

Методы Программной Инженерии в Индустрии Компьютерных Игр

Елена Анатольевна Павлова
Московский инженерно-физический
институт (государственный университет)
email: Elena.pav@gmail.com

Александр Викторович Гаврилов
ООО «Майкрософт Рус»
email:
Alexander.Gavrilov@microsoft.com

Абстракт

В настоящей работе рассматривается возможность применения методов программной инженерии в индустрии компьютерных игр. Анализируются особенности разработки игры как программной системы. Демонстрируется эффективность применения методов программной инженерии в задачах разработки игр на примере применения формальных предметно-ориентированных языков для разработки правил игр в классе пошаговых стратегий. Применение таких языков является новым подходом к разработке правил компьютерных игр.

Ключевые слова: компьютерные игры; методы программной инженерии; формальный язык; предметно-ориентированный язык.

Введение

Компьютерные игры – одна из самых динамично развивающихся областей информационных технологий [1]. Доходы мировой индустрии игр за 2007 год возросли по различным оценкам на 60-67% [2, 3]. При этом, согласно данным Forbes и Games Industry [2], доход Европейского рынка видеоигр в 2007 году составил 17,9 миллиардов долларов, а рынка игр США – 18,8 миллиардов, что сопоставимо с доходами киноиндустрии. Число потребителей компьютерных игр также продолжает расти. Например, в Китае за 2006 и 2007 годы общее число игроков возросло на 6 миллионов человек и составило 40 миллионов [4]. Ожидания потребителей игр постоянно растут, порождая всё более высокие требования к качеству игр и разнообразию их содержания. Заметим, что кроме развлекательного аспекта, игры применяются в обучении и при создании систем подготовки различных специалистов [5].

Исторически индустрия игр являлась частью индустрии развлечений [1] и развивалась независимо от индустрии разработки программного обеспечения. В результате, индустрия игр отличается значительно

более развитой маркетинговой стороной по сравнению с индустрией программного обеспечения и значительно уступает по степени развития процессов разработки, применяемых методов, инструментальных средств, и, соответственно, отличается более низким качеством продуктов.

Часто разработка игр ведётся без использования устоявшихся методов разработки, процессы и методы изобретаются компаниями или отдельными командами самостоятельно «с нуля» для каждого проекта и являются закрытой информацией [6]. Документация может отсутствовать на протяжении нескольких этапов разработки и иметь слабоструктурированную форму, что отражается на её актуальности и приводит к противоречиям.

Например, необходимость проектной документации до сих пор вызывает оживлённую полемику и рассматривается значительной частью разработчиков как нововведение, допустимое для отдельных развитых компаний [7].

В литературе наиболее полно освещаются вопросы физического уровня проектирования игр и этапа реализации (примером могут служить работы [1, 8, 9, 10]). Большая часть работ посвящена решению задач трёхмерного моделирования и моделирования физических законов в играх (примером могут служить работы [8, 9, 10]). При этом общим вопросам разработки уделяется недостаточное внимание.

Отсутствие общепринятых методов разработки и общего научного обоснования является одной из причин низкой доли успешно завершённых проектов разработки компьютерных игр и недостаточно высокого качества выпущенных игр. Ежегодно в мире создается от 3 до 5 тысяч компьютерных игр, из них только несколько десятков становятся программными продуктами и появляются в продаже [11].

Вышесказанное определяет целесообразность применения методов программной инженерии в индустрии игр как способа повышения качества игр.

Задачей настоящей работы является рассмотрение методов программной инженерии и возможностей их адаптации с точки зрения применения в индустрии

игр. Конкретно рассматривается возможность применения формальных предметно-ориентированных языков для разработки правил игр в классе пошаговых стратегий.

1. Специфика применения методов программной инженерии в разработке игр

Вопросы применения методов программной инженерии к разработке игр рассматриваются, например, в работах [12, 13, 14] с точки зрения разработчиков и в работах [15, 16, 17] с научной точки зрения. Однако в перечисленных работах понятие разработки игр ограничивается любительскими проектами для небольшой команды разработчиков. Соответственно, рассматриваются лишь частные случаи применения методов программной инженерии в разработке игр, и не рассматривается влияние, которое может оказать применение методов программной инженерии на индустрию игр. Кроме того, в ряде случаев не учитывается специфика игр, рассмотрение которой с точки зрения программной инженерии позволило бы адаптировать ряд методов и, в итоге, повысить качество игры.

Рассмотрим подробнее особенности игры как программной системы. Специфика требований к играм состоит в следующем:

1. заказчиком игры, как правило, является её издатель. Цель заказчика состоит в получении максимальной выгоды от продаж игры, что отличает его от заказчика в индустрии программного обеспечения, где например, заказчик может планировать внедрение программы в своей компании и может быть явно заинтересован в функциональности программы

2. аудитория игры, как правило, очень широка (с точки зрения возраста и рода деятельности), что порождает дополнительные трудности при сборе требований

3. в игровой индустрии очень сильны творческий и инициативный аспекты, что часто усложняет сбор требований

4. игры очень быстро устаревают морально, поэтому требования к срокам разработки игр выше, чем в индустрии программного обеспечения

5. игры наиболее часто носят развлекательный характер и не предназначены для решения производственных задач. Одним из требований к играм является минимизация времени обучения игре.

