

*Шутов Кирилл Игоревич, студент магистратуры, 2 курс,
МИРЭА-Российский технологический университет (РТУ МИРЭА),
Институт информационных технологий, Россия, г. Москва*
*Акатьев Ярослав Алексеевич, ассистент кафедры Практической и
прикладной информатики (ППИ),
МИРЭА-Российский технологический университет (РТУ МИРЭА),
Институт информационных технологий, Россия, г. Москва*
E-mail: bramind002@mail.ru

*Лобанов Александр Анатольевич, доцент кафедры инструментального и
прикладного программного обеспечения (ИиППО), к.т.н., доцент.
МИРЭА-Российский технологический университет (РТУ МИРЭА),
Институт информационных технологий, Россия, г. Москва*

АНАЛИЗ ОСОБЕННОСТЕЙ ПОВЕДЕНИЯ НЕИГРОВЫХ ПЕРСОНАЖЕЙ В ВИРТУАЛЬНЫХ МИРАХ

Аннотация: Целью данной статьи является формирование алгоритма, определяющего подход к созданию неигровых персонажей (NPC) в виртуальной вселенной. Для этого в данной статье будут проанализированы NPC из различных видеоигр: Assassin's Creed: Unity, Игры Bethesda Softworks, The Legend of Zelda Breath of the Wild. Результатом этого анализа будут характеристики игры, которые влияют на подход в разработке алгоритма выбора поведенческой стратегии. Данные характеристики будут использованы для построения сводной таблицы, по которой можно будет проследить ключевые различия между играми. С помощью индукции сводная таблица будет использована для разработки искомого алгоритма.

Ключевые слова: геймдизайн, геймдев, NPC, видеоигры.

Annotation: The purpose of this article is to form an algorithm that determines the approach to creating non-player characters (NPCs) in the virtual universe. To do this, this article will analyze NPCs from various video games: Assassin's Creed: Unity, Bethesda Softworks games, The Legend of Zelda Breath of the Wild. The result of this analysis will be the characteristics of the game that affect the approach to the development of algorithms for choosing a behavioral strategy. These characteristics will be used to build a summary table, according to which it will be possible to trace the key differences between the games. The pivot table will be used to develop the desired algorithm, by induction.

Key words: gamedesign, gamedev, NPC, video games.

Видеоигры — направление в ИТ, в котором создают интерактивный опыт с помощью электронных устройств. Интерес к этой сфере постоянно растёт, что приводит к увеличению прибыли, а это в свою очередь приводит к тому, что бюджеты с каждым разом становятся всё больше.

Существует множество способов вовлекать и удерживать пользователей в играх. Один из них строится на том, чтобы создать правдоподобный виртуальный мир, который будет интересно изучать и в котором будут происходить большое число неожиданных событий. Одним из составляющих виртуальных миров являются NPC (A non-player character — неигровой персонаж). Хорошо проработанное поведение виртуальных существ, может значительно усилить правдоподобность открытого мира и погружение в него игрока. И наоборот слабо проработанные персонажи могут разрушить иллюзию правдоподобности виртуального мира. Поэтому студии тратят много сил, денег и времен на проработку алгоритмов выбора поведенческой стратегии (набор программных методов, которые используют в видеоиграх для создания иллюзии разума у NPC через поведение персонажей). Однако в их задачу не входит создание сложного поведения, они стремятся создать иллюзию реалистичного взаимодействия персонажей с виртуальным миром.

Игры ещё и искусство, и, как и любому другому искусству, им в первую

очередь важно создать опыт у зрителя. А для этого допускается использование игровых условностей, специальное ослабление противников и любое другое ограничение алгоритма выбора поведенческой стратегии. Ведь задача NPC не только поддержание правдоподобности мира, но они ещё и несут для игрока определённую функциональность.

Другой причиной, по которой разработчики не делают сложный алгоритм выбора поведенческой стратегии – ограничения производительности у разных устройств. Ведь важно чтобы ознакомиться с игрой могло максимальное количество пользователей. А для этого системные требования игры должны быть как можно ниже.

Эти вопросы остро стоят перед разработчиками игр в целом и NPC в частном. Целью данной статьи является анализ алгоритмов поведения неигровых персонажей (NPC) в разных играх, установление методом индукции взаимосвязи между принципами работы NPC и геймплейными составляющими игры (жанр, устройство мира, идея игры и т.д.) и формирование собственного алгоритма, определяющего подход к созданию NPC в виртуальной вселенной.

