

5. Пат. 2190462 РФ, МКИ 7 В 01 F 7/28. Роторно-пульсационный аппарат / Иванец В.Н., Иванец Г.Е., Афанасьева М.М., Сафонова Е.А., Артемасов В.В.// Оpubл. 10.10.2002. - Бюл. №28.

6. Баланов, П.Е. Технология броидильных производств: учебно-методическое пособие / Баланов П.Е. – СПб: НИУ ИТМО; ИХиБТ, 2013. -65 с.

Borodulin Dmitriy Mihaylovich, Doctor of Engineering Sciences, professor
(e-mail: borodulin_dmitri@list.ru)

Safonova Elena Aleksandrovna, Candidate of Engineering Sciences, docent
(e-mail: safonova.kem@yandex.ru)

Prosin Maksim Valerevich, Candidate of Engineering Sciences, teacher
(e-mail: santa-dlx.kem@yandex.ru)

Milenkiy Ilya Olegovich, graduate student
(e-mail: milenkiyilia@mail.ru)

Kemerovo Institute of Food Science and Technology (University), Kemerovo, Russia

STUDY OF THE PROCESS OF HOPPING OF WORT SYRUP USING A MODERN EQUIPMENT

Abstract. The present article is devoted to the study of hopping of wort syrup in the brewing industry using a rotary-pulsation apparatus. A hop extract is prepared in this apparatus, which is then added to the wort syrup during boiling. The optimal technical characteristics of apparatus are determined for maximum possible content of isohumulone in the prepared hop extract. Isohumulone is the main component of bitter substances. Using a rotary-pulsation apparatus at the stage of hopping of wort syrup allows reducing the dose of granulated hop and the boiling process from 2 to 1 hour.

Keywords: beer, rotary pulsation apparatus, hopping, wort syrup, isohumulon, hop extract.

УДК 004.8:004.946.005.336

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ В КОМПЬЮТЕРНЫХ ИГРАХ. ПЕРСПЕКТИВЫ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ИГРОВОЙ ИНДУСТРИИ

Буковшин Вадим Александрович, студент

Донской государственной технической университет,
г.Ростов-на-Дону, Россия

(e-mail: vadimbukovshin199813@gmail.com, Luck1w1nner@outlook.com)

Воскобойников Сергей Георгиевич, к.и.н., доцент

Донской государственной технической университет,
г.Ростов-на-Дону, Россия

(e-mail: voskoboynikof1968c@yandex.ru)

В данной статье рассматриваются особенности технической реализации искусственного интеллекта в играх, основные методы создания интеллектуальных систем, предлагаются варианты их улучшения, а также описываются перспективы развития данной области информационных технологий.

Ключевые слова: компьютерная игра, игровая индустрия, область интерактивных развлечений, игровой искусственный интеллект (ИИИ), ис-

кусственный интеллект (ИИ), интеллектуальная игровая система, внутриигровые персонажи, несобственные игровые персонажи, собственные игровые персонажи, игровой мир, скриптинг, управление перемещением, управление принятием решений, деревья решений, поведенческие деревья, конечные автоматы, системы планирования, цепи Маркова, нейронные сети.

Введение

Мировой рынок медиапотребления стремительно меняется, телевидение всячески вытесняется увлекательными интернет-технологиями и компьютерными играми. По статистическим данным число игроков в компьютерные игры на территории России составляет примерно 46 миллионов человек [7], а продажи видеоигр тем временем затмевают суммарные сборы какого-нибудь всеми ожидаемого блокбастера в кинотеатрах. И правда, за последние 20-30 лет игры кардинально изменились, пройдя путь от смехотворных игровых автоматов в магазинах до сложных компьютерных систем, вычислительной мощности которых зачастую не хватает для системных требований игры. Рост технологических возможностей и интереса к игровой индустрии делает область интерактивных развлечений самой быстро развивающейся областью информационных технологий в мире. За прошедшие шесть лет суммарные продажи игровых консолей текущего поколения, таких известных устройств, как Playstaion 4, XBOX ONE и Nintendo Switch, составили приблизительно 130 млн. копий [6].

Но компьютерные игры не всегда были такими успешными и популярными. Игровая индустрия изменилась с приходом персонального компьютера и возможности сделать что-то большее, чем игровой автомат на улице или безамбиционную игрушку в руках у ребенка. Знаменательным событием для игровой индустрии стал выход таких игр, как: Black&White (Lionhead Studios Inc., 2001), Creatures (Cyberlife Technology Inc., 1996), Goldeneye (Rare Inc., 1997) и Halo (Bungie Studios Inc., 2001), которые закладывают основы для появления целых игровых жанров и определяют дальнейшее направление развития компьютерных игр.

Можно сказать, что на данный момент компьютерные игры – не просто фрагменты кода, а удачно аккумулялированные методы выразительности и передачи информации, совокупность визуальной составляющей, которая представлена довольно развитыми технологиями компьютерной графики и визуализации анимационных эффектов, музыкального сопровождения и применения средств искусственного интеллекта.

Да, игровой искусственный интеллект (далее ИИИ) стал одним из основных средств выразительности в компьютерных играх и зачастую определяет уровень качества конечного продукта разработчиков. Как отмечает Алекс Шампандар в своей книге «Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия» [1, с. 22]: «Средства искусственного интеллекта стали неотъ-

емлемой частью любой компьютерной игры». Данная статья акцентирует внимание на том, почему же ИИИ стал так важен для игровой индустрии.

Цель: охарактеризовать основные технические методики создания искусственных интеллектуальных систем в играх и принципы их функционирования.

