

(для создания уровней) и несколько дополнительных утилит. Но есть большой минус, не самая лучшая графика.

Таким образом, с каждым годом игровая графика развивается, раньше были игры плоские, привлекали только сюжетом, но, благодаря развитию, нам предоставлена возможность наслаждаться 3D-графикой, она передает атмосферу игры и благодаря ей, сейчас много любителей погрузиться в виртуальный мир игр.

УДК 004.67

Балашкова Е. М., Каврук В. А.

### **СТРУКТУРА ДАННЫХ «ФИЛЬТР БЛУМА»**

*Белорусский национальный технический университет,*

*г. Минск, Республика Беларусь*

*Научный руководитель: ст. преподаватель Астанчик Н. И.*

Структуры данных используются для хранения информации в упорядоченном виде, а данные – самый важный феномен в информатике, поэтому истинная ценность структур данных очевидна. Наиболее распространены структуры данных, которые дают точные ответы: массивы, стеки, очереди, файлы, списки, деревья, графы, хеш-таблицы.

Кроме перечисленных структур, существуют вероятностные структуры данных, которые не дают точного ответа, а предоставляют приближение к ответу и способ приблизить эту оценку. Они чрезвычайно полезны в работе с большим объемом данных ввиду того, что позволяют значительно уменьшить объем необходимой памяти. Одной из используемых вероятностных структур данных является Блум-фильтр.

Фильтр Блума представляет собой битовый массив из  $m$  бит и набор из  $k$  различных хеш-функций  $h$ , выдающих значения от 0 до  $m-1$ , соответствующее номерам битов в массиве. Изначально, когда структура данных хранит пустое множество, все  $m$  бит обнулены.

Фильтр реализует вероятностное множество всего с двумя операциями: добавление элемента к множеству и проверка принадлежности элемента множеству.

Для добавления элемента  $e$  необходимо записать единицы на каждую из позиций  $h_1(e), \dots, h_k(e)$  битового массива.

Для проверки принадлежности достаточно посчитать значения хеш-функций для потенциального члена и убедиться, что все соответствующие биты установлены в единицу – это и будет ответом «возможно». Если же хотя бы один бит не равен единице, значит, множество этого элемента не содержит – ответ «нет».

В ответ на запрос поиска есть вероятность получить положительный ответ, даже если этого элемента в данном множестве нет. Но если же ответ фильтра был отрицательным, запрашиваемого элемента точно нет. Чем больше размер этого множества, тем меньше вероятность получить некорректный ответ на запрос о наличии какого-либо элемента.

Обычно фильтр применяется для уменьшения числа запросов к несуществующим данным в структуре данных с более дорогостоящим доступом – «фильтрации» запросов.

Фильтры Блума используются Google BigTable, Apache HBase и Apache Cassandra, а также PostgreSQL для сведения к минимуму числа обращений к жесткому диску при проверке на существование заданной строки или столбца в таблице базы данных. Такой подход к нахождению необходимого элемента в базе данных значительно ускоряет сам процесс поиска и уменьшает количество обращений к жесткому диску.

Компьютерные программы для проверки орфографии основаны на Блум-фильтрах.

Bitcoin использует фильтр Блума для ускорения синхронизации с кошельком.

Quora внедрила общий фильтр Блума в бэкэнд-канал, чтобы отфильтровать истории, которые люди видели раньше.

В веб-браузере Google Chrome использовался фильтр Блума для выявления вредоносных URL-адресов.

Из вышесказанного следует, что преимущество фильтра Блума перед другими структурами данных в том, что он может обходиться на несколько порядков меньшими объемами памяти, жертвуя детерминизмом.

УДК 004.85

Балашкова Е. М., Каврук В. А.

## **МАШИННОЕ ОБУЧЕНИЕ С ПОМОЩЬЮ PYTHON**

*Белорусский национальный технический университет,*

*г. Минск, Республика Беларусь*

*Научный руководитель: ст. преподаватель Астапчик Н. И.*

В настоящее время организации вкладывают значительные средства в новые технологии, такие как искусственный интеллект и машинное обучение, чтобы построить модели для решения реальных проблем с использованием данных.

Машинное обучение считается областью искусственного интеллекта, основная идея которой заключается в том, чтобы компьютер самостоятельно обучился решению поставленной задачи, а не просто использовал заранее написанный алгоритм.

Примерами применения машинного обучения являются:

- системы распознавания речи, жестов, образов;
- рекомендательные системы сайтов;
- медицинская диагностика;
- обнаружение мошенничества;
- категоризация документов.

В машинном обучении есть большое число алгоритмов различной сложности. Язык программирования Python подходит для выполнения таких задач, потому что у него отличная производительность при обработке данных. Данный язык от-