Далее будут рассмотрены неигровые персонажи из нескольких видеоигр. Это позволит проанализировать, как работает алгоритм выбора поведенческой стратегии в играх, на что делают упор разработчики, какие разные подходы они используют и в чём преимущества и недостатки каждого. В данной статье будут выделены следующие подходы:

1. Модели поведения и случайная генерация NPC.
2. Процедурная генерация скриптов и ручное создание NPC.
3. NPC как функции.

Модели поведения и случайная генерация NPC.

Начнём с компании Ubisoft, которая в Assassin's Creed: Unity с помощью NPC хотела погрузить игрока в нужную атмосферу. Разработчикам нужно было вызвать у игрока ощущение, будто Париж во время французской революции живет сам по себе. Также требовалось разработать такой алгоритм выбора поведенческой стратегии, чтобы действия толпы были похожи на реальные;

следовало генерировать случайные ситуации, в которых мог бы участвовать игрок — как в миссиях, так и в свободной игре (См. рисунок 1).



Рисунок 1 — Скриншот из Assassin's Creed: Unity

Париж должен был вести самостоятельную жизнь, в которую игрок мог бы включаться (но не был бы обязан это делать). нужно было наделить NPC моделями поведения и социальными взаимодействиями [2].

Для реализации этого требования в данной игре, разработчики разбили NPC на 4 фракции, каждая со своими функциями:

- «экстремисты»;
- «защитники»;
- «стражники»;
- «горожане».

Каждый из персонажей, независимо от фракции, получает информацию об окружении, об окружающих их персонажах, считывая при этом их фракцию и состояние. Исходя из этого алгоритм выбора поведенческой стратегии

определяет те или иные действия. Например: если экстремистов больше, они начинают драку, если же их численность станет меньше, то они убегут. Некоторые же фракции могут активироваться только во время конфликта, другие выступать против зачинщиков.

Фракция «горожане» имеют классификацию поведения, основанные на характере:

- храбрецы;
- трусы;
- обычные.

Помимо этого, у них есть другие подтипы, основанные на функциях.

Также в игре есть, так называемые «Станции толпы», которые используются на улице и внутри зданий для создания бытовых ситуаций, при этом уникальных вариантов развития событий около 500. Станции располагались вручную дизайнерами уровней.

В дополнении к ним создатели игры добавили спонтанные взаимодействия между жителями. Например, двое знакомых могут столкнуться и завести разговор или начать спорить. Влюблённые встречаются, берутся за руки и уходят [2].

В результате даже без комплексной системы внутри алгоритма, без проработки каждого конкретно взятого персонажа, и с использованием ограниченного числа реакций (900 – 1000), получилось создать «живых» NPC, так как несмотря на малое число реакции отдельно взятого персонажа, они зависят от окружающих условий. В итоге при отличающихся начальных условиях, события начинают развиваться по-разному и чем дольше длится событие, тем сильнее будут различия, что в итоге будет достигать уникального события. Это похоже на игру «Жизнь», клеточного автомата, где простой алгоритм приводит к сложному и непредсказуемому поведению колонии.

Чтобы игра была более разнообразной, в ней предусмотрен непосредственно генератор случайных событий. Основная цель была — сделать так, чтобы для событий требовалось как можно меньше вмешательства

дизайнеров уровней. Поэтому каждое событие «собирается» из логических блоков. К примеру, погоня в событиях «поймать вора» и «убить преступника» — это один и тот же элемент, а вот завершение у них разное. Потенциально можно создавать и более сложные события из большего количества блоков. Эффективность такого подхода подтвердилась в конце разработки, когда создатели игры решили добавить в неё дополнительные события. Благодаря модульной системе, им удалось сделать это за считанные минуты [2].

События генерирует контроллер событий. Он анализирует пространство, ища подходящее место для нового события недалеко от игрока, затем он активирует ближайших подходящего для этого события персонажей (см. Рисунок 2). Либо создаёт необходимых пряча этот процесс за толпой.

