

**Integer Math** – подсчёт операций с целыми числами в секунду. (MOps./Sec.)

**Prime Numbers** – находит простые числа. Сбрасывает в начало цикла после 150 тыс. перебранных значений. (Million Primes/Sec.)

**Compression** – используется метод кодирования (Ian H. Witten, Radford M. Neal, and John G. Cleary in an article called “Arithmetic Coding for Data Compression”.). Подсчитывает количество закодированной (сжатой) информации. (Kbytes/Sec.)

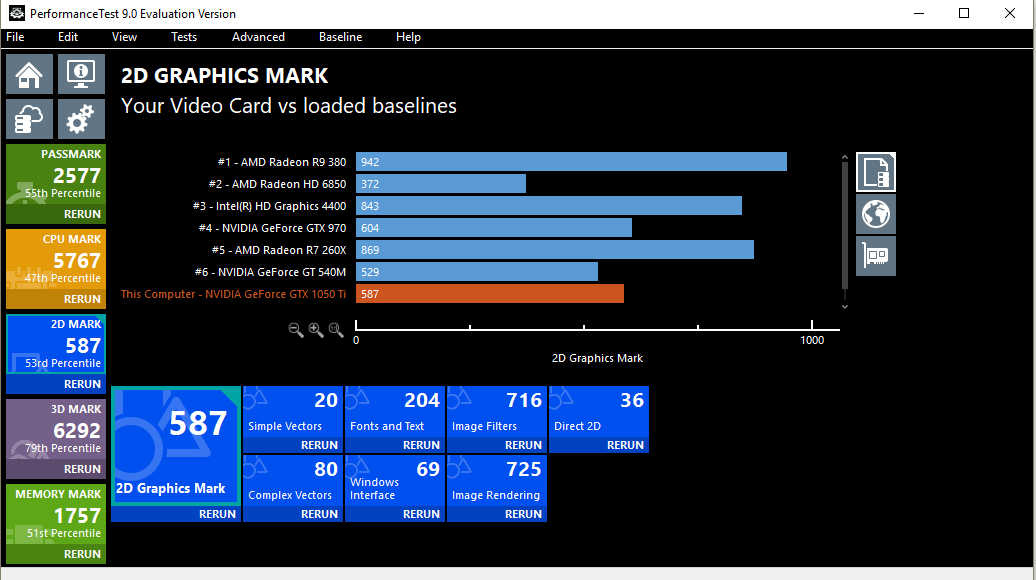
**Physics** – симуляция физики, использующая [Bullet](http://www.bulletphysics.org/) Physics Engine. Повторяет первые секунды моделирования сколько возможно. (Frames/Sec.)  
  
**CPU Single Threaded** – симулирует работу одного ядра. Комбинирует кодирование, операции над числами с плавающей точкой и кодирование. (MOps./Sec.)

**Floating Point Math** – подсчёт операций с плавающей точкой в секунду. (MOps./Sec.)

**Extended Instructions (SSE)** – подсчёт SSE-операций в секунду . (MOps./Sec.)

**Encryption** – использует библиотеку Crypto++. Каждый тест с хэш-функцией или с шифром действует с буффером в один мегабайт. (Mbytes/Sec.)

**Soring** – сортирует 500,000 строк, по 25 символов каждая. Повторяет, при возможности. (Thousand Strings/Sec.)



**Простые векторы** – рисует прямоугольники, покрашенные градиентом, эллипсы, линии, кривые и звёздочки столько раз, сколько это возможно. Каждый 32 цикл цвета меняются на случайный. Считает тысячи операций с векторами в секунду. (thousand vectors/Sec.)

**Комплексные векторы** – рисует сцену джунглей, используя изображения зверей и деревьев. Эта сцена будет нарисована столько раз, сколько это будет возможно за отведенный промежуток времени. Считает кол-во операций с комплексными векторами в секунду. (complex vector/Sec.)

