

А. А. Полховский, Е. В. Комракова
(ГГТУ им. П. О. Сухого, Гомель)

РАЗГОН ФИЗИЧЕСКОГО ДВИЖКА UNITY ПРИ СОЗДАНИИ ИГРЫ TOWER DEFENSE

Физические движки действуют в предположении, что повторение происходит в фиксированные моменты времени, и оба физических движка *Unity* действуют именно так. Технически в *Unity* присутствуют два физических движка: *NVIDIA PhysX* для поддержки 3D-физики и открытый движок *Box2D* для поддержки 2D-физики. Выполняя вычисления, они основываются на конкретных значениях времени. Подход называется обновлением с фиксированным интервалом времени. По умолчанию интервал обладает длительностью 20 миллисекунд, либо 50 обновлений в секунду.

Когда между отображением последовательных кадров возникает чрезмерно огромный промежуток (низкая частота кадров), система обновляется много раз перед началом нового отображения.

Метод *FixedUpdate()* вызывается в момент, когда физическая система начинает ближайший шаг моделирования. Это один из самых необходимых методов обратного вызова в *Unity*, какой можно определить в сценарии *MonoBehaviour* и использовать для выполнения вычислений, не зависящих от частоты замены кадров.

Если прошло много времени с момента последнего фиксированного обновления, физическое и фиксированное обновления будут рассчитываться до момента, пока не достигнет текущего времени. Например, если отображение предыдущего кадра заняло 60 мс, физической системе потребуется реализовать три цикла обновления состояния.

Поэтому, во время трудных вычислений физический движок возможно не успеет завершить обновление за 20 мс, что прибавит снова на 20 мс работы, и так до бесконечности, из-за чего не станет нельзя вывести следующий кадр. Чтобы предотвратить блокировку игры физическим движком, определяется наибольшее время, какое ему разрешено делать обрабатывание между отображениями. Данный предел называют максимальной допустимой длительностью.

Была разработана и оптимизирована игра в жанре «*Tower Defense*», а особенно расчет столкновений и перемещения, масштаби-

Первые шаги в IT-сфере

рование объектов было совершено исключительно внутри методов *FixedUpdate()* и прочих методов обработчиков. Таких как *OnTriggerEnter2D*, *OnCollisionEnter*, *OnTriggerEnter*, *OnMouse* и других.

Д. А. Порохов
(ГГУ им. Ф. Скорины, Гомель)

РАЗРАБОТКА ETL-ПРОЦЕССОВ, ОСНОВАННЫХ НА БИРЖЕВЫХ ДАННЫХ

С развитием бизнеса, бирж и в целом интернета, данных становится всё больше. Следовательно, тот, кто владеет большим знанием, тот владеет миром. Поэтому возникла необходимость анализа данных, разработки различных видов отчётов с целью анализа деятельности компаний, бизнесов, сборов статистик и так далее. Эффективно анализировать данные, не отвлекаясь на посредственные задачи, стало возможно в так называемых ETL-процессах. Процессы ETL (Extract, Transform, Load) могут реализовываться с помощью огромного количества инструментов.

В данной работе используются процедуры.

1. Экстракция с помощью API «investpy», которое в свою очередь является библиотекой языка программирования «Python».
2. Трансформация – использование библиотек «Python» для приведения данных к конечному, нужному для пользователя виду.
3. Загрузка биржевых данных в базу данных.
4. Загрузка биржевых данных в «S3».
5. Создание базы данных с помощью СУБД «PostgreSQL».

В работе был реализован инструмент, который может быть использован, при соответствующей настройке, как в качестве обучающего инструмента для студентов, так и для трейдеров, которые нацелены на анализ биржевых данных. Уникальность этого инструмента в том, что для заинтересованных в данных по акциям лиц, которые не знают где взять данные, как использовать такие данные в Python, или как получить такие данные в MS Excel файлы или базы данных, этот инструмент станет неотъемлемым проводником.