В итоге система удобная для разработки, может создавать уникальные события. Благодаря системности и параллельной работе контроллера событий и поведения самих NPC создаётся «реалистичность» происходящего. Однако из минусов можно отметить однообразные типы событий, которые хоть и уникальные, но в основном генерируют боевые ситуации. С другой стороны, учитывая, что игра – экшн ориентирована, этот недостаток не сразу бросается в глаза, так как цель алгоритма выбора поведенческой стратегии генерировать игроку сражения.

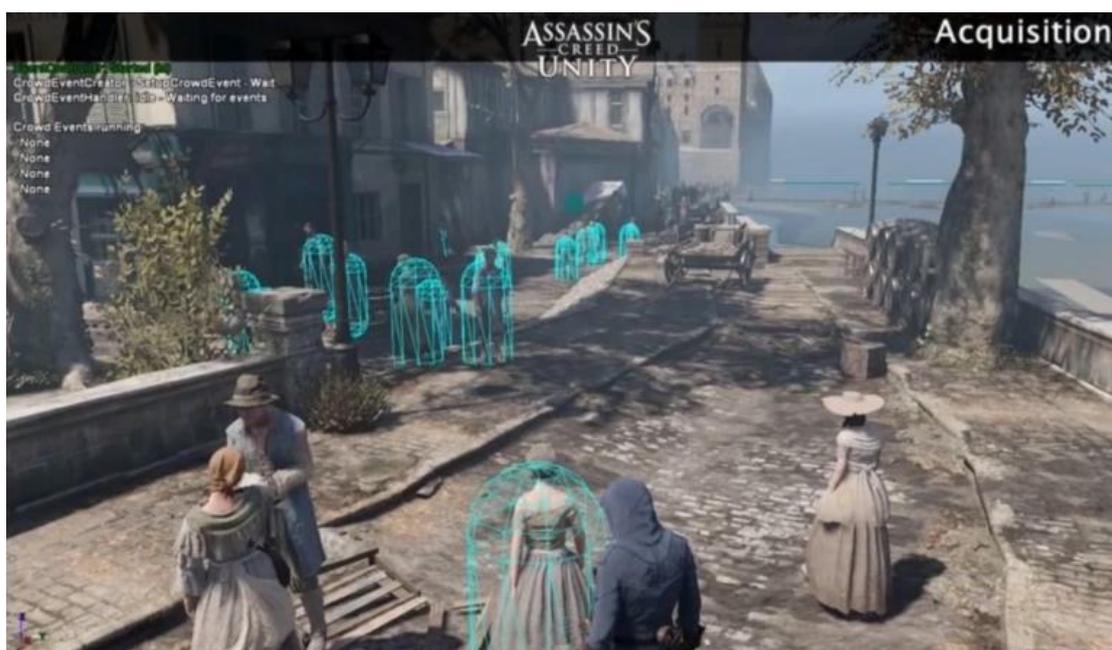


Рисунок 2 — Контроллер событий ищет потенциальных персонажей.

Процедурная генерация скриптов и ручное создание NPC.

Далее рассмотрим технологию, которая специально создана для усложненного поведения NPC. Radiant AI – запатентованная система, разработанная Bethesda Softworks и используется в играх этой компании: The Elder Scrolls Oblivion, Elder Scrolls Skyrim, Fallout 3, Fallout 4. Radiant AI: NPC даются общие цели, и они думают, как их достичь. Отсутствие индивидуального сценариев для каждого персонажа позволяет построить мир гораздо больших масштабов, чем в других играх. Для этого система для каждого NPC процедурно генерирует скрипты.

По сути, это означает, что неигровые персонажи в игре настроены так, чтобы следовать своего рода плану. Например, в Oblivion большинство неигровых персонажей запрограммированы, чтобы есть, спать и работать в определённое время дня, чтобы сделать мир более реальным и динамичным.

Отличительной особенностью является то, что NPC могут свободно интерпретировать свои цели и принимать решения исходя из своих характеристик. Так, например, если персонажу нужно поесть, то он может сделать это разными способами:

- если у него есть еда он просто её съест;
- если у него есть деньги, то он пойдёт к торговцу, чтобы купить еду;
- если у него низкая ответственность, то он попытается еду или деньги украсть;
- если у него достаточно силы, то он пойдёт охотиться.



Рисунок 3 — персонажи на рынке.

