

ОСОБЕННОСТИ АРХИТЕКТУРЫ ВИРТУАЛЬНЫХ АГЕНТОВ, ОСНОВАННЫХ НА ПЛАНИРОВАНИИ В РЕЖИМЕ РЕАЛЬНОГО О ВРЕМЕНИ.

*С.В. Аксёнов, к.т.н. ОИТ ИШИТР,
В.В. Видман, аспирант А7-39.,
В.А. Коровкин, аспирант А7-39,
Томский политехнический университет
E-mail: vvv23@tpu.ru*

Введение

Планирование действий в режиме реального времени является альтернативным методом более распространённых методов моделирования поведения персонажей с помощью сценариев или конечных автоматов. В то время как конечные автоматы осуществляют множество переходов между состояниями для планирования, Real-Time планировщик подбирает последовательность действий виртуального агента под текущую ситуацию.

В работе [1] описаны три основных преимущества планирования в режиме реального времени. Виртуальные агенты, основанные на такой системе, лучше реагируют на случайные ситуации. Цели и действия представляют собой модульные блоки, из которых можно строить поведение, их можно делить и использовать многократно. Архитектура такой системы обеспечивает разделение данных и реализации поведения. Это облегчает разработку и поддержку всей системы, но увеличивает нагрузку на ЦП.

В данной статье мы размышляем над вопросами, которые нужно учитывать при проектировании архитектуры поведения виртуальных агентов, реагирующих на окружение в режиме реального времени.

Архитектура виртуального агента

Агент состоит из нескольких подсистем: рабочая память, датчики, а также различные системы по типу инвентаря, оружия и другие. Датчики обнаруживают изменения в мире и сохраняют собранные данные в динамической рабочей памяти, при этом часть данных перед сохранением оценивается, учитывая настроение агента. Планировщик использует эти представления для принятия решений и, в конечном итоге, передает инструкции подсистемам. Подсистемы включают системы наведения, навигации, анимации и вооружения.

Датчики воспринимают внешние видимые и звуковые раздражители, а также внутренние раздражители, такие как боль и желания. Некоторые датчики управляются событиями, в то время как другие делают опросы нужных определенных предметов. Управляемые событиями датчики полезны для распознавания мгновенных событий, таких как звуки и повреждения. Опрос лучше подходит для датчиков, которым необходимо извлекать информацию из мира. Например, датчик может генерировать список потенциальных тактических позиций. Все знания, генерируемые датчиками, хранятся в памяти. Когда датчики обнаруживают значительные изменения в состоянии мира, агент повторно оценивает актуальность своих целей. Одновременно может быть активна только одна цель. Когда наиболее релевантная цель изменяется, агент использует планировщик для поиска последовательности действий, которые будут соответствовать этой цели. Действие активируется путем установки значений переменных на доске. Подсистемы обновляются с некоторой постоянной скоростью и изменяют свое поведение в соответствии с инструкциями, размещенными на доске. Например, Go_To_Target действие устанавливает новое место назначения. В следующем обновлении навигационная система отреагирует, найдя путь к новому пункту назначения.

Кэширование данных

Кэширование всех знаний в памяти согласованном формате напрямую не повышает эффективность планировщика, а, скорее, предоставляет средства глобальной оптимизации. Факты могут быть помещены в бункеры в зависимости от типа знаний или отсортированы каким-либо образом. Как минимум, самый последний или наиболее часто используемый факт можно кэшировать для немедленного извлечения. Поскольку количество фактов увеличивается, возможно, стоит продолжить оптимизацию запросов. Централизация знаний в рабочей памяти обеспечивает агента постоянным контекстом, учитывая его настроение [2].

Главный сценарный планировщик

Каждый агент не занимается самостоятельно подбором возможных мест для укрытия. Этим занимается Director, который во время командного боя подбирает укрытия таким образом, чтобы действия агентов казались командными, т.е. он берет на себя роль командира. Он передает агентам список укрытий, между которыми они сами решают, как перемещаться. Если бы каждый агент на сцене самостоятельно проводил просчет позиций членов своей команды – это бы значительно увеличило количество и сложность вычислений.