Задачи:

- Дать понятие ИИИ.
- Разобрать основные особенности и задачи ИИИ.
- Описать технические методы современного ИИИ и его применение в компьютерных играх.
- Рассказать о вероятных перспективах и будущих направлениях развития ИИИ.

Подходы к определению понятия «Игровой искусственный интеллект»

С целью исключить возможность разночтения и неправильного понимания предмета рассмотрения данной статьи, обозначим понятие «Игровой искусственный интеллект» и области его определения. Стоит отметить, что разработка игр – отнюдь не наука, программные методы, которые реализуют разработчики не стандартизированы, те или иные алгоритмы, применяющиеся в разработке компьютерных игр не формализованы, поэтому ИИИ не имеет строгого, формального определения.

ИИИ несколько отличается от привычного академического понимания искусственного интеллекта. Его основной целью и по сей день остается создать видимость интеллектуальности внутриигровых персонажей, естественности их поведения, реакций и адекватности их намерений. ИИИ довольно часто использует некоторые отрасли знаний искусственного интеллекта как науки в целом, включая разнообразные алгоритмы поиска оптимального маршрута, алгоритмы управления и принятия решений, о которых будет рассказано чуть подробнее в следующих главах этой статьи, однако игровая индустрия также может похвастать и довольно сложными взаимосвязанными интеллектуальными системами, которые и выделяют ИИИ в совершенно самостоятельную область знаний.

ИИИ часто описывают с помощью программных методик и другого технического инструментария, используемых разработчиками при написании компьютерной игры с целью создания так называемых интеллектуальных игровых систем, предназначенных, в первую очередь, для того, чтобы предоставить игроку интересный игровой процесс, ощущения и впечатления от игры.

Задачи ИИИ

На практике назначение интеллектуальной системы сводится к управлению поведенческими особенностями внутриигровых персонажей.

Говоря о ИИИ, можно с высокой степенью объективности отметить сле-

дующие базовые принципы, которые помогают игровым системам казаться более интеллектуальными и естественными по отношению к игроку. Итак, «хороший искусственный интеллект»:

- предоставляет игроку больше возможностей, чем тому необходимо;
- говорит игроку о своих намерениях;
- имеет предсказуемое поведение;
- может взаимодействовать с игровыми системами;
- реагирует на действия игрока;
- имеет свои цели.

Так или иначе, необходимо привести очевидный пример, чтобы показать, выдерживают ли принципы «хорошего» ИИИ применение к частному случаю. Таким примером станет серия приключенческих компьютерных игр, основанных на комиксах о Бэтмене, темном рыцаре, защитнике и покровителе своего родного Готэм-сити, *Batman: Arkham series*, разработкой которой занимались *Rocksteady Studios* и *Warner Bros. Games Montreal*. В играх серии всегда присутствовал отличный сюжет с элементами детектива, достаточно неплохая боевая система, значительную роль в которой играл искусственный интеллект неигровых персонажей, заключенных тюрьмы *Blackgate*, которые всячески останавливали главного героя. Сфокусируемся на скрытой (англ. *Stealth*) составляющей серии, прохождению ограниченных пространств с преодолением группы вооруженных противников. Такие моменты прекрасно демонстрируют взаимодействие игрока с ИИИ и достаточную вариативность действий, которые может предпринять игрок в определенных обстоятельствах. Противники разбиты по нескольким, определенным разработчиками, зонам и патрулируют их различные области, вашей же целью остается обезвредить эту группу противников любыми доступными способами. По мере того, как вы один за другим будете выводить их из строя, оставшиеся персонажи проверят последнюю позицию обезвреженного вами противника, а увидев своего товарища без сознания, непременно отреагируют, поднимут тревогу и проинформируют остальных – видим отчетливые признаки того, что ИИИ реагирует на действия игрока. Теперь противники находятся в встревоженном состоянии, их цель проверить каждый уголок, каждый уступ, каждый метр вентиляции, в которых вы могли бы укрыться: теперь они будут заходить в различные комнаты, поднимать крышки вентиляций и прикрывать друг друга – на лицо взаимодействие ИИИ с остальными частями игровой системы. Все это время противники переговариваются между собой, используя рацию, вы же, прослушивая их частоту, можете узнать о намерениях противника и спланировать свои дальнейшие действия – в этом смысле ИИИ постоянно информирует игрока, становится более предсказуемым, дает ему большую вариативность и свободу действий. Но не стоит забывать и об условностях, которые допустили разработчики, например, о том, что противники волшебным образом не оборачиваются, когда вы ре-

шили подкрасться к ним сзади и обезвредить их – вот, где игроку и предоставляют больше возможностей, именно те моменты, отсутствие которых могло негативно сказаться на впечатлениях от игры. Все эти особенности позволяют действительно погрузиться в игровой процесс и испытать удовольствие от взаимодействия с ИИИ. Очевидно, что частный случай не может дать нам гарантию выполнения всех этих принципов в каждой компьютерной игре, но можно с уверенностью сказать, что, если ИИИ всегда бросает вам все новые и новые вызовы, преподносит более изощренные испытания, он несомненно будет отвечать требованиям, описанным выше.

