

Оглавление

Благодарности и посвящение	5
Об авторах	5
Как устроена эта книга?	6
Введение. Игровая индустрия	8
Сообщество и события	9
Unreal Engine 5	11
Создание миров и совместная работа	16
Глава 1. Основы разработки игр	18
Игры и игроки	19
Что такое игровые механики и с чем их «едят»?	24
Игровые циклы и циклы разработки игр: смешать, но не путать	25
Что такое игровая документация?	26
Глава 2. Игровые движки	28
Как делают игры?	29
Что такое игровой движок?	30
Особенности Unreal Engine	32
Глава 3. Работа с контентом и ресурсами	36
Установка Unreal Engine и создание первого проекта	37
Интерфейс редактора	41
Стандарты наименования ассетов	42
Контент-браузер	43
Импорт	44
Текстуры	46
Материалы	49
Текстуры в виджетах (UMG)	51
Статические меши	52
UV-развертки	53
Создание материала с нуля	54
Экземпляр материала	58
Уровень детализации (LOD)	60
Редактор статических мешей	60
Акторы статических мешей	66
Пользовательские настройки коллизий	68
Скелеты и скелетные меши	72
Импорт	77
Аудио	79
Глава 4. Основы Blueprints	86
Визуальный язык программирования и его элементы	88
Переменные и типы данных	92
Функции, события и макросы	95

Узлы контроля потока выполнения программы	104
Таймеры и таймлайны	113
Объектно ориентированное программирование в Unreal Engine5	119
Классы в Unreal Engine	131
Настройка Input отображения для ввода данных	133
Level Blueprint	137
Gameplay Framework	138
Взаимодействие объектов	142
Интерфейсы	146
Коллизии	154
Отладка	161
Глава 5. Интерфейс с пользователем	164
Проектирование, ориентированное на пользователя (User-Centered Design)	165
Паттерны поведения и взаимодействия	166
Что такое интерфейс?	167
Обзор UMG	167
Независимый от разрешения экрана пользовательский интерфейс	172
Работа со стандартными виджетами	177
Общие настройки виджетов	190
Программирование UI	193
Проектирование интерфейса на основе данных (Data-driven UI)	201
Глава 6. Искусственный интеллект	204
Искусственный интеллект в играх	205
Встроенные методы реализации ИИ в UE	206
Подготовка материалов	206
Система навигации	209
Передвижение искусственного интеллекта	214
Система патрулирования	221
Искусственный интеллект и система восприятия	225
Дерево поведения	233
Система запросов среды (EQS)	250
Создание прототипа. Часть 1	262
Создание прототипа. Часть 2	266
Глава 7. Анимации	270
Анимация в Unreal Engine 5	273
Animation Sequence и импорт анимации	278
Анимирование движения нового персонажа	281
Анимация атаки и совмещение двух анимаций	289
Animation Layer Interface	293
Глава 8. Переход к C++ коду	298
Вместо заключения	306
Создание уровня с поддержкой World Partition	307
Lumen — новая система глобального освещения	310
Система геометрии Nanite	313
Chaos Physics — новое решение для моделирования физики	317
Новинки в анимации	318
Новая аудиосистема MetaSounds	318

Благодарности и посвящение

Посвящается нашему учителю Подбельскому Вадиму Валериевичу — доктору технических наук и ординарному профессору НИУ ВШЭ.

Никита Веселко, Ольга Максименкова

Мы выражаем глубокую признательность коллегам из игровой индустрии за обсуждения идеи этой книги, ценные комментарии и помощь в понимании особенностей отдельных направлений игровой разработки.

Авторы также благодарят студентов программы бакалавриата «Программная инженерия» факультета компьютерных наук НИУ ВШЭ Шалаеву Марину, Архарова Дмитрия, Панову Екатерину, Миронова Александра, Шепелина Дмитрия и Филатова Юрия за плодотворные обсуждения сложностей, возникающих у новичков, а также за оказанные авторам поддержку и активную помощь в подготовке некоторых текстов глав и примеров кода.

Об авторах

Ольга Вениаминовна Максименкова — кандидат технических наук, научный сотрудник международной лаборатории интеллектуальных систем и структурного анализа ФКН, Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ).

Никита Веселко — владелец продукта (Product Owner) в студии разработки игр IThub games, стажер-исследователь международной лаборатории интеллектуальных систем и структурного анализа ФКН, Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ).

Как устроена эта книга?