Так же Director раздает роли каждому члену команды, т.е. фактически задает тактику ведения боя, основываясь на особенностях каждого отдельно агента. Персонажу с дальнобойным оружием он предложит для перемещения самые высокие позиции, а бойцу ближнего боя укажет что нужно нападать с фланга. При раздаче этих команд, один из членов отряда, который выбран как командир отыграет соответствующие анимации, создавая видимость своей командной деятельности.

Планирование и динамическое поведение

Сложность, добавляемая распределенной обработкой, кэшированием и планированием, имеет смысл только в том случае, если она приводит к заметно более динамичному поведению. NPC, которые планируют в режиме реального времени, могут справиться с тонкостями и зависимостями, и самая большая выгода заключается в возможности перепланировать. Пример описанной архитектуры представлен на рисунке 1.

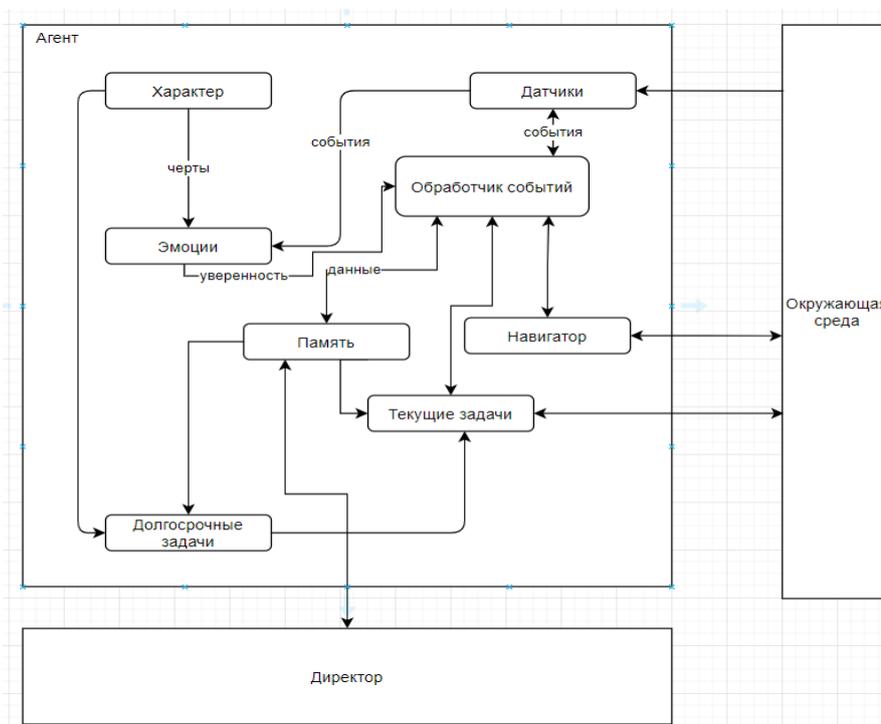


Рис. 1. Структура агента.

Заключение

Данная работа частично позволяет понять всё сложность архитектуры виртуальных агентов, поведение которых должно основываться через три уровня взаимодействия с окружающей их средой. Первый уровень – он должен следовать шаблонам поведения, заданными ему Директором. Второй – он должен корректно реагировать на внешние стимулы. Третий – реагируя на внешние стимулы он должен опираться на внутренние ощущения от происходящего. В дальнейшем планируется рассмотреть такие вопросы поведения агентов как разделение памяти на внутреннюю и общеизвестную, проработка задач и функций директора, а также полное и детальное представление архитектуры.

Список использованных источников

1. Jeff Orkin. Applying Goal-Oriented Action Planning to Games. AI Game Programming Wisdom 2, Hingham, Mass.: Charles River Media, 2003. 217-228.
2. Невмержицкий Антон Александрович. "Использование локального кэширования для увеличения производительности распределенной системы" Вестник науки и образования, no. 11-3 (89), 2020, pp. 14-17.