

3. UML-диаграммы классов. [Электронный ресурс]. UML-моделирование. URL: <https://prog-cpp.ru/uml-classes/> (дата обращения: 26.05.2018).

РАЗРАБОТКА ВИРТУАЛЬНОГО ПОЛИГОНА В UNITY3D

Н.И. Журбич, П.А. Зяблицев
(г. Томск, Томский политехнический университет)
e-mail: niz1@tpu.ru, paz4@tpu.ru

DEVELOPMENT OF VIRTUAL POLYGON IN UNITY3D

N.I. Zhurbich, P.A. Zyabletsev
(Tomsk, Tomsk Polytechnic University)

Abstract: The paper is devoted to research and designing of autonomous car – a self-governing car in which artificial intelligence will take the role of driver. In today's world, the present invention is necessary because today humanity cannot imagine being without any types of transport.

Keywords: self-driving car, UML, Unity3D, motion algorithms of a self-driving car, virtual polygon, software package.

Введение. По прогнозам аналитиков *Berg Insight* первые самоуправляемые машины на дорогах общего пользования появятся в 2020 году [1]. Их количество в это время составит порядка 200 тыс. единиц, а через 20 лет объем рынка технологий автономного вождения составит \$ 560 млрд [2]. С 2015 года технологии, необходимые для создания беспилотных автомобилей, активно развиваются и в России.

Беспилотный автомобиль – транспортное средство, оборудованное системой автоматического управления, которое может передвигаться без участия человека [3].

Целью проекта является разработка трехмерной реалистичной имитации процесса управления движением автомобиля в городской среде. Создаваемый программный комплекс позволит разрабатывать аппаратные и программные системы беспилотного автомобиля и проводить виртуальное испытание разработанных систем, в том числе, алгоритмов автоматического управления транспортным средством.

Данный программный комплекс разрабатывался на межплатформенной среде разработки компьютерных игр *Unity 3D* с помощью инструментов *Microsoft Visual Studio 2017* и *Mono Developer*.

Отличительной особенностью данного проекта является реализация движения беспилотного автомобиля в виртуальной среде, что является новым подходом при создании беспилотного автомобиля. Большинство аналогов на рынке создают свои собственные полигоны для тестирования беспилотного автомобиля, что требует большого количества времени и денежных средств.

Создание виртуального полигона. Проект в *Unity* делится на сцены (уровни) – отдельные файлы, содержащие игровые миры с определенным набором объектов, сценариев и настроек. Сцены могут содержать в себе объекты-модели и пустые игровые объекты, которые не имеют модели [4].

В данной работе используются сцены день/ночь с включением погодных условий (дождь/снег). Также присутствуют навигационные сцены «Меню», «Настройки» (установка требуемых параметров графики).

В сценах используются объекты для построения требуемых моделей. Объекты содержат наборы компонентов, с которыми и взаимодействуют скрипты. У объектов есть название, может быть тег (метка) и слой, на котором он должен отображаться. Например, у любого объекта на сцене обязательно присутствует компонент *Transform*, который хранит в себе координаты местоположения, поворота и размеров объекта по всем трём осям.

У объектов с видимой геометрией по умолчанию присутствует компонент *MeshRenderer*, делающий модель объекта видимой.

Для создания модели и анимации пешехода использовался плагин *Mecanim*. *Mecanim* – это система анимации *Unity*, которая позволяет создавать «состояния» объекта. Данные «состояния» воспроизводят анимацию и определяют логику переходов между разными состояниями объекта.

Основной задачей являлось не только верное конструирование модели пешехода, но и его анимации. Необходимое условие заключалось в том, чтобы создать скелет пешехода для использования анимации. Реализация данного этапа работы приведена на рисунке 1.

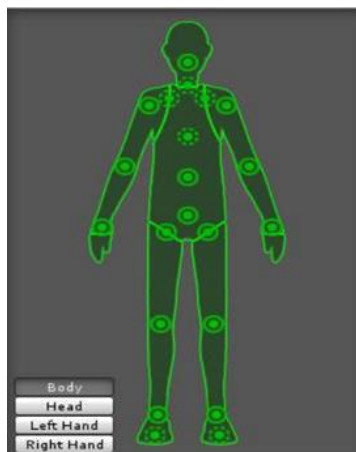


Рис. 1. Составление скелета для объекта

Следующим этапом была разработка анимации к скелету. С этой целью использовался контроллер аниматора (*AnimatorController*). Контроллер аниматора создается *Unity* и позволяет руководить набором анимаций для персонажа и переключаться между ними, когда выполняется некоторое условие. Контроллер управляет переходами между анимациями, используя, так называемую, машину состояний (*StateMachine*) – это некая программа, созданная языком визуального программирования в *Unity*.

Описание алгоритма. Для дальнейшей разработки полигона потребовалось решить следующие задачи:

- внедрение пешехода в систему полигона;
- движение пешехода по определенному маршруту;
- движение беспилотного автомобиля по определенному маршруту.

Для создания маршрута пешехода было предложено использовать набор пустых точек (*Node*). Точки объединяются родительским объектом – полоса движения (*Point*). Точки были выбраны, потому что они самые легковесные и практически не несут нагрузку на вычислительные ресурсы.