И такая вариативность доступна для каждой цели. Благодаря этому создаётся целая виртуальная экосистема, в которую верит игрок. Также стоит отметить, что в отличие от *Assassin's Creed*, где персонажи создавались перед игроком из определённого пула и не имели личностей, а их характеристик генерировались более-менее случайно, здесь у персонажей есть имя, история и многих из них создают вручную, что занимает больше времени.

Когда игрок приходит в какое-либо место, то игра создаёт в нём тех персонажей, которые должны в это время и в этом месте там находиться. При этом за городами расположены определённые точки для каких-либо событий. Когда персонаж к ним приближается на этих точках случайным образом появляются либо персонажи из пула кочевников, либо генерируется уникальное событие с помощью дополнительной системой *Radiant Story*. Однако, если игрок загрузит сохранение игры до встречи с NPC, то заметит, что события там генерируются случайно, и этот момент может выбить игрока из погружения.

Существует интересная история о разработке, которую рассказывал Эмиль Паглиаруло геймдизайнер *Oblivion* [7]. Он описал как ввремя плейтестов, выполняя определённый квест, где нужно было встретиться с торговцем, этого продавца иногда находили мёртвым: дело было в том, что NPC, которым надо было добыть товар, не хватало на него денег. Так что в стремлении заполучить

напиток они убивали продавца.

Из-за Radiant AI города в Oblivion страдали от анархии: жители решали, что самый лёгкий способ решить проблемы – это убить первого встречного и забрать его деньги. Часто жертвами становились важные квестовые персонажи, так что систему решили ограничить, заблокировав некоторые опции для NPC [5]. Стоит учитывать, что это всё могло быть просто маркетинговым ходом.

Однако в любом случае это подводит к важной мысли: если создавать алгоритм выбора поведенческой стратегии и ничем его не ограничивать, то невозможно будет заложить в игру и передать игрокам определённый опыт. Поэтому важно в первую очередь думать об игроке и только потом об атмосфере “живого” мира.

В итоге Radiant AI является революционной технологией, которая позволяет наполнять большие открытые миры большим количеством персонажей. При этом студия грамотно поступает, ограничивая алгоритм выбора поведенческой стратегии, и вводя определённые правила для поддержания заданного опыта игрока. Недостатки у этой системы есть, но они не существенны для жанра RPG (role-playing game — ролевая игра), а именно низкая функциональность большинства персонажей и плохая визуальная читабельность этих персонажей.

NPC как функции.

Далее рассмотрим пример, где разработчики в первую очередь делают упор на функциональность, а не на реализм. В “The Legend of Zelda Breath of the Wild” можно заметить, что поведение персонажей сильно упрощено по сравнению с играми Bethesda. У персонажей здесь тоже есть личность, распорядок дня, они могут реагировать на окружающие события и в целом, если не приглядываться, могут создать впечатление живых людей. Однако в отличие от Radiant AI у персонажей нет вариативности в достижении своих целей. Их возможное поведение строго закриптовано (см. Рисунок 4). Это обуславливается жанром, когда Radiant AI используется в ролевых играх, где подразумевается, что сам игрок может отыгрывать разного персонажа, тем

самым совершая разнообразные действия. И соответственно у NPC должна быть возможность вариативно реагировать на эти действия. The Legend of Zelda же является экшн приключением, а главный герой - доблестный рыцарь. Игра запрещает совершать противоправные действия, а персонажи предоставляют определённые функции, к которым у игрока всегда должен быть доступ, чтобы темп геймплея всегда был достаточно высоким.