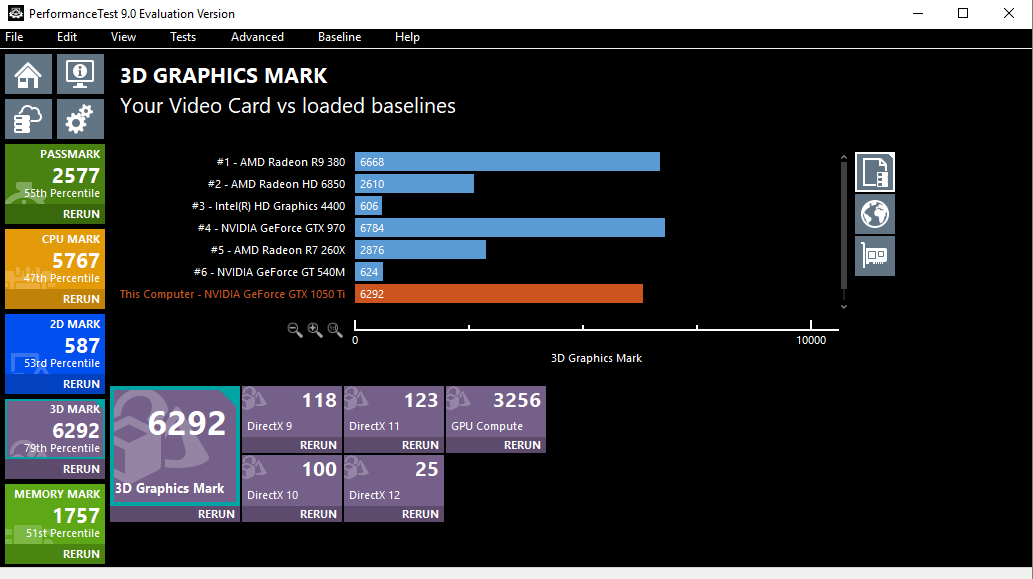
**Шрифт и текст** – тестирует производительность графической карты с рендерингом шрифтов и текста. (Ops./Sec.)

**Windows интерфейс** – тестирует производительность графической карты и настроек windows при взаимодействии с GUI. Тест включает в себя замеры производительности стандартных элементов управления GUI: treeview, listview, слайдеры и окна изменения текста. Так же тестируется изменения размеров окна и его положения. (Ops./Sec.)

**Фильтры изображений** – применяет фильтры поворота, яркости и контраста к изображению. Считает количество операций в секунду. (Filters/Sec.)

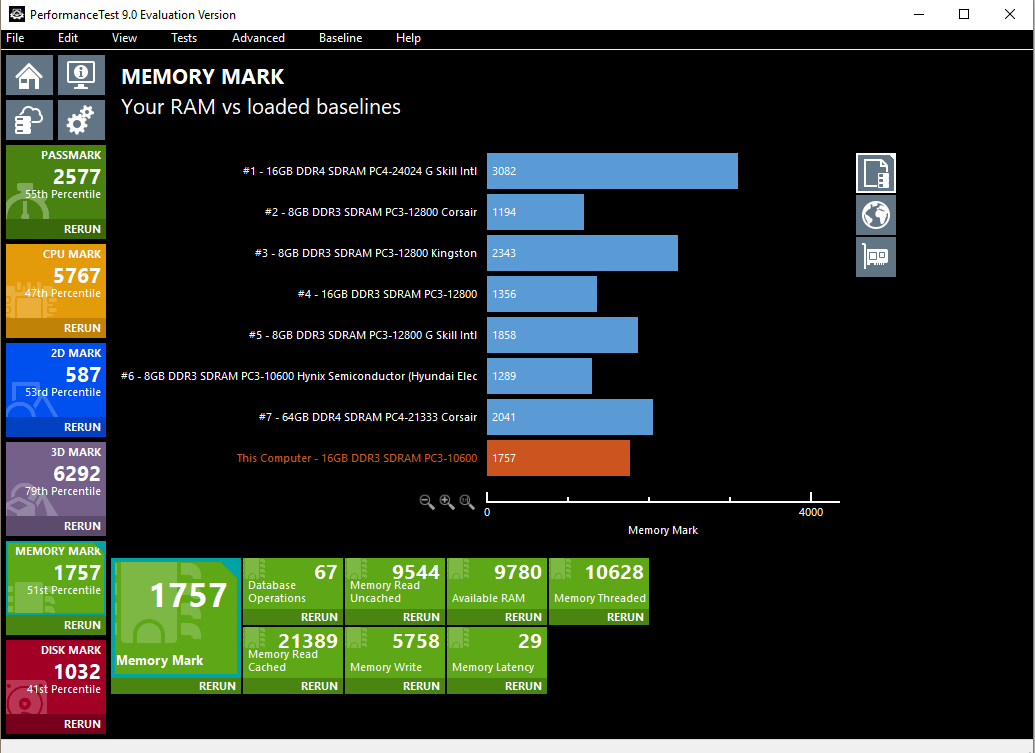
**Рендер изображения** - измеряет скорость с которой изображение может воспроизвестись при разных размерах. Тест всегда оканчивается при завершении цикла. (Images/Sec.)

