

Рисунок 1 – Принцип работы песочницы на примере ПК «SelfPortal»

Это деление удобно в приложении к методам, которые позволяют получить доступ к системе. Например, основным способом доступа к виртуальным машинам категории 1 является Remote Desktop Protocol, тогда как для машин категории 2 — Secure Shell или Virtual Network Computing.

NoVNC Viewer является частью программного пакета, предоставляемого непосредственно системой виртуализации как инструмент контроля над виртуальной машиной. Его невозможно централизовать, так что общее число «входных точек» в песочницу будет равно числу используемых провайдеров виртуализации плюс два.

Поэтому необходимо строго настроить брандмауэр, чтобы запретить любой трафик, не прошедший через заданные системы. Кроме того, следует блокировать любой трафик, созданный данными системами, чтобы предотвратить доступ к основной сети предприятия злоумышленником, который получил возможность пользоваться ими. Такие меры обеспечивают сотрудникам предприятия качественно новый уровень сервиса.

Отметим в заключение, что описанная методика не принесет экономической выгоды предприятиям, которые не предоставляют сотрудникам непосредственный доступ к инфраструктуре, поскольку тогда любой виртуальный сервер будет проверяться системным администратором, так что сложные защитные меры по организации песочницы становятся лишними. Вместе с тем для предприятий, которые непосредственно выделяют сотрудникам вычислительные ресурсы, данная методика может стать инновационным решением.

МОДЕЛИ И АЛГОРИТМЫ ОБЕСПЕЧЕНИЯ НЕОБХОДИМЫМИ РЕСУРСАМИ ДЛЯ ЖИЗНЕДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА НА ОСНОВЕ ТЕОРИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Поплёвка В.И.

Скудняков Ю.А. - доцент каф. ПЭ, к.т.н., доцент

В работе рассматриваются возможности использования некоторых подходов и предложенных в ней моделей и алгоритмов обеспечения необходимыми ресурсами для жизнедеятельности человека на основе теории искусственного интеллекта [1,2].

Для формирования поведения человека наиболее подходящим является игровой искусственный интеллект, так как он позволяет задавать алгоритмы объекта в зависимости от текущего состояния системы.

При формировании игрового искусственного интеллекта используют следующие подходы [3]:

1. Машины конечных состояний

Машины конечных состояний являются наиболее распространенным алгоритмом поведенческого моделирования. Они концептуально просты и быстро кодируются, что приводит к созданию мощной и гибкой структуры искусственного интеллекта с небольшими расходами. Машины разбивают искусственный интеллект на

дискретные части, именуемые состояниями. Состояния связаны между собой переходами, ответственными за переход в новое активное состояние всякий раз, когда выполняются определенные условия.

2. Иерархические машины конечных состояний

Машины конечных состояний – полезный инструмент, но у него есть недостатки. Работа с парой состояний не представляет никаких проблем. Однако, когда происходит работа с объемным искусственным интеллектом, содержащем в себе десятки состояний, то добавление новых состояний может быть чрезвычайно сложной. Иерархические машины конечных состояний решают эту проблему, позволяя объединить наборы состояний в отдельные машины и осуществлять переходы между ними, а не состояниями.

3. Дерево поведения

Дерево поведения описывает структуру данных, начинающуюся с некоторого корневого узла и состоящую из поведений, являющимися отдельными действиями, которые объект может выполнять. Каждое поведение определяется предварительным условием, определяющим возможность совершения данного действия. Также поведения, в свою очередь, могут иметь поведения-наследники, что и дает алгоритму свои древовидные качества.

4. Целеориентированный планировщик действий

При формировании искусственного интеллекта, для которого приоритетом является достижение конкретной цели (например, предотвращение голода или жажды), используют целеориентированный планировщик действий. Данный алгоритм позволяет системе искусственного интеллекта создавать свои собственные подходы к решению проблем, предоставляя описание того, как работает мир, то есть формировать список возможных действий объекта, требования перед каждым возможным действием и последствия совершения конкретного действия. Из возможных действий алгоритм находит то, которое приведет к поставленной цели. Далее, анализируя текущее состояние системы и условия, необходимые для выполнения конкретного действия, алгоритм находит действия, необходимые для достижения условий выполнения следующего действия. И так до тех пор, пока алгоритм не достигнет действия, выполнимого с начального состояния системы. Данный алгоритм можно доработать, заменив действия на состояния, что позволит расширить логику поведения, заменив метод выполнения действия методами входа, выхода и обновления состояния.

Разработанная система использует игровой движок Unity в качестве среды демонстрации результатов своего функционирования и целеориентированный планировщик действий в качестве основной части искусственного интеллекта объекта, симулирующего поведение человека или бота. В системе используется компонентно-ориентированный подход, благодаря чему система легко встраивается в приложения, разработанные с помощью игрового движка Unity.

Моделирование искусственного интеллекта выглядит следующим образом: объект инициализирует компонент AIController значениями из компонента AISetup. AIController, передает значения условий из класса AISense и цель объекта в класс AIAgent. AIAgent передает полученную информацию в класс AIPlanner, который формирует план AIPlan и выбирает следующее действие для объекта. Далее AIAgent устанавливает объекту новое действие. Это вызывает смену состояния объекта AIState. Для хранения значений условий используется класс AICondition. Для хранения значений графа поведения объекта используется класс AINode. Для разработки действий, возможных для объекта, используется класс AIScenario. Графическое представление структуры компонентов системы искусственного интеллекта показано на рисунке 1.

