

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	2
1 ФОРМУЛИРОВКА ЗАДАНИЯ	4
2 ОПИСАНИЕ ПРОГРАММЫ	5
3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	8
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

Курсовая работа посвящена разработке классической игры “Змейка” с помощью объектно-ориентированных средств языка программирования C++ и с применением графической библиотеки SDL 2.0.

“Змейка” — компьютерная игра, возникшая в середине или в конце 1970-х. В этой игре игрок управляет длинным, тонким существом, напоминающим змею, которое ползает по плоскости (как правило, ограниченной стенками), собирая еду (или другие предметы), избегая столкновения с собственным хвостом и краями игрового поля. В некоторых вариантах на поле присутствуют дополнительные препятствия. Каждый раз, когда змея съедает кусок пищи, она становится длиннее, что постепенно усложняет игру. Игрок управляет направлением движения головы змеи (обычно 4 направления: вверх, вниз, влево, вправо), а хвост змеи движется следом. Игрок не может остановить движение змеи.

Первой игрой этого плана был игровой автомат Hustle, выпущенный фирмой Gremlin в 1977 году. Одна из первых реализаций на домашнем компьютере была сделана в 1979 году на компьютере TRS-80 немцем F. Seger. Вскоре после нее вышла версия для Commodore VIC-20, под названием Worms. Микрокомпьютерная версия игры Hustle была выпущена Milton Bradley для компьютера TI-99/4A в 1980 году. Версии игры существовали и на многих советских компьютерах, например — на Радио 86РК.

В число известных вариантов Питона входит игра Nibbles, некоторое время входившая в комплект MS-DOS.

Библиотека SDL — это свободная кроссплатформенная мультимедийная библиотека, реализующая единый программный интерфейс к графической подсистеме, звуковым устройствам и средствам

ввода для широкого спектра платформ. Данная библиотека активно используется при написании кроссплатформенных мультимедийных программ (в основном игр). SDL API доступны для многих языков программирования, в том числе и для C++. Основная часть SDL содержит базовый, весьма ограниченный спектр возможностей. Дополнительную функциональность обеспечивают библиотеки расширений, которые обычно входят в поставку SDL.

1 ФОРМУЛИРОВКА ЗАДАНИЯ

В курсовой работе требуется разработать и написать классическую игру “Змейка”. Для это необходимо выполнить следующие задачи:

- Продумать и создать несколько классов, описывающих объекты, участвующие в процессе работы программы.
- Познакомиться с библиотекой SDL 2.0 и ее функциями.
- Создать приложение на основе объектно-ориентированного программирования на языке C++.

При выполнении курсовой работы следует учитывать следующие рекомендации:

- Программа при выполнении должна иметь дружелюбный интерфейс, включающий заставку, меню и удобные средства для управления программой.
- Заставка должна выводиться на экран при запуске программы и давать представление об игре.
- Управление в программе осуществляется с помощью мыши и клавиатуры.
- Необходимо обеспечить достаточную скорость перерисовки динамически перемещаемых элементов.
- Выполнение курсовой работы должно быть основано на принципах объектно-ориентированного программирования. В работе должна быть построена и обоснована иерархия классов, описывающих предметную область конкретной задачи.
- Программа должна являться графическим приложением для Windows XP (поддержка гарантируется: Windows XP, Windows 7,8,10), написанным на языке C++ с использованием библиотеки SDL 2.0.

2 ОПИСАНИЕ ПРОГРАММЫ

Программа написана на языке высокого уровня C++ с использованием библиотеки SDL 2.0.

В представленной программе используется два файла: главный файл “main.cpp” и подключенные к нему файлы “const.cpp”, “record.cpp”, “game_classes.cpp”, “window_classes.cpp”. Для файлов “const.cpp”, “record.cpp”, “game_classes.cpp”, “window_classes.cpp” есть заголовочные файлы “const.h”, “record.h”, “game_classes.h”, “window_classes.h”.

В файле “main.cpp” имеются следующие функции:

- draw – используется для отображения (прорисовки) на экране всех элементов окна приложения: пунктов меню, фона, игровых объектов (змейки и еды).
- game_timer_callback – функция, вызываемая таймером во время игры. Эта функция отвечает за перемещение змейки по игровому полю.
- main – основная функция, ответственная за вызовы функций из других файлов, а также за ожидание событий.