Так или иначе, все эти принципы могут быть сведены к следующей базовой модели ИИИ, приведенной на рисунке ниже:

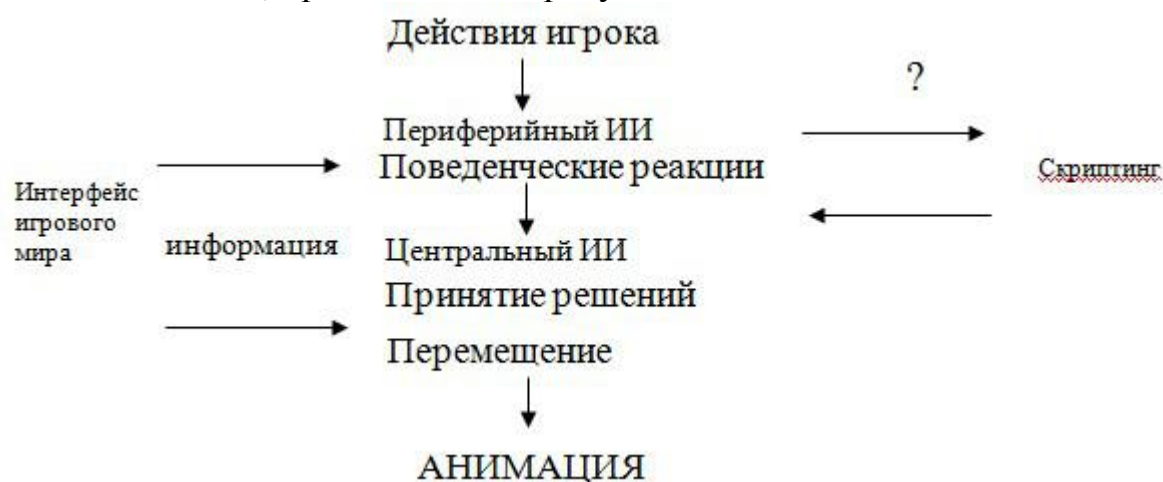


Рисунок 1 - Базовая модель ИИИ в типичной компьютерной игре

ИИИ делится на периферийный и центральный ИИ. Необходимость в этом очевидна. Во-первых, игра должна реагировать не только на действия игрока, но и на изменения в виртуальном мире, за что отвечает периферийный ИИ. Во-вторых, в зависимости от действий игрока или обстоятельств и состояния виртуального мира персонажи должны принимать определенные решения, за что отвечает центральный ИИ. В-третьих, все эти изменения должны иметь свое место на экране в виде визуальных анимаций, озвучивания и т.п. И наконец, при необходимости многие из этих аспектов могут быть жестко определены («заскриптованы») заранее разработчиком. Другими словами, ИИИ должен обязательно выполнять три основные задачи:

- определять перемещение персонажей;
- управлять принятием собственных решений персонажей;
- отвечать за поведенческие особенности персонажей;

Хотелось бы отметить, что распределение ИИИ, предложенное выше, имеет определенную степень условности, учитывая то, что, например, моделирование перемещений требует от несобственных персонажей в равной степени как принятия определенных решений по выбору точки назначения

или варианта маршрута, так и изменений в поведении. Однако данная модель полезна в демонстрации основных особенностей работы ИИИ в компьютерных играх.

На данный момент преимущественное большинство компьютерных игр требует внедрения всех областей ИИИ, которые будут описаны далее.

Реализация перемещений

Любая область ИИИ в равной степени важна для компьютерной игры и требовательна к вниманию разработчика, ведь любая степень интеллектуальности игровой системы может разрушиться, если хотя бы один компонент системы недоработан. Перемещение – один из основных компонентов игровой механики, техническая реализация которого требует больших усилий разработчиков и отнимает много времени.

Перейдем к особенностям моделирования перемещения персонажей в современных компьютерных играх. Сперва отметим, что все игровые особенности, будь то механики перемещения, взаимодействия персонажей или их поведенческие особенности напрямую зависят от окружающего виртуального мира, который заранее придумывается разработчиком, прорабатывается художниками и реализуется программистами. Поэтому можно выделить следующий базис, от которого напрямую зависит модель перемещений:

- структура игрового мира;
- реализация игрового мира;
- методы управления перемещением.

Структура игрового мира – это часть игровой окружающей среды, которая может оказывать физическое воздействие на перемещение. Другими словами, это все объекты игровой системы, которые могут либо изменить направление перемещения персонажа, либо повлиять на скорость его перемещения.

Реализация игрового мира - представление информации об окружающей среде. В большинстве компьютерных игр мир представлен непрерывной областью с определенной степенью дискретизации, которая вводится с целью упростить математические выкладки и уменьшить вычислительную погрешность при обработке данных.

Методы управления перемещением делятся на высокоуровневые управляющие действия ИИИ и низкоуровневые анимационные примитивы. Для лучшего распределения вычислительных ресурсов разработчики часто комбинируют эти приемы, что позволяет использовать некоторые уже готовые технические наработки визуализации анимационных эффектов ходьбы, бега и других моторных функций игрового персонажа и не растрачивать впустую время ожидания для управляющих команд системы.

Как уже было указано выше, игровое перемещение, причем как собственных, так и несобственных игровых персонажей, напрямую зависит от структуры виртуального мира. Перемещение собственных игровых персо-

нажей зависит непосредственно от того, как игроку визуализируют устройство мира, его размерность, наличие уровней, виды объектов игровой системы и прочее. Остальные внутриигровые персонажи не ощущают игровой мир так, как его ощущают люди, ориентируясь в пространстве только за счет данных об окружающей среде. Поэтому перемещение несобственных игровых персонажей всегда зависит от проблемы представления данных о структуре мира искусственному интеллекту, которая решается двумя способами: либо информация об окружающей среде задается ИИИ перед началом игрового процесса, либо периодически обновляется.

