

## РЕАЛИЗАЦИЯ ИГРОВОГО ПРИЛОЖЕНИЯ ТИПА «СОЗВЕЗДИЕ»

*И.И. Коляскин, 5 курс*

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

Microsoft XNA – набор инструментов с управляемой средой времени выполнения (.NET), созданный Microsoft, облегчающий разработку и управление компьютерными играми. Пакет включает в себя обширный набор библиотек классов, специфичных для разработки игр и поддерживающий максимальное повторное использование кода. Игры XNA пишутся для среды времени выполнения, поэтому они могут запускаться на любой платформе, поддерживающей XNA Framework, что делает подобный набор универсальным, но официально поддерживаются только C# и XNA Game Studio Express и все версии Visual Studio 2005 и выше.

Наибольший интерес при создании игрового приложения, например, головоломки с поиском созвездия на небе, представляет файл Game1.cs, в котором определен класс Game1, наследованный от Microsoft.Xna.Framework.Game, где и разрабатывается приложение. В классе Game1 переопределены следующие методы Game:

- Initialize – вызывается для инициализации ресурсов до начала игры;
- LoadContent – используется для загрузки контента (спрайты и т.д.);
- UnloadContent – используется для выгрузки контента;
- Update – в этом методе реализуется логика игры, обработка событий клавиатуры или джойстика, проигрывание аудио и т.д.;
- Draw – вызывается для прорисовки игрового поля.

В папку Content добавляются игровые ресурсы: картинки фона, игровых объектов, таких как звезды, связи между ними, элементов интерфейса. При построении приложения ресурсы из Content конвертируются в формат xnb, и далее с этими ресурсами можно работать с помощью специальных классов, предусмотренных в XNA.

Во время выполнения программы методы Update и Draw вызываются с достаточно высокой частотой. Таким образом, получается плавное изменение положения объектов и мгновенную прорисовку произошедших изменений.

Созданы классы для сущностей, необходимых для игры. Эти классы содержат лишь свойства и конструкторы и предназначены для хранения данных о сущностях. Перечислим сущности, необходимые в игре и их реализацию (без конструкторов).

- 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).

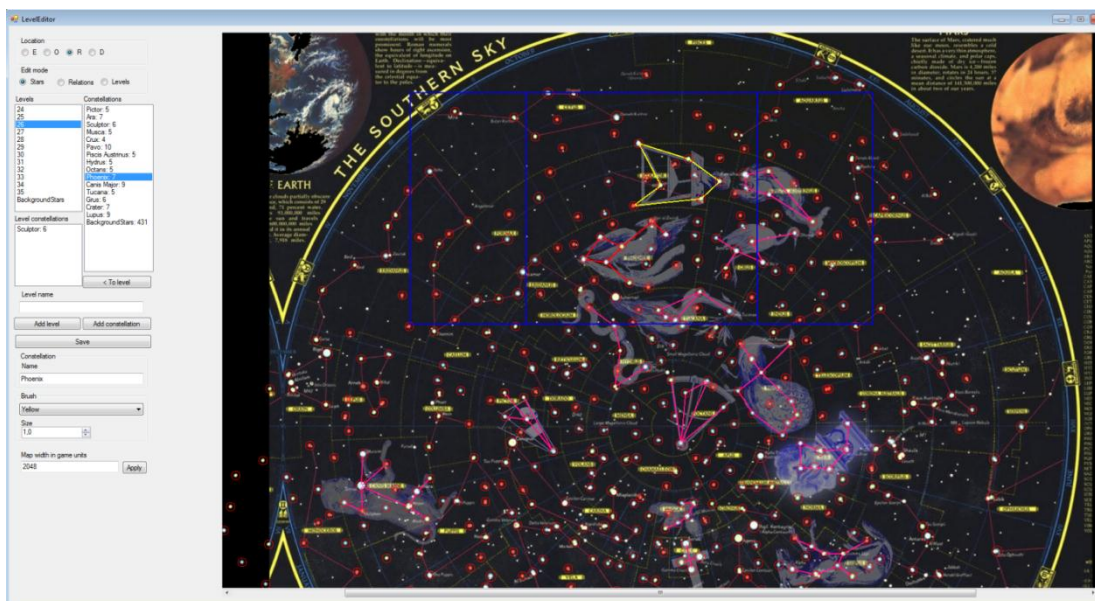


Рисунок 1 – Редактор уровней

Создание самой 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 вызовы методов Update базового класса Game. В методе Draw вызывается метод Draw навигатора по страницам, который прорисует текущую страницу.

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. Для этого класса описаны обработчики событий, которые описывают действия, которые должны быть выполнены при смене страницы или выходе. Так же класс навигатора содержит методы Update и Draw, которые будут вызываться из соответствующих методов класса игры и вызывать в свою очередь соответствующие методы текущей активной страницы.

В приложении предусмотрено три страницы – главное меню, позволяющее начать игру заново, продолжить с момента, где игра остановилась в прошлый раз и выйти из игры; меню выбора одной из четырех локаций; игровой экран, на котором и нужно искать созвездия. При запуске игры открывается страница, содержащая главное меню.

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, которых может быть различное число. Эти параметры будут обработаны вызываемой страницей. В приведенном примере вызывается страница с игровым экраном, а параметр указывает название локации, которую нужно загрузить. Экран выбора локации представлен на рисунке 2.



**Рисунок 2 – Экран выбора локации**

В результате работы программно реализована игровая ситуация головоломки с поиском созвездия на небе. Во время выполнения программы методы Update и Draw вызываются с достаточно высокой частотой, что дает плавное изменение положения объектов.