6. игры предъявляют очень высокие требования к аппаратному обеспечению, что объясняется, в основном, особенностями графических компонент.

Например, игры часто требуют использования наиболее совершенных видеоадаптеров, аудиоадаптеров, наиболее быстрых шин передачи данных

7. игры часто требуют использования устройств, которые редко используются другими классами программ (например, различные манипуляторы, рули, педали, сенсорные ковры)

8. к некоторым частям кода игр предъявляются очень высокие требования производительности, например, к коду графических подсистем.

Особенности проектирования игр состоят в следующем:

1. сценарии игр значительно отличаются от сценариев, создаваемых для вариантов использования в индустрии программного обеспечения, и более близки к сценариям кинофильмов

2. написание сценария игры и разработка концепции осуществляется нетехническим специалистом

3. значительную часть игровых проектов по сравнению с проектами индустрии программного обеспечения составляют работы дизайнеров, художников, музыкантов, специалистов по видео и трёхмерной графике. Процессы и методы для этой части проекта существенно отличаются от индустрии программного обеспечения.

4. циклическая разработка для художественной части игрового проекта считается слишком затратной и модель работы напоминает водопадную, что отражается на общей модели разработки игры и качестве игры

5. в индустрии игр существует общепринятое деление игры на высокоуровневые компоненты [1, 8]: логический, физический, графический компоненты, компоненты искусственного интеллекта, данных и взаимодействия с пользователем

6. некоторые аспекты проектирования и реализации игр достаточно хорошо проработаны в индустрии и нашли своё воплощение в компонентах, называемых игровыми движками. Несколько таких компонент используется при разработке практически каждой игры, так как реализация всех высокоуровневых компонент игры с нуля считается затратной и нецелесообразной. Использование сторонних высокоуровневых компонент накладывает значительные ограничения на проектирование и реализацию игры.

Особенности реализации игр состоят в следующем:

1. практически в каждом проекте используется сложная с точки зрения индустрии программного обеспечения графика. При этом доля проектов,

использующих трёхмерную графику, продолжает расти.

2. значительная часть игр предполагает взаимодействие двух и более (до нескольких миллионов) игроков по сети в режиме «мягкого» реального времени.

3. значительная часть игр использует элементы искусственного интеллекта для моделирования поведения персонажей в игре.

4. в играх часто применяются системы захвата движения и сопутствующие программные модули

Следует отметить, что ряд аспектов применения методов программной инженерии в разработке игр уже достаточно хорошо проработан. Так, сопровождение и управление конфигурацией в ряде случаев уже вышли на промышленный уровень [18]. При этом процессы сбора требований, проектирования и конструирования программного обеспечения зачастую остаются на ремесленном уровне «штучных поделок» [1, 8].

Недостаточно развиты области тестирования и управления качеством. Например, тестирование во многих случаях выполняется вручную разработчиками до выпуска продукта или перекладывается на плечи пользователей после выпуска. Этот принцип получил широкое распространение и описывается в литературе как “play a little, tweak a little” [1, 8] (немного поиграй и настрой).

Масштабное использование методов программной инженерии (в особенности в проектировании и конструировании) может позволить увеличить долю успешных проектов, повысить общий уровень качества продуктов и снизить себестоимость создания программного продукта за счет создания специализированных средств разработки и повторного использования компонент и шаблонов.

2. Применение формальных методов программной инженерии для проектирования правил игр в классе пошаговых стратегий

В качестве примера применения методов программной инженерии в разработке игр рассмотрим разработку игр в классе пошаговых стратегий, специфичные для этого класса задачи и решение одной из задач при помощи формальных методов программной инженерии. Согласно работе [19] формальные методы (в частности формальные языки), относятся к области знаний (SE knowledge area) методов и средств программной инженерии (software engineering tools and methods).

Варианты классификации игр и особенности пошаговых стратегий детально рассматриваются в [1, 8, 9, 10]. В качестве исследуемой задачи рассмотрим разработку правил игры, как одну из ключевых для выбранного класса.

Разработка правил игры включает в себя этап проектирования правил [1], которому с точки зрения программной инженерии могут быть поставлены в соответствие сбор требований и создание концепции. Затем следует этап проверки и балансировки правил. Далее выполняется реализация правил в виде кода на языках программирования высокого уровня. Этот этап может быть отнесён к этапу реализации в терминологии программной инженерии. Как правило, дизайнер периодически (десяtkи раз) вносит исправления в правила и процесс повторяется с начала [1]. Заметим, что в описанном процессе с точки зрения программной инженерии полностью отсутствует значительная часть этапа проектирования.

Проектирование правил игры традиционно выполняется вручную дизайнером игр [1, 8], который не является техническим специалистом [1]. Правила представляются в виде слабо структурированных документов и обширных таблиц. Автоматизация разработки правил, их последующей обработки и генерация кода на основе правил существенно затруднены. В настоящей работе в качестве одного из вариантов решения проблемы будет рассмотрена разработка и применение специализированного средства создания правил, использующего свой внутренний формальный язык представления правил.