Теперь становится очевидным, что перемещение зависит от структуры, ведь оно зависит от типа игрового мира: если он статичен, положение объектов в нем не меняется со временем, значит не составит труда сообщить об этом искусственному интеллекту один единственный раз перед началом игры, если же наоборот, мир динамичен, некоторые его объекты постоянно меняют свое местоположение, тогда приходится обновлять информацию о состоянии системы с целью избежать непредвиденных ситуаций. Следовательно, игровая система будет считаться информированной, то есть, ей будут доступны сведения о расположении игровых объектов, и теперь не составит труда найти оптимальный путь от начальной точки до точки назначения. Это также полностью исключает ошибки программного кода, в результате которых мы можем увидеть, как игровые персонажи буквально застревают в игровых объектах, ходят по воздуху и т.п.

Хочется сказать, что семейство алгоритмов нахождения оптимального пути было значительно расширено в рамках науки «Искусственный интеллект», а в рамках ИИИ стало технически совершенным. В большинстве компьютерных игр используется так называемый алгоритм A^* (англ. *A star*), который условно является модификацией алгоритма Дейкстры и алгоритма поиска в ширину и находит маршрут наименьшей стоимости от начальной вершины до выбранной конечной.

Реализация принятия решений

Любое изменение состояния игровой системы будет требовать определенного множества решений от определенного множества игровых персонажей, поэтому разработка простых и эффективных моделей управления принятием решений позволяет сделать игровой процесс более органичным и естественным. На данный момент в компьютерных играх используется довольно много подобных алгоритмов, включая реализацию деревьев решений (англ. *Decision Trees*), конечных автоматов (англ. *Finite State Machine*) и систем планирования (англ. *Goal-Oriented behaviour*), каждый из которых будет рассмотрен далее.

Деревья решений

Деревья решений – один из самых быстрых и легко реализуемых алгоритмов принятия решений в компьютерных играх, широко использующийся для управления персонажами. Деревья решений очень похожи на би-

нарные деревья поиска, которые являются их частным случаем, с учетом того, что их узлы являются некоторыми условиями. Узлы данного дерева могут иметь несколько потомков, каждый из которых, в свою очередь, может иметь целое поддерево состояний системы. Задача узлов дерева – перевести игровую систему в конечное состояние.

Возвращаясь к серии игр о темном рыцаре (Batman: Arkham series), взаимодействие с ИИИ которых мы рассматривали ранее, ИИИ вооруженных противников мог быть реализован с помощью деревьев решений:



Рисунок 2 - Пример реализации дерева решений

Отличительной особенностью такого подхода является то, что система мгновенным образом меняет свое состояние при прохождении определенного узла, а основным преимуществом является то, что в игровые системы с конечным множеством состояний можно с легкостью внедрить функции их перехода и описать целое множество игровых ситуаций. Недостатки данного метода заключаются в невозможности интеграции новых игровых связей в уже готовое дерево и возможная коллизия двух деревьев решений при пересечении областей их действия.

Дерево решений, приведенное на рисунке 2, довольно упрощено и описывает только один из целого множества игровых моментов. В крупных компьютерных играх, где реализован данный подход, может быть интегрировано до сотни деревьев подобного типа, которые будут иметь большую степень вложенности и разветвленности, - это сделано для того, чтобы описать все множество игровых ситуаций со всеми возможными исходами.

Выбор дерева будет определяться состоянием системы в данный момент времени, от которого будет зависеть вид корневого узла.

Большая скорость реализации и эффективное применение делают метод деревьев решений удобным инструментом для реализации моделей принятия решений для ИИИ, который и сейчас используется во многих компьютерных играх.

Конечные автоматы

Конечный автомат – еще одна модель управления принятия решений, реализуемая в компьютерных играх для систем с конечным множеством состояний. Чтобы определить конечный автомат необходимо задать конечное множество состояний игровой системы и набор переходов из одних состояний в другие. Реализуется данный подход с помощью графа по причине того, что с увеличением числа состояний не возникает проблем с его расширением и общей производительностью системы.

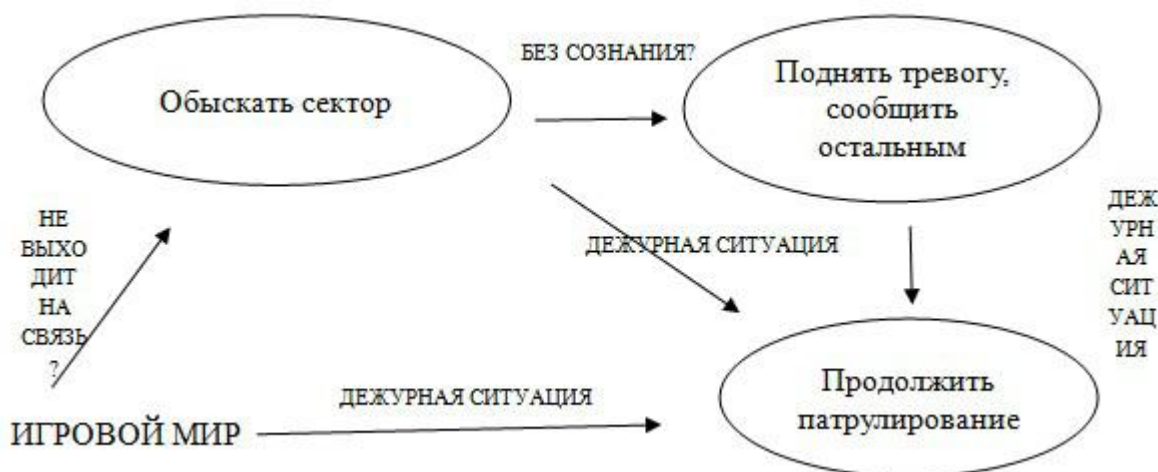


Рисунок 3 - Пример реализации конечного автомата

Теперь игровая ситуация, описанная выше, реализована с помощью конечного автомата. В данном случае узлы графа – состояния игровой системы, а его ребра – переходы из одних состояний в другие.