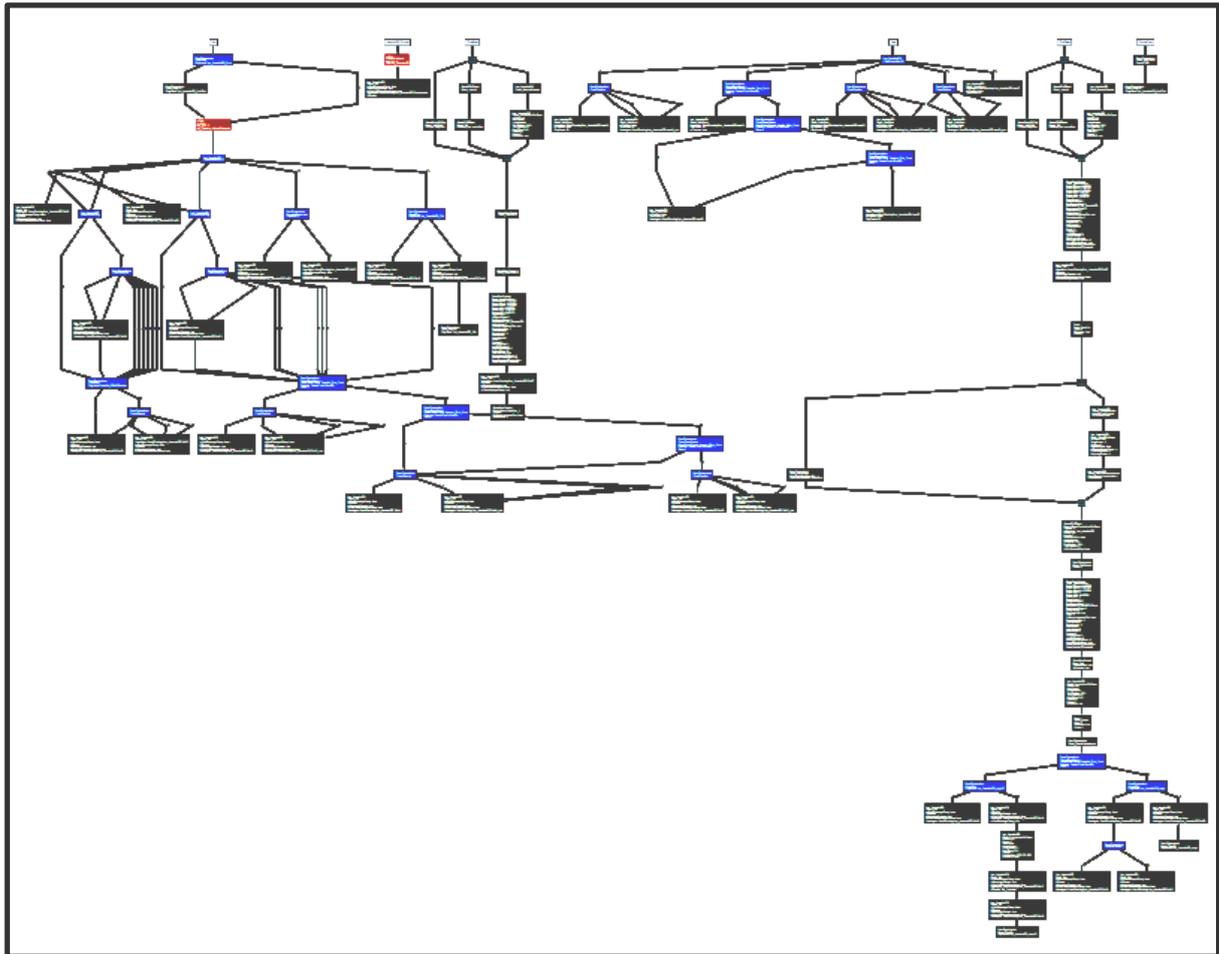


Рисунок 4 — дерево поведения NPC в The Legend of Zelda Breath of the Wild.

В поселениях и между ними игра создаёт персонажей аналогично играм Bethesda, только разнообразных ситуаций меньше, а сюжет в таких событиях не генерируется.

Ориентир на функциональность можно проследить на примере странствующего торговца (См. Рисунок 5). Он всегда появляется возле конюшней. Даже когда игрок мгновенно телепортируется из одного места в

другой. Торговец всё равно там появится. Разработчики не стали избегать этой игровой условности и, например, создавать для каждой конюшни своего торговца. Им было важно, чтобы данный персонаж ассоциировался с торговлей, и функция и внешний вид были едины.



Рисунок 5 — странствующий торговец.

По части NPC Breath of the wild уступает по проработке и атмосфере двум другим рассматриваемым играм. Однако в самой игре упор сделан на дикую природу, где окружению проработано на высоком уровне. Поэтому персонажи не задерживают игрока, а просто выполняют свои цели (выдача заданий, торговля и др.), чтобы игрок быстрее вернулся в основную часть игры.