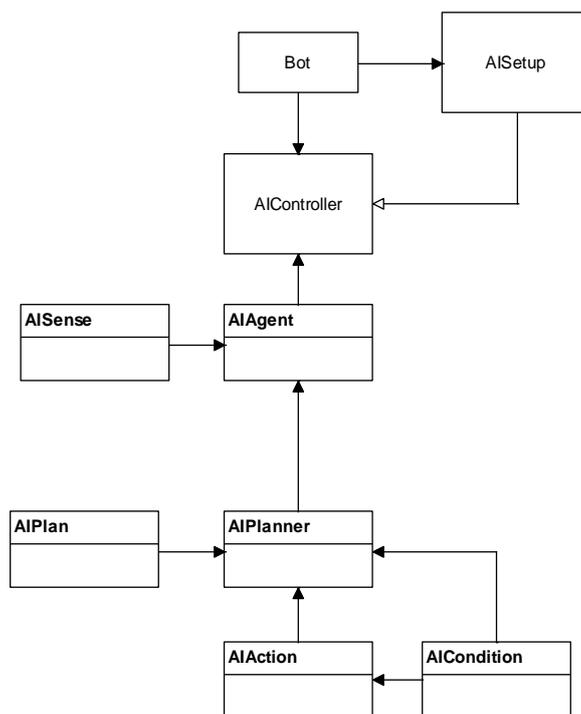


Рисунок 1 – Графическое представление структуры системы искусственного интеллекта

Для работы системы необходимо добавить к объекту компоненты AISetup и AIController. С целью формирования состояний объекта реализуются классы-наследники от класса AIState, а в классе AISense осуществляется реализация логики определения состояний системы. Описание сценария поведения объекта с помощью AIScenario приведено на рисунке 2, которое указывается в компоненте AISetup.

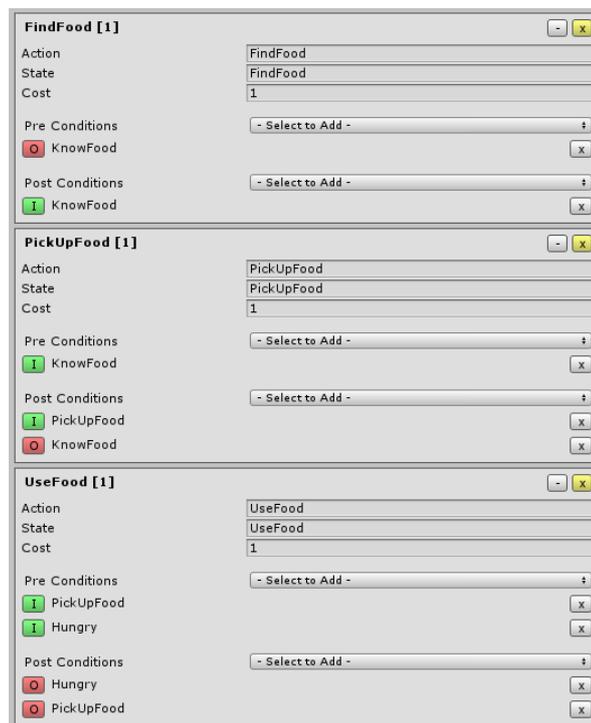


Рисунок 2 – Пример сценария поведения объекта

Список использованных источников:

1. Wikipedia [Электронный ресурс]. – Режим доступа: <http://wikipedia.org>. – Дата доступа: 15.03.2018.
2. Остроух, А.В. Интеллектуальные системы / А.В. Остроух. – Красноярск: Научно инновационный центр, 2015. – 110 с.
3. Rabin, S. Game AI Pro: Collected Wisdom of Game AI Professionals / S. Rabin - A K Peters/CRC Press, 2013. - 626 с.

МИКРОПРОЦЕССОРНОЕ УСТРОЙСТВО МОНИТОРИНГА ПЕРЕМЕЩЕНИЯ АВТОМОБИЛЯ

*Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь*

Прибыткин М.А.

Журавлёв В.И. – доцент каф. ПЭ, к.т.н., доцент

В работе рассмотрены особенности проектирования устройств для мониторинга движения автотранспорта.

На сегодняшний день одной из главных областей применения GPS-технологий является мониторинг автотранспортных средств. Современные спутниковые системы контроля позволяют получать оперативную информацию о точном местоположении объекта, его остановках, скорости, с которой он передвигается, уровне топлива в баке и т.д. [1,2].

Разрабатываемое устройство должно осуществлять следующие функции:

- слежение за транспортным средством;
- индикация местоположения;
- приём передача и данных;
- возможность аварийного вызова вручную;
- резервное питание в случае аварийной ситуации.

Для этого в структуре устройства должны быть предусмотрены: модуль GPS/ГЛОНАСС, модуль GSM, устройство громкой связи, устройство резервного питания, устройство вызова вручную (рисунок 1).