Для наглядности определим конечное множество A_{batman} состояний игровой системы, конечное множество переходов F_{batman} и ее начальное состояние $Q_{0batman}$:

$A_{batman} = \{ \text{«Продолжить патрулирование» (нормальное, пассивное состояние системы), «Обыскать сектор» (состояние системы, спровоцированное подозрительной игровой активностью), «Поднять тревогу, сообщить остальным» («агрессивное, активное состояние системы») };$

$F_{batman} = \{ \text{«Товарищ не выходит на связь», «Товарищ без сознания», «Отсутствие угрозы, дежурная ситуация» };$

$Q_{0batman} = \text{«Дежурная ситуация» (состояние покоя).}$

Таким образом, конечный автомат K_{batman} для данной ситуации может выглядеть следующим образом:

$$K_{\text{batman}} = \{A_{\text{batman}}, F_{\text{batman}}, Q_{\text{obatman}}\}.$$

Из преимуществ можно отметить простоту реализации и постоянное усовершенствование метода на основе формальных определений теории конечных автоматов, из недостатков – серьезное усложнение структуры конечных автоматов при проектировании систем с большим числом состояний.

Без сомнений можно сказать, что две модели управления принятием решений, описанные выше, а именно деревья решений и конечные автоматы достаточно похожи друг на друга, поэтому разработчики интеллектуальных систем часто комбинируют их, создавая более универсальный и стабильный к непредсказуемым последствиям искусственный интеллект. Такие системы применяются в играх типа FPS (англ. *First Person Shooters*), таких как Halo (Bungie Studios Inc., 2001), F.E.A.R. (Monolith Productions Inc., 2005), различных приключенческих играх (англ. *Action-games*), а именно Batman: Arkham series (Rocksteady Studios/ Warner Bros. Games Montreal, 2009-2015) реже в RPG (англ. *Role-Play games*) и стратегиях реального времени (англ. *Real-Time strategy*). Перечисленные игры стали примером удачного внедрения нового подхода к понятию интеллектуальных игровых систем и их удачной реализации, что и сделало их довольно известными в свое время.

Системы планирования

Настоящая статья была бы неполной, если бы в ней не упоминались довольно амбициозные системы планирования (англ. *Goal-Oriented behaviour*), признаки которых стали появляться в некоторых компьютерных играх 2005-2010 гг. Данный подход базируется на очень простой идее – сделать поведение персонажей более естественными, определив их цели, мотивацию и соответствующие действия.

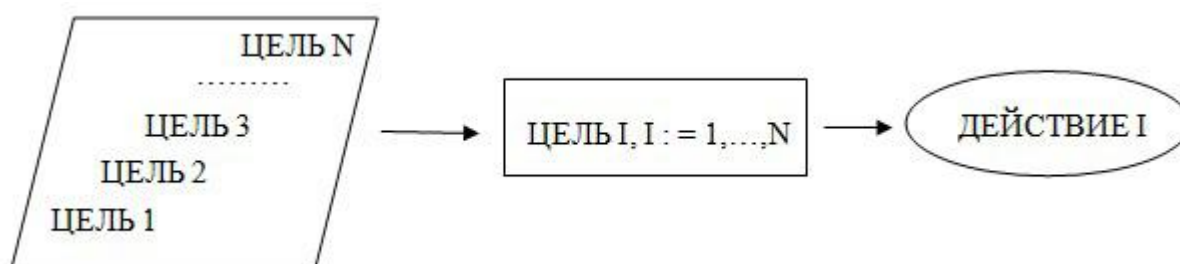


Рисунок 4 - Схематичное изображение системы планирования

На рисунке 4 схематично изображены принципы работы систем планирования. При таком подходе каждый несобственный игровой персонаж будет иметь цель или целое множество целей, которые будут ранжироваться по системе приоритета, то есть, если в данный момент цель i имеет больший приоритет, чем остальные цели, значит первоначальной задачей персонажа будет удовлетворить цель i соответствующим действием i . Данный подход довольно трудоемок в определении всех множеств целей и сложен

в реализации, поэтому он не так распространен, как те же поведенческие деревья или деревья решений, однако в игровой индустрии имеется несколько удачных примеров реализации этого алгоритма, один из которых – знаменитая A-Life (англ. *Artificial Life*), система симуляции жизни в компьютерной игре по мотивам повести братьев Стругацких «Пикник на обочине» под названием S.T.A.L.K.E.R. (GSC Game World, 2002), которая заслуживает отдельного внимания.

По словам самого Дмитрия Ясенева, ведущего программиста проекта [5], «Суть её заключается в том, что персонажи в игре живут своей жизнью и существуют всё время, а не только когда их видит игрок. Это идёт вразрез с привычными оптимизациями, используемыми при разработке игр...».