ДЛЯ КОГО ЭТА КНИГА

За последние несколько лет Unreal Engine эволюционировал из «просто» игрового движка в самый мощный инструмент для создания цифровых интерактивных миров. Сфера применения UE, куда входят, например, мобильные игры и разработка высокоточных физически корректных инженерных симуляций (какие незаменимы при создании авиационных двигателей, суперкаров и ядерных реакторов) поистине впечатляет!

Если вы хотите попробовать себя в качестве творца высококачественных синтетических миров и попытаться на новом уровне решить эту визуализационную задачу, но при этом не обладаете ни навыками программирования, ни опытом работы с таким огромным комплексным инструментом, как UE, эта книга поможет вам в осуществлении вашей мечты.

ЧТО ВЫ НАЙДЕТЕ В ГЛАВАХ?

Введение познакомит вас с областью геймдева и ее направлениями. Также здесь приведены списки основных интернет-ресурсов и мероприятий, посвященных игровой индустрии.

В первой главе содержится справочная информация о некоторых концепциях игрового дизайна. Это теоретический минимум, посвященный таким понятиям, как игры, игроки, игровая механика и игровой цикл. Здесь же вы узнаете о различных процессах разработки игр и разновидностях игровой документации.

Во второй главе вы познакомитесь с понятием игрового движка и его характеристиками, а также узнаете, чем отличается UE от других своих «собратьев».

Глава третья посвящена работе с контентом и ресурсами в UE: как добавить контент в проект, какие существуют особенности создания материалов и структур, подробный рассказ о статических и скелетных мешах и инструментах для работы с ними в UE. Уделено внимание и работе со звуком.

В четвертой главе находится информация о языке визуального программирования, который называется Blueprints. Чтобы начинающим было проще погрузиться в эту тему, мы собрали в этой главе материалы по основам алгоритмизации и программирования, а также сделали краткий экскурс в объектно ориентированную парадигму разработки, подкрепленный примерами диаграмм UML.

Глава пятая повествует о парадигме проектирования интерфейсов, делающей упор на пользовательский опыт, и кратко рассказывает о паттернах поведения и взаимодействия, важности их применения в процессе создания интерфейсов игр. Кроме того, глава познакомит вас с UMG — инструментом проектирования пользовательских интерфейсов движка UE.

В шестой главе центральной темой является искусственный интеллект (ИИ). Здесь обсуждаются общие вопросы применения ИИ в играх и особенности его реализации посредством встроенных средств движка UE.

В седьмой главе вы познакомитесь с понятием анимации и особенностями ее использования в игровой разработке, узнаете о секвенсоре (Sequencer) и о том, как реализованы анимации в UE.

Восьмая глава представляет собой небольшое введение в нюансы перехода на C++ проекты в UE.

В заключение изложена информация о перспективных направлениях развития движка.

ВВЕДЕНИЕ

Игровая индустрия

Мы назвали этот раздел «Игровая индустрия», подчеркнув природу появления UE. Большинство примеров и процессов, описанных в этой книге, будут относиться именно к играм. При этом фактор универсальности UE не позволит нам обойти вниманием и другие сферы применения движка.

Сообщество и события

Внутри любой серьезной индустрии всегда существует довольно много разнообразных сообществ. Игровая индустрия, которую нам с вами повезло застать в период взросления, тоже не является исключением. Специалисты различных направлений (художники, программисты, продюсеры, аналитики, гейм-дизайнеры и многие другие) регулярно встречаются на бизнес-конференциях как регионального, так и международного уровня. Самыми крупными событиями такого рода, конечно, являются ежегодные знаменитые Выставка электронных достижений (Electronic Entertainment Expo, E3) и Конференция разработчиков игр (Game Developers Conference, GDC) (<http://www.gdconf.com>), на которых презентуются инновационные технологии, а также демонстрируются самые свежие игры и их концепты.

Но чем могут быть полезны бизнес-конференции новичку? Важно понимать, что успех в делах зависит не только от ваших навыков, но и от актуальности профессиональных знаний, умения общаться и работать в команде, «ощущения» окружающего вас мира. Посещение индустриальных конференций и новые знакомства позволяют определиться и понять, хотите ли вы, чтобы этот мир стал вашим, готовы ли вы ко взаимодействию с людьми, работающими в этой области, к решению ее задач, графику работы и т. п. На мероприятия, проходящие в России и СНГ, например White Nights Conference for Gaming Industry (<http://wnconf.com/en>), можно зарегистрироваться волонтером и провести несколько дней в закулисе важного игрового события в качестве рядового участника конференции.