В файлах “window_classes.h” и “window_classes.cpp” содержатся объявления и определения используемых в программе классов, которые в основном отвечают за элементы окна приложения:

- Screen – класс, отвечающий за фоновое изображение окна приложения.
- Menu – класс, отвечающий за пункты меню. Является родительским классом для классов Menu_Start, Menu_Info, Menu_End.

- Menu_Start – дочерний класс класса Menu, отвечает за запуск игры (при кликании по объекту данного класса левой кнопкой мышки).
- Menu_Info – дочерний класс класса Menu, отвечает за вывод на экран приложения справочной информации об игре (при кликании по объекту данного класса левой кнопкой мышки).
- Menu_Exit – дочерний класс класса Menu, отвечает за выход из приложения (при кликании по объекту данного класса левой кнопкой мышки).
- Input_Window – класс, отвечающий за окно ввода, в которое пользователь может ввести свое имя латинскими символами.

В файлах “game_classes.h” и “game_classes.cpp” содержатся объявления и определения классов, которые в основном отвечают за игровой процесс:

- Snake – класс, отвечающий за объект “змейку”. В данном классе определены все необходимые атрибуты и методы для управления “змейкой”, за которую играет пользователь приложения.
- Food – класс, отвечающий за объект “еда”. Объект данного класса появляется на экране в случайно определенных местах и служит “пищей” для “змейки”: когда “змейка” встречается с “едой”, длина змейки увеличивается, а “еда” появляется в новом случайном месте.
- Game_Field – класс, отвечающий за объект “игровое поле”. В игре при каждом старте новой игры генерируется новое игровое поле разных размеров (размеры каждый раз определяются случайным образом).

В файлах “const.h” и “const.cpp” содержатся объявления и определения глобальных постоянных, используемых в игре.

3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При запуске игры появляется окно, показанное на рис. 3.1.



Рис. 3.1. Начальное окно приложения “Змейка”

Пользователь может ввести свое имя латинскими символами (рис. 3.2). Если пользователь не введет свое имя, то по умолчанию ему будет присвоено имя “anonymous”.

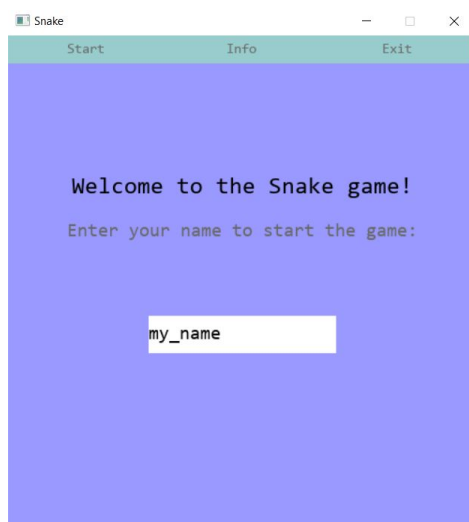


Рис. 3.2. Ввод имени игрока

Чтобы начать игру, пользователь должен кликнуть левой кнопкой мышки по пункту меню “Start” (рис. 3.3).

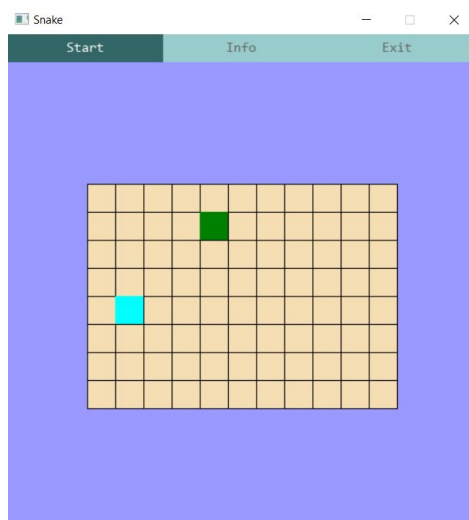


Рис. 3.3. Окно приложения “Змейка” после старта игры

В окошке при этом появляются два квадратика: голубой и зеленый. Голубой квадратик представляет “змейку”, а зеленый – “еду”. Чтобы изменить направление движения “змейки”, нужно использовать клавиши “Вверх”, “Вниз”, “Влево”, “Вправо” на клавиатуре.