Искусственный интеллект наряду с логичными моделями перемещения и естественными анимациями – неотъемлемая часть этого игрового мира. Ключевой особенностью технической реализации разработчиков из GSC Game World стала комбинация средств ИИИ и периодического «скриптинга» программного кода в связи с особенностями сюжета. ИИИ делится на интеллект несобственных персонажей, сталкеров, и интеллект Зоны, который представляет собой интеллект агрессивных живых существ, монстров. Однако, если последний представлен низкоуровневыми анимационными примитивами, которые были описаны ранее, ведь все поведение таких существ описывается с помощью непрерывных анимаций, то интеллект сталкеров имеет особую ценность. Решения этих персонажей определяются совокупным действием нескольких десятков операторов, каждый из которых имеет свое множество целей и свою систему приоритетов. В зависимости от обстановки окружающего мира среди всех операторов целей выбирается один и дальнейшие поведенческие особенности персонажа определяются именно его работой. Таким образом, из-за того, что разные персонажи в одни и те же моменты времени имеют различные цели, создается эффект реальности происходящего, а в совокупности с неплохим игровым дизайном, детализацией объектов и музыкальным сопровождением игроки проводят за игрой сотни часов. И это только часть титанической работы, которую проделали специалисты по ИИИ из GSC Game World.

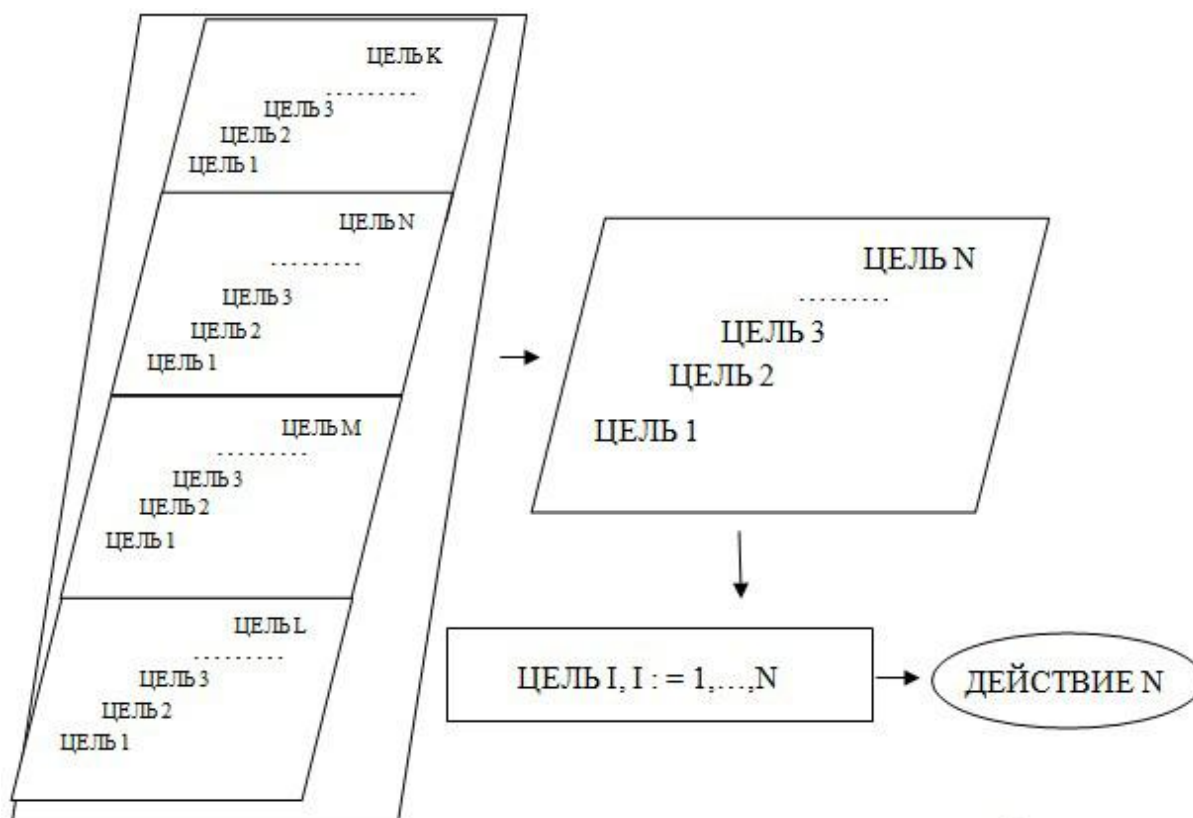


Рисунок 5 - ИИИ несобственных игровых персонажей игры S.T.A.L.K.E.R.

И пусть от некоторых моментов игрового процесса пришлось отказаться на старте проекта, пусть некоторые игровые механики работали не так стабильно, как задумывалось разработчиками, эти идеи вдохновили многих программистов игровой индустрии заниматься усовершенствованием систем планирования и подобных алгоритмов, а также их реализацией в будущем.

Цепи Маркова и перспективы развития ИИИ в компьютерных играх

Все это разнообразие алгоритмов делает ИИИ довольно широкой областью информационных технологий с развитой и почти совершенной методологией. Однако, как можно было заметить, все представленные ранее алгоритмы, хоть и довольно хороши в реализации и позволяют создавать действительно потрясающие интеллектуальные системы, которые мы наблюдаем в современных компьютерных играх, делают искусственный интеллект довольно предсказуемым. Это не всегда плохо, как мы увидели на примере игр Batman: Arkham series, но в конечном итоге игрок попросту предсказывает действия несобственных персонажей и теряет интерес к игровому процессу.

С целью избежать таких последствий многие разработчики работают над внедрением новой довольно экспериментальной концепции с довольно объемным и сложным математическим аппаратом – цепями Маркова.