Современные технологии позволяют выстраивать сообщества и в виртуальном мире. Здесь нет границ, часовых поясов и необходимости физически присутствовать в определенное время в конкретном месте. Одним из самых старых сетевых ресурсов является база знаний, состоящая из статей специалистов по различным аспектам

разработки игр, Gamasutra (<http://www.gamasutra.com>), и если повезет, вы еще успеете застать этот сайт в его старом виде, так как сейчас площадка претерпевает ребрендинг и переезд на новый домен Game developer (<https://www.gamedeveloper.com/>). В русскоязычном сегменте также имеются как узкопрофильные, так и разнообразно специализированные сообщества, знакомство с которыми мы предоставим читателю осуществить самостоятельно, но все же отметим важный в контексте этой книги ресурс UE4 Daily (<https://ue4daily.com/>), где можно познакомиться с русскоговорящим тематическим сообществом посредством проектов и публикаций, узнать об образовательных возможностях, связанных с UE, а также просто почитать новости мировой игровой индустрии.

Отдельного упоминания заслуживают гейм-джемы (game jam) — события, в которых участвуют и профессионалы, и любители, и просто сочувствующие. Цель джема — за ограниченное время создать игру на заданную тему. Получившиеся в результате игры и их прототипы чаще всего попадают на своего рода игровой файлообменник itch.io — открытый магазин приложений независимых разработчиков видеоигр. Конечно, требования к играм могут различаться от джема к джему, но неизменной остается базовая идея: люди приходят туда, чтобы ощутить дух и радость творчества. Здесь можно протестировать любую геймплейную и гейм-дизайнерскую гипотезу, стихийно создать команду и расширить круг знакомств.

Существуют исследования в области образования, показывающие, что такие мероприятия заметно влияют на процесс обучения разработке игр. Кроме того, в гейм-джеме можно не только участвовать, но и самостоятельно его организовать. Начните знакомство с джемами, например с Global Game Jam (<https://globalgamejam.org/>) или Ludum Dare (<https://ldjam.com/>).

Зрелым индустриям сопутствуют развитые научные направления. Область научной коммуникации подразумевает существование социальных связей, накопление и формирование нового знания и его распространение. Те ученые и лидеры технологических компаний, что имеют возможность содержать собственные R&D-подразделения (Research and Development), изучают разные аспекты игр посредством тестирования гипотез, разработки новых алгоритмов и создания специализированной аппаратуры. Исследователи в области компьютерной графики могут похвастаться существованием такой мировой конференции по данной тематике, как ACM SIGGRAPH; сфера изучения человеко-машинного взаимодействия представлена не менее известной ACM SIGCHI; гуманитарные исследования обсуждаются в рамках мероприятий по game studies. Также в настоящее время продолжается формирование профильных конференций и появляются научные журналы в области игровой индустрии.

материалы, ориентированные на Blueprints или C++ в UE4, будут актуальными и для UE5. Для студентов это означает, что они по-прежнему могут полагаться на широкий спектр учебных ресурсов, доступных в печатном виде и онлайн. Студентам, незнакомым с UE, может потребоваться дополнительная помощь при изучении более старого учебного материала. Если в процессе его изучения и адаптации к новой версии движка возникают трудности, вы всегда можете обратиться к предыдущим версиям UE4, которые доступны для загрузки через лаунчер Epic Games и Git.

Gameplay Framework

Gameplay Framework остается основой для разработки игр на UE. Все так же в силе такие концепции, как акторы, классы и интерфейсы, а также дизайн и принципы ООП в виде инкапсуляции и взаимодействия классов. А такие новые функции UE5, как модульный игровой процесс (Modular Gameplay) и реестры данных (Data Registries), дополнительно улучшают уже проверенные временем системы.

Синтаксис Blueprint не претерпел изменений и в UE5. Проекты, использующие Blueprint, не нуждаются в каких-либо модификациях, хотя производительность приложений может быть снижена. В этом случае разработчикам потребуется прибегнуть к оптимизации кода.

Структура проекта и контроль версий

В пятой версии движка базовая структура проектов Unreal сохранилась. Соответственно, не изменился и привычный набор практик и рабочих процессов с использованием систем контроля версий или других подходов для совместной работы над файлами. Улучшен процесс сохранения уровней на диск (система One File Per Actor), но основные концепции структуры папок (Content, Config, Binary и Source) остались прежними.

ГДЕ ИСКАТЬ УЧЕБНЫЕ МАТЕРИАЛЫ?