Исходя из вышеизложенного можно сделать вывод, что подход к разработке неигровых персонажей, их проработка и сложность в первую очередь зависит от жанра игры и от планируемого разработчиками игрового опыта. Помимо этого, можно проследить, что у каждого подхода есть свои преимущества и недостатки. Чем более непредсказуемое поведение персонажей, тем сложнее разработчикам контролировать игровой процесс, и наоборот.

Для лучшего сравнения особенностей рассматриваем игр была сформирована таблица, в которой отражены отличительные характеристики, влияющие на подход к разработке алгоритма выбора поведенческой стратегии (См. Таблицу 1). В конце данной таблицы представлены используемые способы создания персонажей и событий в игре.

Таблица 1 — Сводные характеристики из рассматриваемых игр.

Игра	Assassin's Creed: Unity	Игры Bethesda Softworks	The Legend of Zelda Breath of the Wild
Жанр	Экшн-приключение	RPG	Экшн-приключение
Количество персонажей	10000-11000	~450	~140
Цель NPC	Создание атмосферы	Выдача заданий и создание атмосферы	Выдача заданий
Цель открытого мира	Вовлекать игрока в разнообразные события	Вовлекать игрока в разнообразные события, позволить игроку самому изучать мир	Позволить игроку самому изучать мир
Способ создания персонажей	Случайная генерация персонажей	Ручное создание персонажей	Ручное создание персонажей
Способ создания событий	Использование контроллера событий	Использование контроллера событий	Ручное размещение событий на карте

На основании сводной таблицы можно определить алгоритм для выбора подхода к созданию NPC в игровой вселенной (см. Рисунок 6).