**Direct 2D** – тестирует производительность Direct 2D. Direct 2D это система доступная в Windows с версии Vista и использует графическую карту для ускорения рендеринга. (Frames/Sec.)



Запускает 4 теста на DirectX (9, 10, 11, 12):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | DirectX 9 | DirectX 10 | DirectX 11 | DirectX 12 |
| **Объекты** | 7 планет, 500 деревьев, земля, вода, небо | 10 островов, 20 метеоров. | 50 гигантских медуз | До 100,000 астеройдов, 71 космических кораблей, 1 космическая станция |
| **Режим** | Полноэкранный, 1024х768, 4х Сглаживание | Полноэкранный, 1680х1050, 8х Сглаживание | Полноэкранный, 1920х1080, 4х Сглаживание | Полноэкранный, 3840х2160, 2х Сглаживание |
| **Основные характеристики** | Технология создания реалистичной воды и изменение текстуры земли в зависимости от высоты над водой.  (Vertex and Pixel Shader 2.0) | Vertex and Pixel Shader 3.0 effects. | Vertex and Pixel Shader 5.0 effects.  Прозрачность медуз (только на 8x Сглаживание)  Тесселяция земли. | Vertex and Pixel Shader 5.1 effects.  Шейдеры цвета и деформации космических кораблей. 5 текстур на каждом космическом корабле.  Поддержка многопоточности, пакетов и списков команд из DX12 |
| **Информация о ресурсах** | Земля – 32.258 треугольных полигонов.  Вода – 260.610  Другие – 560.  Кажое дерево – или 2214 или 1978. | Каждый метеор имеет до 100.000 частиц.  Каждое дерево имеет 2497 листовых полигонов. | Каждая медуза – 8244 полигона.  Космическая станция – 2843.  Каждая из примерно 10,000 звезд – полигон.  Количество полигонов земли зависит от размеров и расстояния камеры | Полигоны каждого корабля – 5,000-30.000.  71 разных текстур. Максимальный размер которых 2048х2048. |



**Database Operations** – этот тест сильно нагружает C++ STL контейнеры чтобы проверить способность память поддерживать большие структуры данных (такие как базы данных). (KOps./Sec.)

**Read Cached** – этот тест измеряет время чтения маленького блока памяти. Этот блок достаточно маленький, чтобы поместиться в кэше (если он есть). (Mbytes/Sec.)