Компания Epic Games предоставляет множество качественных материалов для изучения UE5, которые помогут вам освоить новые инструменты движка.

Идущий вместе с движком демонстрационный проект Valley of the Ancient представляет собой короткий сегмент геймплея, иллюстрирующий новые возможности движка. Проект можно загрузить из маркетплейса, для запуска потребуется высокопроизводительный компьютер, работающий на ОС Windows (кроме того, потребуется ≈100 Гб свободного места на диске).

Как видно из рисунка 1, в окне всегда отображается командная строка (**cmd**), что упрощает и ускоряет работу с командным языком.

Чтобы разработчику было доступно больше экранного места для взаимодействия с областью просмотра, в пятой версии появилась возможность легко сворачивать и разворачивать **Content Browser** (комбинация клавиш **Ctrl+Spacebar** или клик по кнопке **Content Drawer**), а также прикреплять любую вкладку редактора к сворачиваемой боковой панели.

Новая система добавления элементов в избранное обеспечивает быстрый доступ к часто используемым свойствам на панели **Details**, а новая кнопка **Create**, расположенная на главной панели инструментов, позволяет легко размещать акторов на сцене.

В UE5 значительно улучшен и вместе с тем упрощен процесс создания проектов.

В UE5 облегчен доступ к бесплатным сторонним ассетам, предоставляемый плагином Bridge (**Content > Quixel Bridge**, опция активирована по умолчанию). Библиотека ассетов доступна при условии авторизации в учетной записи Epic Games (на нее должна быть зарегистрирована лицензия UE), соответственно, контент может быть использован только в Unreal Editor.

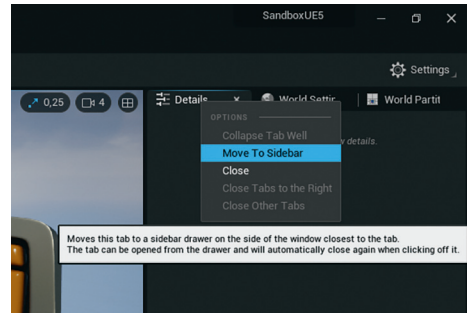


Рисунок 2. Контекстное меню вкладки, подсвечен пункт прикрепления ее к панели



Рисунок 3. Восстановить/закрыть свернутую вкладку

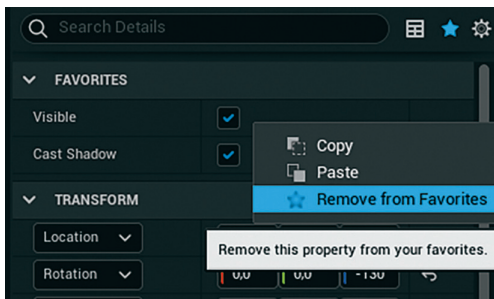


Рисунок 4. Пункт удаления элемента из избранного

В UE5 добавлена возможность автоматического преобразования типа в Blueprints. По умолчанию эта опция активна, но ее можно отключить в настройках редактора (**Edit > Editor Preferences > Content Editors > Blueprint Editor > Workflow > Enable Type Promotion**).