Для тех, кто еще не знаком с курсом теории вероятностей, цепью Маркова называется последовательность испытаний, в каждом из которых происходит, и притом единственное, из несовместных событий S_1, \dots, S_N . На деле мы всегда имеем конечное число состояний системы, причем в каждый момент времени система может принимать только одно из этого множества состояний, и таблица вероятностей, определяющих все возможные переходы системы из текущего состояния S_i в одно из несовместных состояний $S_1, \dots, S_N, i := 1, \dots, N$.

S	S	S_1	S_2	S_3	S_N
S_1		P_{11}	P_{12}	P_{13}	P_{1N}
S_2		P_{21}	P_{22}	P_{23}	P_{2N}
S_3		P_{31}	P_{32}	P_{33}	P_{3N}
.....	
S_N		P_{N1}	P_{N2}	P_{N3}	P_{NN}

Рисунок 5 - Таблица 1 переходов

Два события называются несовместными, если они не могут произойти одновременно в рамках данного опыта. И это только небольшая часть теории цепей Маркова, которая успешно может быть применена в создании ИИИ. Но на сегодняшний день игровая индустрия не может похвастаться ни одной компьютерной игрой с ИИИ, основанном на цепях Маркова. В данной статье будет предложена одна из бесконечного множества игровых ситуаций, которые могут быть обыграны с использованием данного подхода.

Допустим, мы имеем FPS (англ. *First Person Shooter*) и вооруженного противника, у которого в запасе есть две гранаты и с десяток патронов. Так как известно, что осколочная граната с большей вероятностью сможет поразить игрока, несобственный игровой персонаж однозначно решает бросить гранату, однако возникает вопрос: «Хватит ли двух гранат, чтобы полностью поразить игрока?». Оговорим условности: считаем, что модель прицеливания несобственных персонажей уже написана разработчиком, априорные вероятности попадания игрока в область поражения гранаты – просчитанными.

Определим множество состояний системы, то есть, самого игрока:

S_1 – «Игрок невредим»;

S_2 – «Игрок получил легкие повреждения»;

S_3 – «Игрок получил тяжелые повреждения»;

S_4 – «Игрок полностью уничтожен».

Повреждения, наносимые осколками гранаты, зависят от ряда условий, которые будем считать случайными; “удачное” попадание способно полностью уничтожить даже совершенно невредимую цель. Промах же, соответственно, никак не изменит состояние цели.

Запишем таблицу переходов, вероятности которой заполнены в соответствии с точностью броска, динамикой цели и множеством других игровых обстоятельств.

S \ S	S ₁	S ₂	S ₃	S ₄
S ₁	0,1	0,4	0,3	0,2
S ₂	0	0,1	0,2	0,7
S ₃	0	0	0	1
S ₄	0	0	0	1

Рисунок 6 - Таблица 2 переходов

Пусть до момента броска игрок был невредим, то есть находился в состоянии S₁. После первого броска это, очевидно, изменится, причем с вероятностями, заданными в первой строке данной таблицы, стало быть:

$$P_1(1) = 0,1; P_2(1) = 0,4; P_3(1) = 0,3; P_4(1) = 0,2;$$

После второго броска игрок может находиться в одном из четырех вышеописанных состояний, например, иметь тяжелые повреждения, что могло произойти следующими способами:

- первый бросок – промах, второй – тяжелые повреждения;
- первый бросок – легкие повреждения, второй – тяжелые повреждения;
- первый бросок – тяжелые повреждения, второй – промах.

Для того, чтобы найти вероятность P₃(2), нам необходимо воспользоваться формулой полной вероятности:

$$P_3(2) = P_1(1) \cdot P_{13} + P_2(1) \cdot P_{23} + P_3(1) \cdot P_{33} = P_1(1) \cdot P_{13} = 0,1 \cdot 0,3 + 0,4 \cdot 0,2 = 0,03 + 0,08 = 0,11.$$

Вычислим оставшиеся вероятности:

$$P_1(2) = P_1(1) \cdot P_{11} = 0,1 \cdot 0,1 = 0,01;$$

$$P_2(2) = P_1(1) \cdot P_{12} + P_2(1) \cdot P_{21} = 0,1 \cdot 0,4 + 0,4 \cdot 0,1 = 0,04 \cdot 2 = 0,08;$$

$$P_4(2) = P_1(1) \cdot P_{14} + P_2(1) \cdot P_{24} + P_3(1) \cdot P_{34} + P_4(1) \cdot P_{44} = 0,1 \cdot 0,2 + 0,4 \cdot 0,7 + 0,3 \cdot 1 + 0,2 \cdot 1 = 0,02 + 0,28 + 0,3 + 0,2 = 0,8.$$

$$P_1(2) + P_2(2) + P_3(2) + P_4(2) = 0,01 + 0,08 + 0,11 + 0,8 = 0,2 + 0,8 = 1.$$

Сумма всех вероятностей должна равняться единице, что отвечает требованию нормированности и означает, что нет неучтенных вероятностей. Судя по полученным данным, можно с большой уверенностью сказать, что после повреждений, полученных от разрыва двух осколочных гранат, игрок будет полностью уничтожен, т.к. остальные вероятности ничтожно малы, поэтому несобственный персонаж игровой системы не будет тратить оставшиеся патроны впустую, а незамедлительно бросит в игрока две осколочные гранаты.