На точки определяется скрипт, который содержит информацию о следующем: обязательна ли точка для возврата, является ли она точкой стыка, метка точки (*start, end, centerofrotation, none, constriction*), ID, позиция, все возможные пути из этой точки.

На пешехода определяется скрипт, также содержащий информацию о полосе. Основная функция скрипта – нарисовать в редакторе видимые линии для соединения точек необходимого маршрута.

В результате решения данной задачи пользователю достаточно определить маршрут, по которому следует двигаться пешеходу, и любой объект сможет двигаться по заданному пути.

Блок-схема алгоритма движения пешехода представлена на рисунке 2.



Рис. 2. Блок-схема алгоритма движения пешехода

Заключение. В данной работе был создан программный комплекс, позволяющий производить тестирование алгоритмов движения беспилотного автомобиля в виртуальном полигоне.

В процессе разработки были решены следующие задачи:

- создание виртуального полигона;
- разработка анимации и алгоритма движения пешехода;
- разработка алгоритма управления системой светофоров на полигоне.

Дальнейшие планы разработки. Следующим этапом разработки программного комплекса является отладка алгоритмов взаимодействия между автомобилем и пешеходом. Для решения проблемы взаимодействия пешехода и беспилотного автомобиля понадобится построить карту путей игрового уровня.

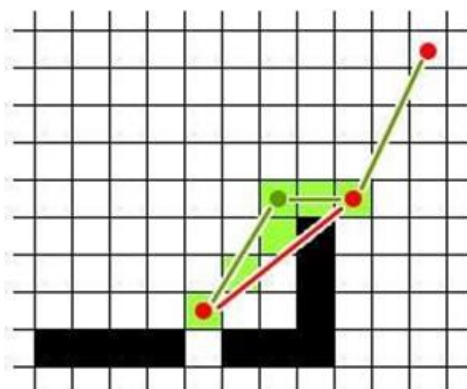


Рис. 3. Построение графа путей

Построение карты (графа) путей происходит единожды при запуске уровня. Для этого проводится поиск всех отмеченных определенным тегом объектов и привязывается к заданной пользователем координатной сетке. Для каждого объекта должна существовать заранее заданная карта его проходимости, размеченная при моделировании объекта, таким образом,

получается битовая карта проходимости. На рисунке 3 продемонстрирован процесс построения графа путей для автомобиля.

ЛИТЕРАТУРА

1. Беспилотные автомобили, мировой рынок [Электронный ресурс]. Деловой портал TAdviser. URL: <http://www.tadviser.ru/index.php/> (дата обращения: 16.03.2018).
2. Япония потратит более \$ 25 млрд. на беспилотные автомобили [Электронный ресурс]. Федеральное агентство новостей. URL: <https://riafan.ru/789196-yaponiya-potratit-bolee> (дата обращения: 15.03.2018).
3. Беспилотные автомобили интересуют больше половины россиян [Электронный ресурс]. 3D news. URL: <https://3dnews.ru/947327> (дата обращения: 28.03.2018).
4. Руководство к cartutorial (Unity 3D) [Электронный ресурс]. Habr. URL: <https://habr.com/post/247667> (дата обращения: 28.04.2018)

АНАЛИЗ ДАННЫХ С ПОМОЩЬЮ ТЕХНОЛОГИЙ BIG DATA

Н.И. Журбич, П.А. Зяблицев
(г. Томск, Томский политехнический университет)
e-mail: niz1@tpu.ru, paz4@tpu.ru

DATA ANALYSIS USING BIG DATA TECHNOLOGIES

N.I. Zhurbich, P.A. Zyablitsev
(Tomsk, Tomsk Polytechnic University)

Abstract: This article focuses on the use of social data analysis in various areas of activity and ways of analyzing and processing this data.

Keywords: Big Data technologies, data mining, methods of data analysis, personal data, areas of use.

Введение. Социальные данные – самое ценное сырье XXI века, новая нефть, которая неиссякаема, считает исследователь, автор новой книги «*Big Data*. Вся технология в одной книге» Андреас Вайгенд [1]. Однако, как и нефть социальные данные требуют обработки до того, чтобы извлечь из них полезную информацию. Данный вопрос как никогда актуален в эпоху информационных технологий, так как объемы этих данных растут в геометрической прогрессии. Данная статья посвящена вопросу применения анализа социальных данных в разных сферах деятельности и способы анализа и обработки этих данных.

Области использования социальных данных (персональных данных). Персональные данные (ПД) или личностные данные — любые сведения, относящиеся к прямо или косвенно определенному, или определяемому физическому лицу (субъекту персональных данных), которые предоставляются другому физическому или юридическому лицу либо лицам [2].

Под социальными данными в свою очередь понимается совокупность персональных данных большого количества людей. Специалистам в области *Data Science* важны не личные данные конкретного человека, а большие объемы этих данных для анализа и построения общей модели поведения группы людей.

Источниками социальных данных являются:

- Данные из социальных сетей.
- Данные интернет провайдеров (посещаемые сайты, покупки в интернете, скачиваемый контент и т.д.).
- Данные о физической активности пользователя (сбор информации с носимых устройств и устройств контроля жизнедеятельности).
- Данные из государственных систем.