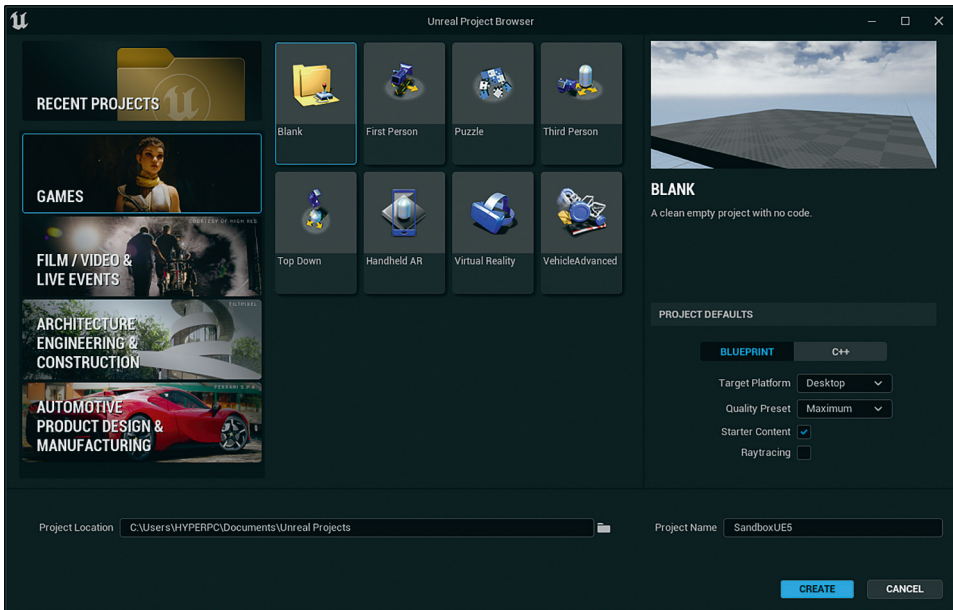


Рисунок 5. Новый интерфейс окна-проводника для работы с проектами

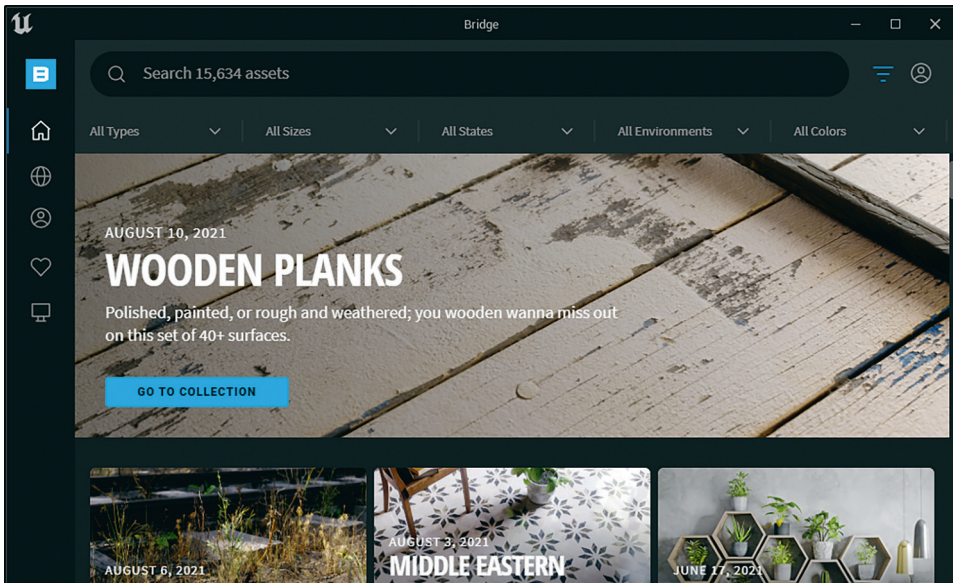


Рисунок 6. Окно плагина Bridge, открывающего доступ к библиотеке бесплатных ассетов

сбрасывать выделенные ячейки. Обратите внимание, что разрешение портирования доступно только на тех уровнях, которые были созданы после активации этой опции, а чтобы в панели World Partition отображалась корректная информация об уровне, он должен быть сохранен.

Отладка доступна с помощью консольной команды `wp.Runtime.ToggleDrawRuntimeHash2D`, она включает/отключает отрисовку карты с ячейками и загружаемыми в режиме реального времени объектами.

Благодаря новой системе One File Per Actor, которая разбивает каждый уровень на наборы файлов, группы разработчиков могут одновременно работать над игровым миром, не мешая друг другу. Это значит, что изменения, внесенные вами в уровень, будут сохраняться и отправляться только для тех акторов, которых вы редактировали или помещали на уровень.

Система One File Per Actor автоматически включается только в уровнях с разрешенной World Partition. Чтобы задействовать систему **совместной работы** в обычных уровнях для всех акторов, нужно в панели **World Settings** включить опцию **Use External Actors**. Кроме того, One File Per Actor можно включить и для отдельного актора, установив значение его параметра **Packaging Mode** в **External**.

Не обошли стороной изменения и в работе UE с системами контроля версий, в частности с Perforce, теперь разработчики могут выполнять почти все SCM-операции непосредственно в редакторе.

Использование рассмотренных новых инструментов движка позволяет значительно упростить процесс как командной разработки, так и персональной (например, для студентов, желающих попробовать себя в разработке игр с открытым миром).

ГЛАВА 1

ОСНОВЫ РАЗРАБОТКИ ИГР

отличие игры от реального мира — это «понарошковость» всего происходящего. Игрок всегда может переиграть!

Поскольку в этой книге мы будем обсуждать преимущественно компьютерные (видео) игры, давайте познакомимся с еще несколькими определениями этого явления, данными другими исследователями.