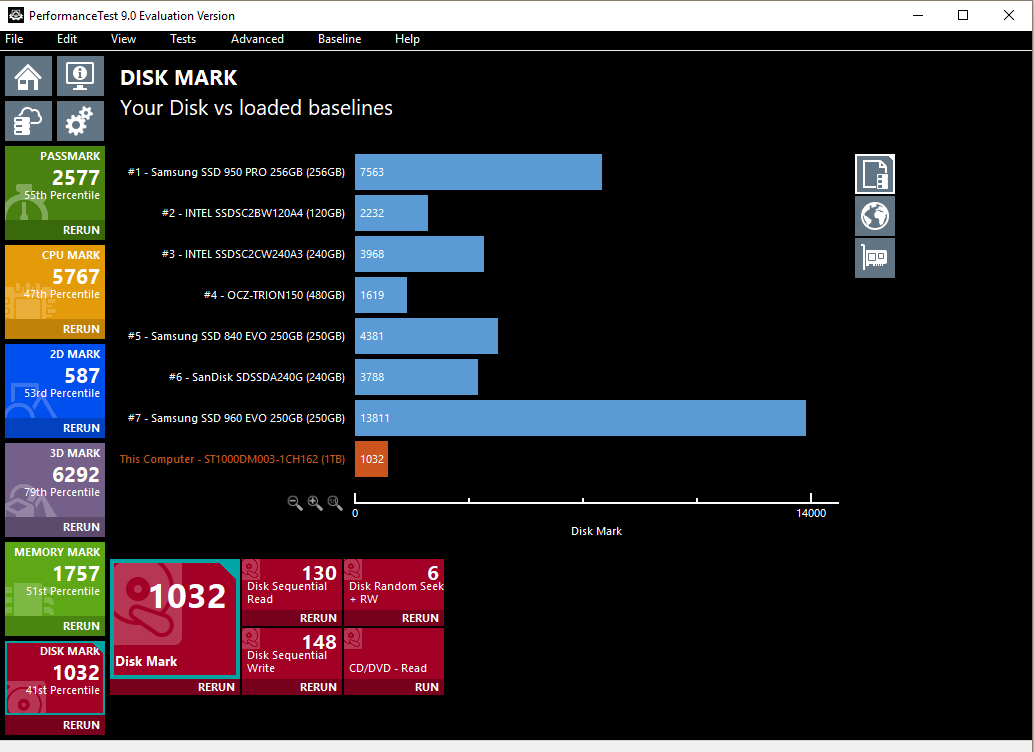
**Read Uncached** – этот тест измеряет время чтения большого блока памяти. Этот блок слишком большой и не может уместится в кэше. (Mbytes/Sec.)

**Write** – тест измеряет время, которое тратиться на запись информации в память. (Mbytes/Sec.)

**Available RAM** – тест просто показывает сколько памяти доступно для приложений. Это не полное количество памяти доступное в системе, память используемая в других приложениях не учитывается. (Megabytes)

**Latency** – время требуемое чтобы перенести один байт памяти в CPU для обработки. Измеряется в наносекундах. (ns)

**Threaded** – тест похож на Read Uncached, но использует два отдельных процесса одновременно чтобы протестировать как память копируется с несколькими одновременными доступами. (Mbytes/Sec.)



**Последовательное чтение –** создаётся файл, который читается последовательно от начала до конца. (Mbytes/Sec.)

**Последовательная запись –** создаётся файл, в который последовательно записывается информация. (Mbytes/Sec.)

**Случайный поиск по диску –** создаётся файл, поиск по которому осуществляется по случайному числу, а затем происходит запись или чтение 16 Kbytes данных. (Mbytes/Sec.)

**Вывод:** при выполнении Лабораторной работы была применено ПО «PerformanceTest by PassMark Software». Программа даёт обширный инструментарии по тестированию производительность основных ресурсов ПК. Учитываются все основные операции совершаемые при работе с компьютером во всех сферах деятельности(от элементарных операций до работы с данными, графикой и алгоритмами). ПО имеет полную, необходимую и достаточную информацию о системе.