Заметим еще раз, что это только единственный, притом, один из самых тривиальных, вариант развития событий, которых может быть сотни тысяч. Например, игровые персонажи могут высчитывать такие вероятности

поражения игрока в зависимости от вооружения, которое имеется у них в данный момент времени, и планировать свои дальнейшие действия; основываясь на данных о последнем местонахождении игрока, предсказывать его наиболее вероятный маршрут и многое другое. Очевидно, что с таким подходом, поведение внутриигровых персонажей станет более живым, естественным и с каждым разом будет предоставлять какой-то новый игровой опыт.

Еще одна интересная концепция, которая еще не имеет достаточного внимания у разработчиков игровой индустрии – обучающийся ИИИ. Это довольно сложная в реализации технология, основанная на так называемых нейронных сетях, помогающая в создании игровых персонажей, которые в последствии смогут запоминать информацию, классифицировать ее и на основе полученных данных неким образом эволюционировать.

Однако некоторые игровые студии поступают довольно хитрым путем, используя уже имеющиеся алгоритмы с некоторым уловками в программном коде для создания иллюзии обучения персонажей. Отличным примером такого подхода является компьютерная игра *Alien: Isolation* (Creative Assembly, 2014), игровой мир которой основан на фильме «Чужой» 1979 года. Реализация ИИИ в этой игре технически находится на очень высоком уровне, несмотря на то, что ничего нового искусственный интеллект нам предоставить не может. ИИИ в игре реализован с помощью довольно сложных, но уже всем знакомых поведенческих деревьев, аналог которых рассматривался в этой статье чуть ранее, каждое из которых имело по сотне узлов для любых непредвиденных в игре ситуаций. Однако, что интересно, некоторые узлы деревьев были заблокированы для того, чтобы в процессе прохождения игрок видел все новые и новые особенности и интересные механики, которые ему преподносит игра, на самом деле, просто открывая то или иное заблокированное ранее поддерево способностей. Например то, как ксеноморф учится проверять новые области, где бы вы могли прятаться, или мгновенно замечает вас на большем расстоянии, даже если вы не издавали ни единого звука.

К интересным решениям пришли разработчики из *Monolith Productions*, которые мы увидели в игре *Middle-earth: Shadow of Mordor* (2013). Компьютерная игра, известная своей системой ИИИ под названием «Немезис» (англ. *Nemesis*), удивила всех своим подходом к реализации интеллектуальных систем. В ней несобственные игровые персонажи генерируются и появляются в игровом мире со случайным именем, внешними характеристиками, повадками и другими особенностями, например, такими как слабости. Но что интересного в данной системе, так это то, что вы можете устанавливать некоторые отношения с персонажами, например, если вы уйдете от столкновения и сбежите с поля боя, враги это запомнят и при следующей встрече будут глумиться над вами, а враг, который вас одолел, может подняться по карьерной лестнице и потом припомнить вам это.

Также «Немезис» предоставляет неплохой спектр вариативности действий в зависимости от разных слабостей и других особенностей ваших врагов, ведь если одного оппонента можно поджечь, то другой будет устойчив к таким повреждениям, что заставляет игрока учиться и адаптироваться к игровому миру.

Заключение

Подводя итог всему вышесказанному, хочется отметить, что игровая индустрия не стоит на месте, современные компьютерные игры и компьютерные игры прошлого столетия едва ли можно назвать хоть отдаленно схожими, искусственный интеллект – тем более. Методы, которые внедрялись двадцать лет назад и достойно показывают себя сегодня, уже не будут применяться в будущем. В конце концов, игры имеют неограниченный потенциал и всегда будут удивлять нас своими новыми достижениями.

Список литературы

1. Шампандар А.Д. Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2007. – 768 с.
2. Millington, Ian Artificial intelligence for games / Ian Millington, John Funge. – 2nd ed. – 895 с.
3. Woodcock St. Game AI: The state of the industry [Электронный ресурс]. – 1998. – Режим доступа: http://www.gamasutra.com/view/feature/131705/game_ai_the_state_of_the_industry.php
4. Элементы игрового ИИ на основе табличной логики и Марковских цепей [Электронный ресурс]. – 2012. – Режим доступа: http://www.gamedev.ru/code/articles/Markov_chain_AI
5. Казунов В. S.T.A.L.K.E.R.: Чистое небо – интервью о проблемах выживания искусственного интеллекта в Чернобыльской зоне [Электронный ресурс]. – 2008. – Режим доступа: <http://www.gametech.ru/cgi-bin/show.pl?option=article&id=85>
6. Playstation 4 vs Xbox One: какая консоль продается лучше? [Электронный ресурс]. – 2016. – Режим доступа: <http://ichip.ru/playstation-4-vs-xbox-one-kto-lider-prodazh.html>
7. Грэй С. Свежая статистика о российских геймерах [Электронный ресурс]. – 2013. – Режим доступа: <https://hi-news.ru/research-development/svezhaya-statistika-rossijskix-gejmerax.html>

Bukovshin Vadim Alexandrovich, student

(e-mail: vadimbukovshin199813@gmail.com, Luck1w1nner@outlook.com)

Voskoboinikov Sergey Georgievich, Cand.Hist.Sci., associate professor

Don state technical university, Rostov-on-Don, Russia

(e-mail: voskoboinikov1968c@yandex.ru)

Abstract. This article deals with the technical implementation of artificial intelligence in games, the main methods of creating intelligent systems, suggests ways to improve them, and also describes the prospects for the development of this area of information technology.

Keywords: computer game, game industry, interactive entertainment area, game artificial intelligence, artificial intelligence, intellectual game system, in-game characters, non-player characters (NPC), player characters, game world, scripting, movement control, decision-making management, decision trees, behavioral trees, finite automata, planning systems, Markov chains, neural networks.