

И.И. Коляскин, магистрант

*Научный руководитель – М.И. Жадан, к.ф.-м.н., доцент
Гомельский государственный университет им. Ф.Скорины*

Одной из технологий, упрощающих и ускоряющих разработку игр, является Microsoft XNA. Данный фреймворк включает в себя обширный набор библиотек классов, специфичных для разработки игр. Такие игры пишутся для среды времени выполнения .NET Framework на языке C#, поэтому они могут запускаться на поддерживающих его платформах (Windows, Windows Phone и Xbox).

В целях изучения возможностей XNA Framework было создано игровое приложение, представляющее собой головоломку, в которой игрок должен найти среди звезд заданное созвездие. В игре имеется несколько уровней различной сложности. Каждый уровень содержит определенное количество созвездий для поиска. Форма и расположение созвездий близки к реальным. Игра содержит множество декоративных элементов, такие как туманности на фоне, изменяющие форму, яркость и цвет, иллюстрации ко всем созвездиям, фоновые звезды, а различные анимации.

Рассмотрим основы работы с XNA на основе описанного приложения. В заранее подготовленную папку Content добавляются игровые ресурсы: картинки фона, игровых объектов, таких как звезды, связи между ними и т. д.

Для описания игровых сущностей созданы специальные классы. Эти классы содержат лишь свойства и конструкторы и предназначены для хранения данных. Перечислим сущности, необходимые в игре:

– Location – содержит название, индекс уровня сложности, список уровней и значение зума, или же приближения для камеры;

– LevelInfo – содержит название уровня, или его номер, позицию камеры, представленную как экземпляр класса Vector2 и список созвездий;

– Constellation – содержит название созвездия, список звезд, список связей, а также позицию иллюстрации для созвездия, которая будет отображаться, когда игрок отгадает его;

– Star – содержит Id, представляющий собой порядковый номер звезды в своем созвездии, размер, тип, определяющий цвет звезды и позицию;

– Relation – содержит значения свойства Id двух звезд.

Так же создано перечисление StarKind, для описания типа звезды, экземпляры которого могут принимать четыре значения: Yellow, White, Blue и Red.

Для хранения данных о локациях используются XML-файлы, имеющие структуру описанных выше сущностей. Один файл содержит описание одной локации. Поскольку объем информации, необходимой для описания локаций, слишком велик, был создан редактор уровней, который позволяет быстро создавать уровни, добавлять звезды, связывать их, образуя созвездия, и устанавливать прочие необходимые параметры. Редактор уровней был создан на технологии Windows Forms (Рисунок 1).

Создание логики игры начинается с переопределения класса Game. В этом классе должны быть переопределены основные методы, такие как Initialize, LoadContent, UnloadContent, если требуется, и, самое главное, методы Update и Draw. В методе Initialize задаются главные настройки, касающиеся всей игры. Например, разрешение экрана, инициализация игровой камеры, менеджера ресурсов, который будет предоставлять текстуры и прочие ресурсы в дальнейшем и т. д. В методе LoadContent в навигатор по страницам загружаются игровые страницы. В Update вызываются методы Update навигатора по страницам и менеджера пользовательского ввода, а также вызов метода Update базового класса Game. В методе Draw вызывается метод Draw навигатора по страницам, который прорисует текущую страницу.

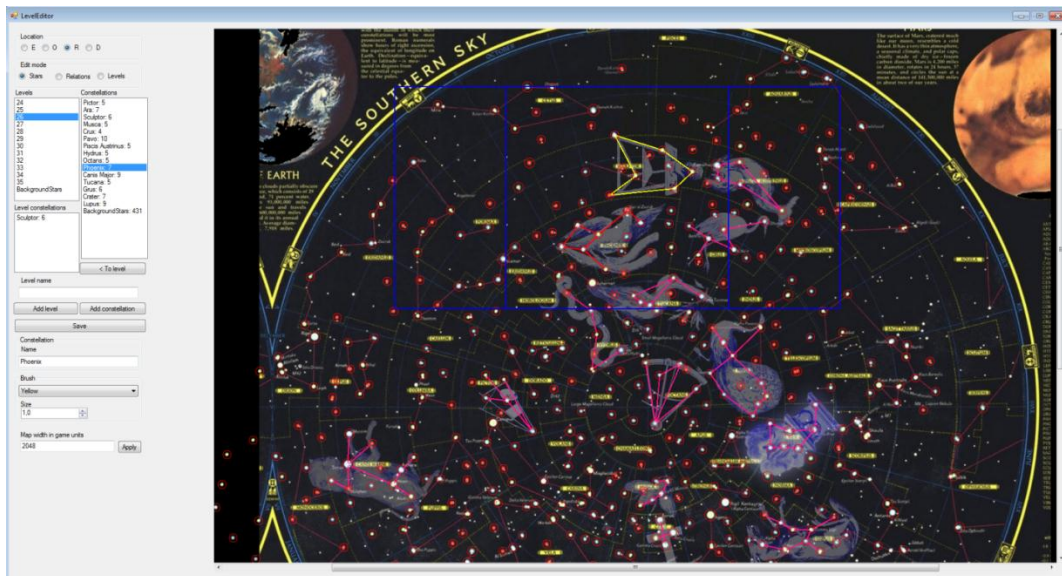


Рисунок 1 – Редактор уровней

Приложение может содержать несколько страниц. Например, на одной может содержаться главное меню, на второй настройки, на третьей само игровое поле. Для переключения страниц создан класс Navigator, который содержит список страниц и хранит текущую активную страницу. В приложении предусмотрено три страницы – главное меню, меню выбора одной из четырех локаций и игровой экран. Страницы с меню и выбором локации содержат кнопки, направляющие игрока на другие страницы либо изменяющие какие-либо параметры. Каждой кнопке необходимо задать такие параметры как позицию, размер, изображения для различных состояний и назначить обработчики событий для взаимодействия. В обработчиках события описываются переходы на другие страницы или еще какие-нибудь операции. Навигатор, как видно, кроме названия страницы для перехода позволяет указывать какие-либо параметры. Эти параметры будут обработаны вызываемой страницей. В приведенном примере вызывается страница с игровым экраном, а параметр указывает название локации, которую нужно загрузить. Игровой экран представлен на рисунке 2.



Рисунок 2 – Игровой экран

В методе Draw рисуется фоновое изображение, звезды, «связи» между ними и прочие игровые объекты. Обработка движения объектов, а также их реакция на действия пользователя происходит в методе Update. При вызове метода Update проверяется состояние кнопки мыши. Если она нажата, «связь» от выделенной звезды будет тянуться за указателем. Если указатель, тянущий «связь», находится над другой звездой, то при отпускании кнопки она зафиксирована вторым концом на этой звезде. Так происходит выделение созвездия. Во время выполнения программы методы Update и Draw вызываются с достаточно высокой частотой, что создает иллюзию плавного изменения картинки и мгновенную прорисовку изменений.

Microsoft XNA поддерживает лишь одну мобильную платформу – Windows Phone. Чтобы приложение обрело большую популярность имеет смысл сделать его кроссплатформенным, то есть способным запускаться на других не менее популярных мобильных платформах – Android и iOS. Для этих целей может быть использована технология Xamarin, которая позволяет компилировать программу на C# с минимальными изменениями под упомянутые выше платформы, используя кроссплатформенную реализацию .NET – Mono. Хотя чаще всего создание полностью кроссплатформенного приложения не представляется возможным, Xamarin сильно облегчает разработку игры, которую пользователи смогут запустить на Android, iOS и Windows Phone.