УДК 004.424

РАЗРАБОТКА ПАРАЛЛЕЛЬНЫХ СИСТЕМ НА ПРИМЕРЕ СИМУЛЯТОРА ПАДЕНИЯ СНЕГА

Никитин Д. А., Парафиянович Т. А.
Белорусский государственный университет информатики и радиоэлектроники

Аннотация. В статье представлена разработка на основе итерационного процесса составления программ с использованием параллельного программирования в языке C#; информация о методах параллельного выполнения кода: многопоточности (класс Thread) и библиотеке TPL (класс Task); спроектирован и разработан законченный мини-проект симулятор падения снега.

摘要。本文介绍了基于 C# 中使用并行编程的迭代编程过程的开发； 有关并行代码执行方法的信息：多线程（线程类）和 TPL 库（任务类）； 设计并开发了一个完整的迷你项目降雪模拟器。

Введение. При разработке инструментальных программных средств, насыщенных большим функционалом, возникает вопрос о качестве и скорости работы программы. В настоящее время для программ различного вида и назначения критически важно время выполнения запросов, скорость работы системы и отзывчивость пользовательского интерфейса. Внедрение в программное средство параллельности выполнения задач может повысить быстродействие.

Для разработки программного средства выбран язык программирования C#. При работе с оконными приложениями, написанными на данном языке, может возникнуть проблема блокирования пользовательского интерфейса, в случае если вся логика программы выполняется в главном потоке. Для решения этой проблемы применяются механизмы параллельного программирования. Параллельность в программировании – способ организации компьютерных вычислений, при котором для выполнения задачи выделяется свободный ресурс, не затрагивающий выполняющиеся в этот момент задачи [1]. Симулятор падения снега подразумевает генерацию большого количества снежинок, а на программном уровне каждая снежинка представляет собой объект, их нужно обрабатывать и применять к ним различные операции.

Язык программирования C# имеет несколько методов параллельного выполнения кода, в работе рассматриваются следующие: библиотека TPL (класс Task) и многопоточность (класс Thread). Оба метода имеют уникальные особенности, но в данной ситуации рассматривается функционал, позволяющий параллельно выполнять несколько задач. Для разработки программного средства используется метод параллельного выполнения кода – библиотека TPL (класс Task).

Основная часть. На этапе проектирования программного средства определены конкретные операции, необходимые для вынесения в отдельные потоки:

1. Генерация снежинок – задача заключается в создании объектов с определенным временным промежутком;

2. Движение снежинок – каждая снежинка, находящаяся на холсте должна перемещаться с определенной скоростью в определенном направлении;

3. Обновление холста – изменение координат снежинок предполагает постоянное обновление холста, оно заключается в очищении холста и перерисовывании снежинки.

4. Подсчет количества отображенных кадров в секунду – счетчик кадров необходим для определения эффективности работы разрабатываемой логики.

Программное средство предусматривает возможность формирования сугроба после преодоления снежинкой края экрана. Решается эта задача путем использования графического полигона совместно со списком, состоящим из количества элементов соответствующих количеству пикселей по оси X.

Для разработки пользовательского интерфейса программы выбрана технология Windows Presentation Foundation (WPF), которая является основной технологией построения графических интерфейсов в языке C#, при разработке оконных программ для операционных систем семейства Windows (рис. 1).

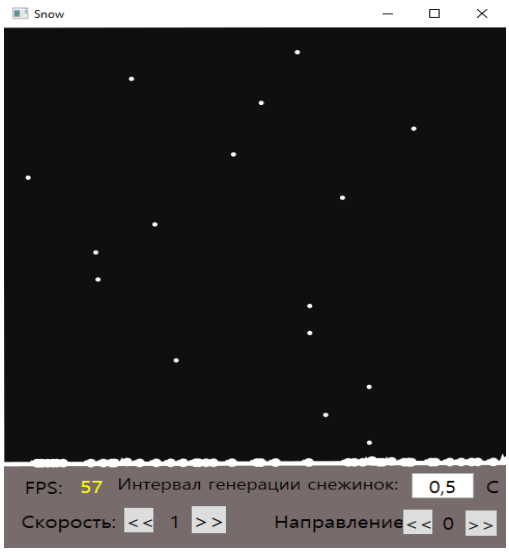

Рисунок 1. Пользовательский интерфейс программы

Каждая операция, необходимая для вынесения в отдельные потоки, не должна содержать в себе параллельности, для этого определен класс, отвечающий за объединение этих операций и внедрение параллельности. Этот класс включает метод, который реализует бесконечный цикл с вызовом на параллельное исполнение описанных операций. Спроектированное решение позволяет запустить симуляцию без дополнительных надстроек на рабочем слое. Помимо этого бесконечный цикл в данном методе необходим в связи с тем, что операциям свойственно единовременное выполнение.

В качестве хранилища снежинок используется коллекция «Список». Она расположена в классе симуляции, это способствует возможности работы операций с одним набором данных.

В момент запуска симуляции для генерации снежинок создается таймер, который при превышении заданного заранее периода обнуляется и создает снежинку.

Движение снежинок должно осуществляться по двум осям, поэтому необходимо обработать соответствующие условия:

− снежинка при приземлении упадет в сугроб или сама создаст его – в зависимости от этого необходимо проверять границу по оси Y;

− снежинка преодолела границу сугроба по оси Y – данное условие является вложенным для вышеописанного условия;

− снежинка преодолела границу холста по оси Y;

− снежинка преодолела границы по осям X – данное условие обрабатывает проверку на преодоление снежинкой левой и правой границы холста.

Движение снежинок можно осуществить двумя способами: выделить параллельный процесс для каждой из снежинок или в качестве параллельного процесса обрабатывать весь список снежинок. Первый вариант может вызвать ошибку одновременной обработки списка данных из различных процессов, поэтому выбран второй вариант.

Обновление холста представляет абстракцию для возможности адаптации программы под различные технологии разработки пользовательского интерфейса. Класс симуляции должен содержать замещающий метод обновления холста для возможности подсчета числа кадров.

Подсчет числа прорисованных кадров вычисляется каждую секунду благодаря таймеру. При вызове метода отображения снежинок происходит увеличение определенного поля, по превышению счетчика таймера в одну секунду значение этого поля переносится, а само поле очищается для повторного расчета.

Заключение. В ходе исследования на основе итерационного процесса составления программ, когда каждая функция имеет несколько итераций, на каждой из которых производятся определенные вычисления, с уникальным набором значений переменных, разработан законченный мини-проект с применением методов параллельного программирования. Ознакомиться с исходным кодом программы можно в GitHub репозитории «denden1s/Snow-simulation».

Список использованных источников

1 Параллельные вычисления [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Параллельные_вычисления. – Дата доступа: 20.03.2022.