2. AI для людей: простыми словами о технологиях // [Электронный ресурс] //ХабраХабр: URL - <https://habr.com/ru/company/jetinfosystems/blog/471626/>
3. Machine Learning: где деньги лежат? // [Электронный ресурс] //ХабраХабр: URL - <https://habr.com/ru/company/jugru/blog/340026/>
4. Rise of the Machine Learning: How AI Helps Create Photorealistic Digital Humans // [Электронный ресурс] // NVIDIA BLOG URL - <https://blogs.nvidia.com/blog/2019/10/10/machine-learning-digital-humans/>

Сацук М.М., Терехова В.С., Попов А.А.

Разработка программного обеспечения для повышения производительности программных продуктов, разработанных на игровом движке «Unity»

Сибирский государственный университет науки и технологий имени академика

М. Ф. Решетнева

(Россия, Красноярск)

doi:10.18411/sciencepublic-12-12-2019-07

idsp: sciencepublic-12-12-2019-07

Аннотация

Рассматривается проблема низкой производительности программ, в частности игр, разработанных на игровом движке Unity. Целью данной работы является разработка программного обеспечения, позволяющего повысить производительность нижеперечисленных программных продуктов.

В данной статье представлены основные технические особенности Unity, а также аспекты программной реализации разрабатываемой программы. Разработка велась в кроссплатформенной среде разработки Code::Blocks на языке программирования C++.

Ключевые слова: повышение производительности, программное обеспечение, оптимизатор, игровой движок, Unity.

Abstract

The problem of low performance of programs, in particular games developed on the Unity game engine, is considered. The purpose of this work is to develop software that can improve the performance of the above software products.

This article presents the main technical features of Unity, and aspects of the software implementation of the described program. Development was conducted in the cross-platform development environment Code:: Blocks in the C++programming language.

Keywords: performance enhancement, software, optimizer, game engine, Unity.

Далее Актуальность темы исследования обусловлена большой популярностью программных продуктов, в особенности компьютерных игр, разработанных на игровом движке Unity [1]. Результаты многочисленных исследований показывают, что данная платформа обладает низкой производительностью, потреблением большого объема памяти и слабой работоспособностью на мобильных устройствах [2]. Это связано с высоким уровнем абстракции как самого основного языка программирования C#, так и самого движка, что безусловно упрощает и ускоряет процесс разработки, но сильно влияет на быстродействие.

Преимуществами данного игрового движка является [3]:

1. Игровой движок предоставляет огромное множество функциональных возможностей, с возможностью интеграции в различные программы и компьютерные игры. В это множество входит следующее: моделирование физических сред, динамические тени, карты нормалей и т.д.
2. Наличие визуальной среды разработки.

3. Межплатформенная поддержка. Под этим пунктом подразумевается не только места развертывания (установка на мобильных устройствах, игровых консолях и на персональных компьютерах), но и наличие инструментария разработки (интегрированная среда разработки может использоваться и под операционной системой Mac OS, и Windows).
4. Модульная система компонентов Unity, благодаря которым происходит конструирование игровых объектов, в то время как последние представляют собой комбинируем пакеты функциональных элементов. По сравнению с механизмами наследования, объекты в описываемом движке реализуются посредством объединения функциональных блоков, а не при помощи помещения в узлы дерева наследования. Данный подход облегчает процесс создания прототипов что актуально при разработке игр.

Недостатки:

1. Ограничение визуального редактора при работе с многокомпонентными схемами, из-за чего в сложных сценах визуальная работа затрудняется.
2. Отсутствие поддержки Unity ссылок на внешние библиотеки, работу с которыми программистам приходится настраивать самостоятельно, в результате этого затрудняется командная разработка.
3. Использование шаблонов экземпляров. С одной точки зрения эта концепция предполагает гибкий подход визуального изменения объектов, но с другой стороны, редактирование таких шаблонов крайне затруднительно.
4. Недостаток WebGL-версии движка, в силу специфики архитектуры – трансляция программного кода из C# в C++ и далее в JavaScript, имеется ряд нерешенных проблем с быстродействием, потреблением памяти, производительностью и работоспособностью на мобильных устройствах.
5. Минимальный набор для разработки Unity с 2D-графикой требует более 1 гигабайта памяти на локальном жестком диске.

Поскольку каждый программный продукт, реализуемый на игровом движке Unity имеет свои конфигурационные файлы проекта, в которых зафиксированы все настройки программы, разработанный алгоритм оптимизации можно применять к большинству приложений. В качестве примера был выбран проект «Prime World». Это условно-бесплатная ролевая многопользовательская онлайн стратегия с элементами МОБА, разработанное на игровом движке Unity, компании Nival. Тестирования замеров производительности проводились конкретно в этой игре. Выбор проекта обусловлен техническими особенностями, а именно допущением разработчиков к редактированию конфигурационных файлов под свое усмотрение, отсутствием строгой политики в отношении изменения файлов игры, а также общей необходимостью большинства пользователей к повышению производительности. Разработанная программа так и называется – Оптимизатор Prime World (OptimizerPrimeWorld).

