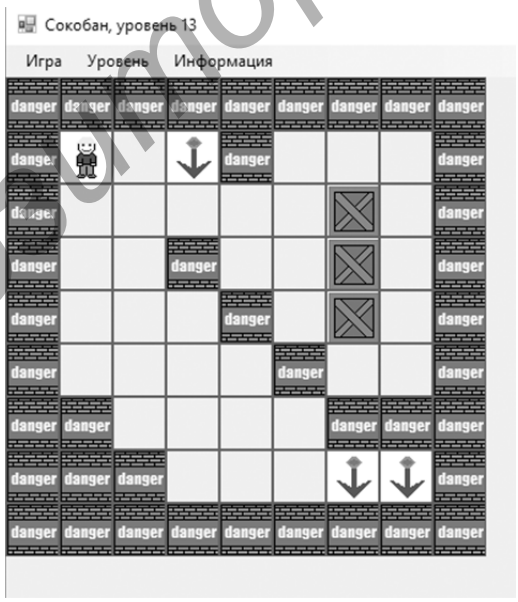

Рисунок 2 — Интерфейс разрабатываемого приложения

Заключение. Игра «Сокобан» была успешно разработана с использованием объектно-ориентированного подхода, что позволило достичь следующих результатов:

Организация кода игры в виде классов позволила упростить процесс разработки, тестирования и сопровождения приложения. Реализованный интерфейс позволяет игрокам комфортно управлять главным героем и перемещать элементы игрового поля.

Разработана полная документация приложения для облегчения сопровождения и развития программы.

Таким образом, разработка игры «Сокобан» с применением объектно-ориентированного подхода успешно завершилась и привела к созданию удобного, быстрого и функционального приложения.

Список цитируемых источников

1. *Боровский, А. Н.* Практическое программирование на С++./ А. Н. Боровский. — СПб. : БВХ-Петербург, 2012. — 496 с.

УДК 004.896

А. И. Калько, К. Ю. Матусевич, М. В. Прокопович
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА КАЧЕСТВА ВОЗДУХА С ИСПОЛЬЗОВАНИЕМ ARDUINO

Введение. Пользователь современного компьютера не задумывается о функционировании отдельных частей ПК. Он просто запускает нужные программы и работает с ними. Точно так же и Arduino позволяет пользователю сосредоточиться на разработке проектов, а не на изучении устройства и принципов функционирования отдельных элементов. Нет надобности и в создании законченных плат и модулей. Разработчик может использовать готовые платы расширения или просто напрямую подключить к Arduino необходимые элементы. Все остальные усилия будут направлены на разработку и отладку управляющей программы на языке высокого уровня. В итоге доступ к разработке микропроцессорных устройств получили не только профессионалы, но и просто любители что-то сделать своими руками. Наличие готовых модулей и библиотек программ позволяет непрофессионалам в электронике создавать готовые работающие устройства для решения своих задач. А варианты использования Arduino ограничены только возможностями микроконтроллера и имеющегося варианта платы, ну и, конечно, фантазией разработчика [1].

Основная часть. В данном исследовании основной задачей ставилась разработка системы, которая анализирует воздух в помещении и отправляет данные в мессенджере Телеграм. В состав системы входят следующие компоненты:

– микроконтроллер Arduino Uno R3;
– датчик влажности и температуры DHT11;
– датчик давления и температуры BMP180;
– Wi-Fi модуль ESP8266 12E для подключения устройства и сети интернет.

Главным компонентом для реализации поставленной задачи является плата Arduino Uno. Основной элемент платы — микроконтроллер Atmel. На большинстве плат Arduino, включая Arduino Uno, установлен микроконтроллер ATmega. На плате Arduino Uno, используемой в данном проекте, установлен микроконтроллер ATmega 328.

Программа, написанная в среде Arduino, носит название скетч. Скетч пишется в текстовом редакторе, который имеет цветовую подсветку создаваемого программного кода. Во время сохранения и экспорта проекта в области сообщений появляются пояснения и информация об ошибках. Окно вывода текста показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины. Разрабатываемым скетчам дополнительная функциональность может быть добавлена с помощью библиотек, представляющих собой специальным образом оформленный программный код, реализующий некоторый функционал, который можно подключить к создаваемому проекту. Специализированных библиотек существует множество [2].

Вид собранного устройства представлен на рисунке 1. После подключения питания и беспроводного интернета, пользователь начнет получать сообщения с данными, которые снимает микроконтроллер Arduino. Выбор был сделан в пользу данного микроконтроллера, так как устройство должно производить некоторые вычисления и проверки, а также передавать и принимать данные с телефона и других устройств [3]. Данные отправляют при первом подключении, а потом с интервалом в десять минут. Такой интервал обусловлен ненадобностью «спама» со стороны устройства в сторону пользователя, но в тоже время поможет оперативно оповестить пользователя об изменении состояния воздуха. В последующем планируется сделать этот интервал настраиваемым, с возможностью включать/отключать отправку сообщений в определенное время.

Для сбора данных о качестве воздуха в помещении используются два датчика — DHT11 и BMP180. Датчик DHT11 позволяет измерить температуру и влажность воздуха. Датчик BMP180 измеряет атмосферное давление и температуру. Оба датчика подключены к плате Arduino Uno через аналоговые порты.