

Д. С. Лапицкий, А. Л. Казаков

(ГГУ им Ф. Скорины, Гомель)

РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ «СТРАТЕГИЯ»

ДЛЯ WINDOWS PHONE

Пакет Microsoft XNA, позволит разработчикам игр избежать многих технических трудностей, возникающих при написании кода, а также обеспечит существенное снижение стоимости конечной продукции. Кроме того, благодаря XNA программисты смогут создавать принципиально новые игры с высококачественной графикой.

Инструментарий XNA изначально разрабатывался с целью максимально облегчить процесс разработки игр для консоли Xbox и компьютеров, работающих под управлением операционной системы Windows. Вместе с тем, пакет XNA позволит распространить общие сервисы Xbox Live, такие как, например, аутентификация, на игры для персональных компьютеров. Инструментарий XNA позволяет Microsoft конкурировать с Sony на рынке видеоигр для следующего поколения игровых приставок.

Основная идея игрового приложения состоит в том, что все взаимодействие в приложении должно происходить с игровыми объектами исключая игровых персонажей. В разработанном приложении есть возможность полностью контролировать игровой процесс. Главной отличительной чертой которого является невозможность непосредственно управлять игровыми персонажами, а создание заданий происходит путем взаимодействия с окружением. Таким образом, оставляем в очереди задания, которые впоследствии выполняют освободившиеся персонажи. К особенностям можно отнести возможность добавления и извлечения из очереди заданий, добычу разнообразных ресурсов, и невозможность строительства различных сооружений при нехватке ресурсов.

Логика игрового приложения реализована при помощи метода матриц. Каждая ячейка матрицы содержит идентификатор ячейки, приоритет, параметр для обработки и делегат, содержащий метод для обработки данной ячейке. Матрица хранит список ячеек, очередь задач, стартовую позицию. Реализация логики приложения реализована в классе AllImplementation. В конструкторе данного класса к матрице добавляются новые ячейки с заданным ID, приоритетом и методом обработки. Данная ячейка проверяет ресурсы и, если необходимо, изменяет приоритет ячеек матрицы. Далее располагаются добавления ячеек, связанных со строительством зданий. Сначала проверяется наличие необходимых ресурсов для строительства. Далее преобразуем параметр типа object к типу List<House>. Вызовом функции AddHouse класса Builder заносим в переменную *h* экземпляр класса House.