Для программной реализации был выбран язык программирования C++ [4], используя среду разработки Code::Blocks. Это свободная кроссплатформенная среда разработки, написанная на C++ и использует библиотеку wxWidgets. Имеет открытую архитектуру, с возможностью масштабироваться за счёт подключаемых модулей.

В качестве интерфейсной составляющей, разрабатываемая программа выполнена в интерфейсе командной строки Windows, что позволяет ускорить как время разработки описываемого приложения, так и его быстродействие. Из технических требований следует отметить что программа работает только на операционной системе Windows, начиная с версии WindowsXP, занимая 640 КБ места на жестком диске.

Алгоритм анализирует технические характеристики компьютера, на котором был запущен оптимизатор, и подстраиваясь под имеющиеся ресурсы компьютера –

подбирает оптимальные параметры приложения на Unity, благодаря которым качество графики изменяется не существенно, но при этом ощущается большой прирост быстродействия (параметр FPS).

Рядовому пользователю недоступны большая часть настроек, которые было бы можно изменять. Проанализировав слабые места игрового движка и исходные файлы игрового проекта Prime World, были выявлены параметры конфигурации, которые пагубно влияют на производительность, и при этом не несут особо важной смысловой нагрузки, в том числе не влияют на сам игровой процесс.

В главном окне программы (рис. 1) расположено 3 ключевых блока, которыми оперирует пользователь: оптимизация игры, оптимизация игрового лаунчера (клиента игры), и очистка жесткого диска от накопившегося мусора, в частности старых лог-файлов игры. Использование каждого из этих пунктов увеличивает производительность различных составляющих игры, что в совокупности и приводит к приросту быстродействия проекта «Prime World» в целом. Ключевой особенностью является возможность вернуть все настройки в исходное состояние (рис. 2). Также присутствует алгоритм проверки на совместимость программы-оптимизатора с операционной системой пользователя и клиентом игры, во избежание возможных проблем на этапе оптимизации.

Рис. 1. Главное окно программы OptimizerPrimeWorld

Рис. 2. Окно запуска оптимизации игры

В результате работы была разработана полноценная программа для повышения производительности игрового движка Unity, на примере повышения быстродействия игры Prime World.

Разработанное программное обеспечение успешно внедрено в реализацию, за время внедрения было установлено на компьютеры, с различной аппаратной конфигурацией, более 700 пользователей, которые оценили повышение производительности.

В дальнейшем планируется учитывать больший объем исходных файлов проектов на Unity, с целью выявления возможных оптимизаций конфигурационных файлов для еще большего повышения производительности проектов.

1. Подробная информация о возможностях игрового движка и его технические характеристики [Электронный ресурс]. 2019. URL: <https://unity3d.com/unity> (дата обращения 17.11.2019).
2. Оптимизация производительности игры параметрами импорта звука Unity [Электронный ресурс]. 2019. URL: <https://habr.com/ru/post/437474/> (дата обращения: 22.11.2019).
3. Особенности, преимущества и недостатки движка Unity [Электронный ресурс]. 2019. URL: <https://cubiq.ru/dvizhok-unity/> (дата обращения: 27.11.2019).
4. Джефф Кент, C++ Основы программирования - М.: NT Press Москва, 2013. - 365 с.
5. Макс Шлее, Qt 5.10 Профессиональное программирование на C++ - М.: БХВ-Петербург, 2018. - 1051 с.

Синицын С.В., Семикина Е.В., Долбилов А.Д.

Повышение эффективности автоматических систем управления за счет упрощения программно-алгоритмического обеспечения

*Филиал ФГКВУ ВО «Военная академия ракетных войск стратегического назначения имени Петра Великого» Министерства обороны РФ
(Россия, Серпухов)*

doi:10.18411/sciencepublic-12-12-2019-08

idsp: sciencepublic-12-12-2019-08

Введение и обоснование актуальности упрощения программно-алгоритмического обеспечения автоматических систем управления.

Традиционно выделяют два основных направления обеспечения эффективности автоматических систем управления (АСУ): аппаратное – за счет применения современной элементной базы и программно-алгоритмическое – за счет реализации эффективного программного обеспечения (ПО). Как правило, в настоящее время, первое направление реализуется посредством разработки и изготовления финальных изделий российского производства из импортной элементной базы и на оборудовании ведущих западных и китайских фирм. Приоритет этого направления обоснован недостаточным развитием российского промышленного производства изделий отечественной микроэлектронной промышленности. Попытки производителей изготовить электронное устройство автоматического управления на основе российской элементной базы ведет к увеличению его стоимости в несколько раз, что априори коммерчески неэффективно. Например [1], стоимость отечественного 32-х разрядного микроконтроллера K1921BK01T в разы превышает стоимость его аналога LM4F132, а если учесть стоимость средств отладки, необходимых для разработки изделий, то разница в цене становится существенней.

Недостатком применения импортной микроэлектроники является невозможность изготовления на ее основе АСУ образцов вооружения и военной техники, а так же продукции двойного назначения. Поэтому изготовление такой продукции на основе отечественной микроэлектроники остается актуальным направлением, которое в настоящее время подчеркивается программой «импортозамещения».