Когда игроку удастся попасть “змейкой” на “еду”, длина змейки увеличится на один квадратик. При каждом увеличении длины “змейки” немного меняется и ее цвет. На рис. 3.4 показано изображение окошка приложения с “выросшей змейкой”.

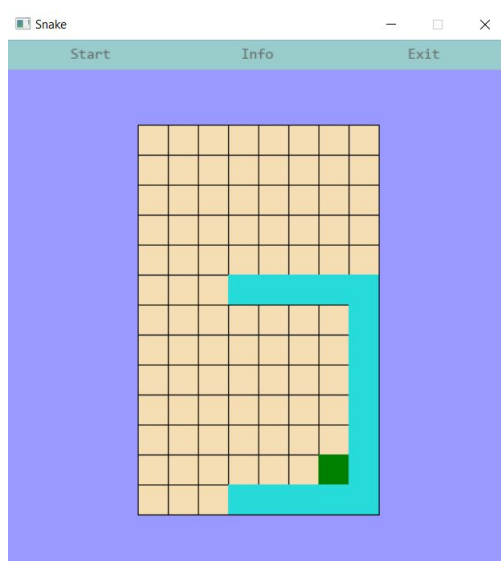


Рис. 3.4. Окно приложения “Змейка” во время игры

Если игрок не успевает повернуть “змейку” в нужном направлении и “змейка” врежется в “стенку” (границу окошка), игра прекращается (рис. 3.5). При этом на экран выводится список из пяти лучших результатов в игре. Черным цветом выделяется результат только что завершившейся игры. После этого игрок может заново запустить игру, кликнув по пункту меню “Start”.



Рис. 3.5. Окно приложения “Змейка” при поражении игрока

Если кликнуть левой кнопкой мышки по пункту меню “Info”, то на экране появится справочная информация по игре (рис. 3.6).

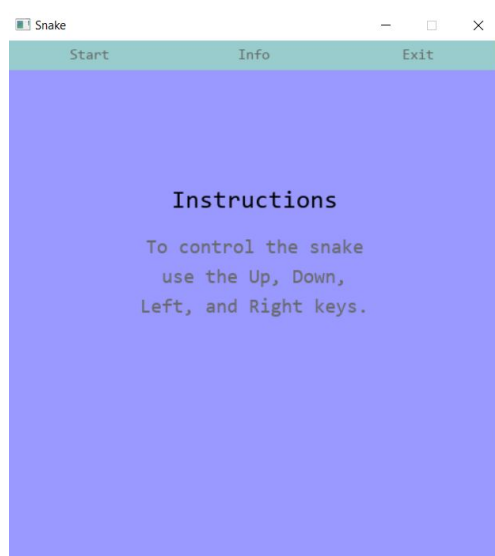


Рис. 3.6. Окно со справочной информацией

Если кликнуть левой кнопкой мышки по пункту меню “Exit”, то приложение закроется.

ЗАКЛЮЧЕНИЕ

В ходе курсовой работы была разработана программа, реализующая классическую игру “Змейка”. Все правила игры соблюдаются и соответствуют поставленной задаче.

В процессе работы над программой была построена иерархия классов, произошло знакомство с библиотекой SDL 2.0 и ее функциями. Результатом курсовой работы стало полноценное, самостоятельно выполненное приложение на базе объектно-ориентированного программирования, с использованием библиотеки SDL 2.0. Все задачи работы выполнены, цель достигнута.

В качестве возможных улучшений игры предлагаем следующие варианты:

- реализовать увеличение скорости перемещения змейки по мере увеличения ее длины;
- сохранять и выводить на экран по требованию игрока список рекордов (например, десять наибольших достигнутых длин змейки и дата, когда это произошло);
- разнообразить графику, например, прорисовать разные части змейки разным образом (голова, хвост).

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Лафоре Р. Объектно-ориентированное программирование в С++ / СПб.: Питер, 2004.
2. Страуструп Б. Язык программирования С++ / М.: Бином, 2004.
3. Доусон М. Изучаем С++ через программирование игр / СПб.: Питер, 2016.
4. Подбельский В.В. Язык Си++ / М.: “Финансы и статистика”, 2003.