

В проекте была использована физика, чтобы придать динамику и коллизию игровым объектам. Галочка *Use Gravity* поставлена с целью имитации силы притяжения. *Constraints* никакие не ставятся, так как все действия игры происходят в 3D пространстве. Поле массы выставлены для объекта *Human 75*, а для катящихся шаров – 500.

Компонент *Rigidbody* очень помогает в проекте, потому что он содержит большую часть логики физических вычислений и много полезных методов, которые помогают рассчитывать коллизии, силу трения с поверхностью, задавать тип силы (импульсная, игнорирующая массу, мгновенная, постоянная). В проекте чаще всего использовался метод *AddForce()* для изменения действующей силы на игровые объекты, при этом тип силы, который используется для изменения силы, есть импульсная. Это задаёт некоторый необычный поведение, с рывками, что делает игру довольно весёлой и не такой простой.

**Р. В. Топольцев**

(ГГУ им. Ф. Скорины, Гомель)

## ИГРОВОЕ ПРИЛОЖЕНИЕ «ТРИ В РЯД» В СРЕДЕ UNITY 3D

Игры в жанре «Три в ряд» не являются чем-то новым и уже давно, достаточно прочно, закрепились в игровой индустрии. Они характеризуются тем, что их игровое поле состоит из сетки элементов, а задачей игрока является манипулирование элементами таким образом, чтобы создавать заданные игрой шаблонные комбинации, и тем самым заставить собранные элементы исчезнуть.

Большой особенностью игр «Три в ряд», которая сильно влияет на их успех – такие игры абсолютно не требовательны ко времени. В них можно поиграть 5 минут, например, во время перерыва, или во время поездки в транспорте, а можно и вовсе увлечься, забыв про счёт времени.

Для разработки игры был выбран язык программирования C#, а игровым движком стала межплатформенная среда разработки компьютерных игр Unity, которая имеет чрезвычайно производительный визуальный рабочий процесс и мощную межплатформенную поддержку. Благодаря этому разработанное приложение можно без проблем адаптировать как для мобильных устройств, так и для персональных компьютеров.

Разработанная игра представляет собой бесконечный уровень со счётом, на котором пользователь может неограниченное количество времени практиковаться в уничтожении рядов игровых объектов путём их манипулирования и составления комбинаций для уничтожения.

При запуске игры автоматически создается игровое поле размером 576 x 896. Заполнение происходит случайным образом, поэтому шанс того, что пользователь будет начинать игру на одном и том же игровом поле, крайне мал. Помимо этого, исключаются ситуации, при которых на самом старте уже будут сформированы какие-либо комбинации, готовые к уничтожению. Манипуляция игровыми объектами разрешается только в пределах игрового поля и выход за его границы запрещен. За каждое удачное составление комбинации из элементов игроку начисляются очки. Доступно сохранение игрового процесса.

**А. В. Трухан**

*(ГрГУ им. Я. Купалы, Гродно)*

## **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ГЕНЕРАЦИИ ЛАНДШАФТОВ СРЕДСТВАМИ UNITY**

Целью данной работы является реализация следующих алгоритмов генерации ландшафтов средствами Unity [1]: шума Перлина в 2D, шума Перлина в 3D, алгоритма по созданию естественных туннелей. Перечисленные алгоритмы, программно-реализованные средствами языка программирования «C#» и игрового «движка» Unity позволят генерировать уникальные ландшафты с целью помочь гейм дизайнерам создавать игровые миры с наименьшими трудозатратами. При моделировании ландшафтов с помощью шума Перлина в 2D мы сначала генерируем карту высот, а затем передаем ее в метод, который создает текстуру и отрисовывает ее в цветном или черно-белом цвете. В случае программной реализации моделирования в 3D карта высот создается точно так же, как и в предыдущем пункте, однако ее визуализация происходит в трёхмерном пространстве. Программная реализация алгоритма по созданию естественных туннелей в толще земли позволяет, перемещаясь по координатам создавать клетки случайным образом выбирая вектор и условия движения до тех пор, пока