Такое специализированное средство позволит явно проектировать правила игры на концептуальном, логическом и физическом уровнях. Будет показано, что такое решение, характерное для задач автоматизации в индустрии разработки программного обеспечения, имеет ряд преимуществ и является предпочтительным по сравнению с альтернативными решениями.

3. Применение языка для проверки корректности правил игр

В основу разработки инструментального средства положена концепция использования формального предметно-ориентированного языка для разработки правил игр в классе пошаговых стратегий. Правила игры включают множество типовых сценариев игры.

Предложенный предметно-ориентированный язык позволяет описывать сущности из предметной области конкретной игры (а именно связанные с сущностями данные и поведение), определять

ограничения, налагаемые на поведение сущностей, определять отношения между сущностями, определять реакцию на поведение некоторой сущности со стороны других сущностей.

Корректность модели сценария игры проверяется путём проверки синтаксического и семантического соответствия интерфейсов взаимодействующих объектов модели при помощи метода, предложенного в работе [20].

Прототип результирующей программной системы получается на основании трансформационного подхода в рамках преобразования конструкций предметно-ориентированного языка на основании семантических правил.

Таким образом, использование языка позволяет не только описывать модель правил в терминах дизайнера игр, но и осуществлять быструю автоматизированную разработку верифицируемого прототипа.

Заключение

В настоящей работе была рассмотрена возможность применения методов программной инженерии в индустрии компьютерных игр. Проведённый анализ индустрии показал актуальность задачи повышения качества игр за счёт применения методов программной инженерии в разработке игр. Анализ специфики процесса разработки игр позволил определить этапы разработки игр, для которых наиболее целесообразно применение указанных методов и их адаптаций.

В настоящей работе рассмотрено использование формального предметно-ориентированного языка для создания средства разработки правил игр в классе пошаговых стратегий. Предложенный подход на основании формального языка даёт возможность разработать инструментальное средство, позволяющее более качественно проектировать правила игры. На примере использования языка была продемонстрирована эффективность методов программной инженерии для решения задач разработки компьютерных игр.

Источники

[1] Rollings A., Morris D. Game Architecture and Design. A New Edition.– Indianapolis: New Riders Publishing, 2004

[2] Androvich M. Forbes: Europe is least mature videogame market. – <http://www.gamesindustry.biz/articles/forbes-europe-is-least-mature-videogame-market>, 2008

[3] Путинцев Т. Игровая индустрия растёт. – <http://www.dtf.ru/news/read.php?id=31442>, 2003

[4] В Китае резко возросло число игроков. – <http://www.lenta.ru/news/2007/05/07/china/>

[5] Бондарева Т.В., Грызлова О.В., Евтюхин Н.В. Компьютерные деловые игры как инновационное средство обучения /Телекоммуникации и информатизация образования, (2001), 1 (январь), 58-69

[6] Плюммер Дж. Гибкая и масштабируемая архитектура для компьютерных игр. – <http://www.dtf.ru/articles/read.php?id=40757>, 2004

[7] Материалы конференции разработчиков компьютерных игр. КРИ 2007, – http://kriconf.ru/2007/index.php?type=info&doc=speech_records, 2007

[8] Wihlidal G. Game Engine Toolset Development.– Boston, MA: Thomson Course Technology PTR, 2006

[9] Meigs T. Ultimate Game Design: Building Game Worlds. NY: McGraw-Hill/Osborne, 2003

[10] Bates B. Game Design. Second Edition. – Boston, MA: Thomson Course Technology PTR, 2004

[11] Консалтинговая группа MD. – <http://www.md-consulting.ru/>

[12] Gruhl R. XNA Game Studio Express, an Overview. Proceedings of GDC 2007 – <https://www.cmpevents.com/sessions/GD/S4440i1.ppt>, 2007

[13] Flynt J. P., Salem O. Software Engineering for Game Developers. – Boston, MA: Thomson Course Technology, 2004

[14] Rucker R. Software Engineering and Computer Games.– London: Addison Wesley, 2003

[15] Yang H-C. A General Framework for Automatically Creating Games for Learning. Proceedings of ICALT'05. – IEEE, 2005, p. 28-29

[16] McNaughton M., Cutumisu M., Szafron D., Schaeffer J., Redford J., Parker D. Script-Ease: Generating Script-Code for Computer Role-Playing Games. Proceedings of ASE'04. – IEEE, 2004, p.386-387

[17] Meng L.S., Prakash E. C., Loh P. K. K. Design and Development of a Peer-to-Peer Online Multiplayer Game Using DirectX and C#. – IEEE, 2004, p. 278-281

[18] Common Questions about Blizzard – <http://www.blizzard.com/us/inblizz/genfaq.html>, 2008

[19] Guide to the Software Engineering Body of Knowledge. 2004 Version. SWEBOOK. – Los Alamitos, California: IEEE, 2004

[20] Гаврилов А.В., Павлова Е. А. Формализация проектирования сложных информационных систем на основе анализа функциональных интерфейсов. Информационные технологии №9, 2008. – М.: Новые технологии, 2008, стр. 9-15