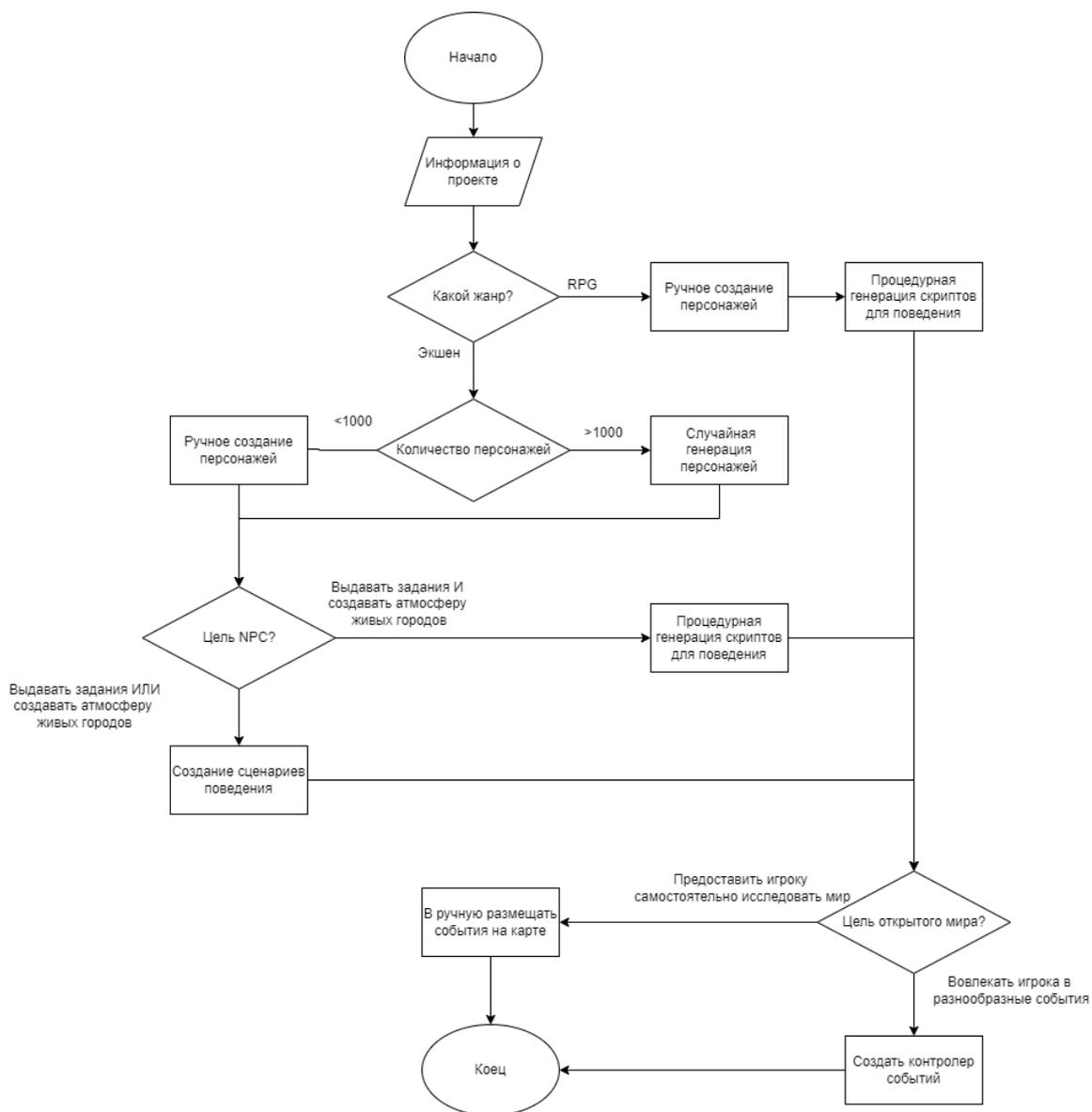


Рисунок 6 — Алгоритм для определения подхода к созданию NPC в игровой вселенной.

В результате были проанализированы следующие подходы к созданию NPC: модели поведения и случайная генерация NPC, процедурная генерация скриптов и ручное создание NPC, NPC как функции. Были выделены ключевые факторы рассматриваемых игр, которые определяют принцип разработки неигровых персонажей. На основе собранной информации был разработан алгоритм для определения подхода к созданию NPC в игровой вселенной. С помощью него разработчики могут подобрать для своей игры наиболее подходящую структуру действий неигровым персонажам для выбора поведенческой стратегии.

Библиографический список:

1. Статья на Skillbox Media “Не совсем человек: искусственный интеллект в играх” <https://skillbox.ru/media/gamedev/iskusstvennyy-intellekt-v-igrakh/> (Дата обращения: 30.05.2022).
2. Статья на DTF о создании поведения NPC в Assassin's Creed Unity [Электронный ресурс] URL: <https://dtf.ru/gamedev/6747-kostry-i-revolucionery-kak-sozdavalos-povedenie-tolpy-dlya-assassin-s-creed-unity> (Дата обращения: 14.04.2022).
3. Видеозапись выступления разработчиков из Ubisoft о создании поведения NPC в Assassin's Creed Unity [Электронный ресурс] URL: https://www.youtube.com/watch?v=Rz2cNWVLncI&ab_channel=GDC (Дата обращения: 14.04.2022).
4. Статья “Radiant AI” [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Radiant_AI (Дата обращения: 19.04.2022).
5. Статья “Игра с искусственным интеллектом, который был «слишком умным»” [Электронный ресурс] URL: http://muz4in.net/news/igra_s_iskusstvennym_intellektom_kotoryj_byl_slishkom_umnym/2018-10-09-46899 (Дата обращения: 19.04.2022).
6. Статья на DTF “Radiant AI от Bethesda как будущее ролевых игр” [Электронный ресурс] URL: <https://dtf.ru/flood/16882-radiant-ai-ot-bethesda-kak-budushchee-rolevyh-igr> (Дата обращения: 19.04.2022).
7. Видеозапись выступления Эмиль Паглиаруло геймдизайнера Oblivion о создании историй в играх Bethesda [Электронный ресурс] URL: https://www.youtube.com/watch?v=Bi51-wjcw8&ab_channel=infiniDK (Дата обращения: 28.04.2022).
8. Статья о параметрах NPC в “The Legend of Zelda Breath of the Wild” [Электронный ресурс] URL: <https://zeldamods.org/wiki/ActorParam> (Дата обращения: 21.04.2022).
9. Видеоролик о разработке “The Legend of Zelda Breath of the Wild” [Электронный ресурс] URL:

https://www.youtube.com/watch?v=4i7y5XnQubk&t=5s&ab_channel=NintendoРоссия (Дата обращения: 21.04.2022).

10. Шелл Джесси Геймдизайн. Как создать игру, в которую будут играть все. Москва: Изд-во Альпина Пабlishер. 2021.