Николас Эспозито, руководитель отдела исследований в школе визуальной коммуникации и искусства Gobelins, в своем докладе¹, сделанном на конференции Digital Games Research Conference в 2005 году, кратко охарактеризовал понятие видеоигры как разновидности игры, основанной на некоторой истории, в которую можно сыграть благодаря аудиовизуальной аппаратуре. Существуют и активно обсуждаются и другие варианты определений, которые любопытный читатель может найти в статье Jonne Arjoranta «How to define Games and why we need to»².

Игры создаются людьми для людей, поэтому разработчику очень важно понимать, что представляют собой игроки как явление.

Концепция игрока характеризуется несколькими связанными между собой понятиями. Чтобы в дальнейшем их было легко различать и использовать, давайте разберем каждое из них. Прежде всего следует поговорить о **пользователях**, так как именно они запускают на своих компьютерах программы и выполняют в них различные действия для решения своих задач. Иногда пользователи даже играют в компьютерные игры. Именно в этот момент пользователь становится **игроком**, то есть выступает в социальной роли «игрока» в реальном мире. Далее мы иногда будем взаимозаменять термины «пользователь» и «игрок», но только там, где это не вызовет путаницы и двусмысленности.

Следующее понятие — **аватар**. Это представитель игрока в мире игры, социальная роль, в игре называемая **персонажем**, например можно быть темным или светлым магом, морским пехотинцем или ученым. В каждом из этих случаев вы будете управлять соответствующим персонажем.

КАК ИГРА ВОЗДЕЙСТВУЕТ НА ИГРОКА?

Интересно отметить, что среди прикладных программ цифровые развлечения и игры обладают уникальным свойством: в их задачи входят вовлечение и развлечение пользователя. Подобные требования не предъявляются к калькуляторам, средам разработки программного обеспечения или системам бизнес-аналитики, а вот все происходящее в игре должно удерживать внимание игрока, мотивируя его не бросать

¹ *Esposito, N.* A short and simple definition of what a videogame is: In DiGRA 2005 conference: Changing views-worlds in play. 2005 (<https://www.utc.fr/~nesposit/publications/esposito-2005definition.pdf>).

² <https://link.springer.com/article/10.1007/s40869-019-00080-6#ref-CR8>.

игровой процесс. Если же в процессе игры игрок заскучает, то, скорее всего, он переключится на другое, более интересное занятие, поэтому разработчики игр стремятся создавать такие виртуальные миры, которые способны надолго увлечь игрока, не надоедая ему.

Прикладные программы (application software, applications, apps) согласно IBM Research — это программное обеспечение, необходимое для решения или помощи в решении задач пользователей. К прикладным программам относятся офисные продукты, системы управления данными. Веб- и мобильные приложения также могут быть прикладными программами, например мобильные приложения торговых марок или супермаркетов, социальных сетей и т. п.

Системные программы (system software) согласно тому же portalу представляют собой набор функций (или целый комплекс программ) по управлению подсистемами компьютера. К системным программам относятся операционные системы, системы управления дисками и другими аппаратными средствами, утилиты и т. п.

Так что же заставляет игрока продолжать играть? Большинство «хитрых манипуляций» можно разделить на три большие группы: позитивное подкрепление, негативное подкрепление и принуждение. Позитивное подкрепление (positive reinforcement) означает вознаграждение игрока за его действия. Негативное (отрицательное) подкрепление (negative reinforcement), наоборот, предполагает устранение негативного воздействия на игрока в случае выполнения им желаемого авторами игры действия. Не следует путать негативное подкрепление с принуждением (punishment), смысл которого именно в непосредственно негативном воздействии на играющего, чтобы заставить его действовать определенным образом. Забавно, что согласно исследованиям принуждение — самый эффективный способ сделать так, чтобы игрок избегал определенных действий. Вспомните, например, необходимость перемещаться по лаве, кислоте или другим токсичным субстанциям, отнимающим здоровье. Повторять подобные действия и попадать в соответствующие среды лишний раз не хочется, у игрока появляется желание избегать их, ища обходные пути, но «успешный» прорыв через опасные препятствия воспринимается как преодоление и маленькая победа. Если же цель — добиться от игрока постоянного повторения каких-либо действий, в этом случае стоит использовать положительное подкрепление. Игры жанра «три в ряд» не скупятся на похвалу за каждые три убранного объекта, даже если игрок проходит уже пятисотый по счету уровень.

В битве за удержание внимания игрока часто используют **постепенное открытие доступа к контенту**, это может быть любое содержимое игры, с которым сталкивается игрок. Если контент доступен сразу и полностью, он подобен прочитанной книге,