

Таким образом, на основе векторной модели коррекции ошибок выявлено, что долгосрочные рыночные механизмы корректировок курса российского рубля по отношению к доллару США и цен на нефть, золото, серебро, платину асимметричны.

### Литература

1. Центральный банк Российской Федерации [Электронный ресурс] – Режим доступа: [http://www.cbr.ru/currency\\_base/](http://www.cbr.ru/currency_base/) – Дата доступа: 04.05.2015
2. Обзоры цен на нефть и металлы [Электронный ресурс] – Режим доступа: <http://fx-commodities.ru/> – Дата доступа. – 1.02.2015.
3. Носко, В.П. Эконометрика. Введение в регрессионный анализ временных рядов / В. П. Носко. – Москва, 2002. – 252 с.

УДК 004.7

*И. И. Коляскин, М. И. Жадан*

### РАЗРАБОТКА ПРИЛОЖЕНИЙ НА ОСНОВЕ MICROSOFT XNA FRAMEWORK

*Статья рассказывает об основных этапах и принципах создания игровых приложений с использованием технологии Microsoft XNA на языке программирования С# в среде разработки Microsoft Visual Studio на конкретном примере. Используя возможности перечисленных технологий и продуктов, создано несложное игровое приложение, которое впоследствии легко может быть собрано под наиболее распространенные платформы — Windows Phone, Android и iOS.*

Microsoft XNA – набор инструментов с управляемой средой времени выполнения (.NET), созданный Microsoft, облегчающий разработку и управление компьютерными играми. XNA стремится освободить разработку игр от написания «повторяющегося шаблонного кода» и объединить различные аспекты разработки игр в одной системе.

Поддерживаются платформы Windows, Microsoft Xbox 360 и Microsoft Zune. Теоретически писать можно на любом .Net языке, в любой IDE, но официально поддерживаются только С# и XNA Game Studio Express и все версии Visual Studio 2005 и выше. XNA также дает возможность портировать игры на поддерживаемые платформы с минимальными изменениями.

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

XNA Framework скрывает низкоуровневые технологические детали, связанные с разработкой игры. Таким образом, фреймворк заботится о разнице между платформами, позволяя разработчикам уделять больше внимания смысловому содержанию игры. XNA Framework предоставляет поддержку создания и двухмерных, и трёхмерных игр и позволяет использовать возможности контроллеров Xbox 360.

Основой игры является класс Microsoft.Xna.Framework.Game, где и будет разрабатываться игра. В классе Game определены следующие методы:

- void Initialize() – вызывается единожды, для инициализации ресурсов до начала игры;
- void LoadContent() – вызывается единожды, используется для загрузки контента (спрайты и т. д.);

- void UnloadContent() – вызывается единожды, используется для выгрузки контента;
- void Update(GameTime gameTime) – в этом методе реализуется собственно логика игры, обработка коллизий, обработка событий клавиатуры или джойстика, проигрывание аудио и т. д.;
- void Draw(GameTime gameTime) – вызывается для прорисовки игрового поля.

Для создания приложения с помощью Microsoft XNA Framework необходимо установить его, после чего в Microsoft Visual Studio станет доступен новый вид создаваемого проекта – Windows Game. Такой проект сразу же будет содержать ссылки на компоненты XNA Framework и класс-наследник класса Game, который и будет описывать игру.

Ниже будет описан процесс создания игрового приложения на технологии XNA Framework. Игра представляет собой головоломку, где игрок должен находить созвездия, задаваемые в начале каждого уровня. Игровой экран представляет собой небо со звездами, а игроку дается возможность соединять эти звезды, образуя созвездия.

Для описания игровых объектов создаются классы «сущностей». Эти классы содержат лишь свойства и конструкторы и предназначены для хранения данных об этих самых сущностях. Перечислим сущности, необходимые в игре и их реализацию (без конструкторов):

- Location (локация), которая будет содержать название, индекс уровня сложности, список уровней и значение зума для камеры. В игре будет четыре локации, в каждой из которых будет несколько уровней.

- LevelInfo (уровень), которая содержит название уровня, или его номер, позицию камеры, представленную как экземпляр класса Vector2 и список созвездий.

- Constellation (созвездие), которая содержит название созвездия, список звезд, список связей, а также позицию иллюстрации для созвездия, которая будет отображаться, когда игрок отгадает его.

- Star (звезда), которая содержит Id, представляющий собой порядковый номер звезды в своем созвездии, размер, тип, определяющий цвет звезды и позицию.

- Relation (отношение или связь), которая содержит значения свойства Id двух звезд, которые объединены это связью.

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, чтобы создавать её вручную, 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). Для создания созвездий использовались реальные карты северного и южного полушария небесной сферы.

Редактор уровней содержит собственные модели для представления сущностей. Поэтому для него описан так называемый маппер, который переводит экземпляр одного класса в другой. Однако редактор должен сохранять данные в XML-файл в том же формате, в котором их будет использовать игра. Для этого были описаны классы для загрузки и сохранения информации о локациях общие для игры и редактора.

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, такие как главное

меню, меню выбора локации и игровой экран. 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, и вызов метода Draw базового класса Game.

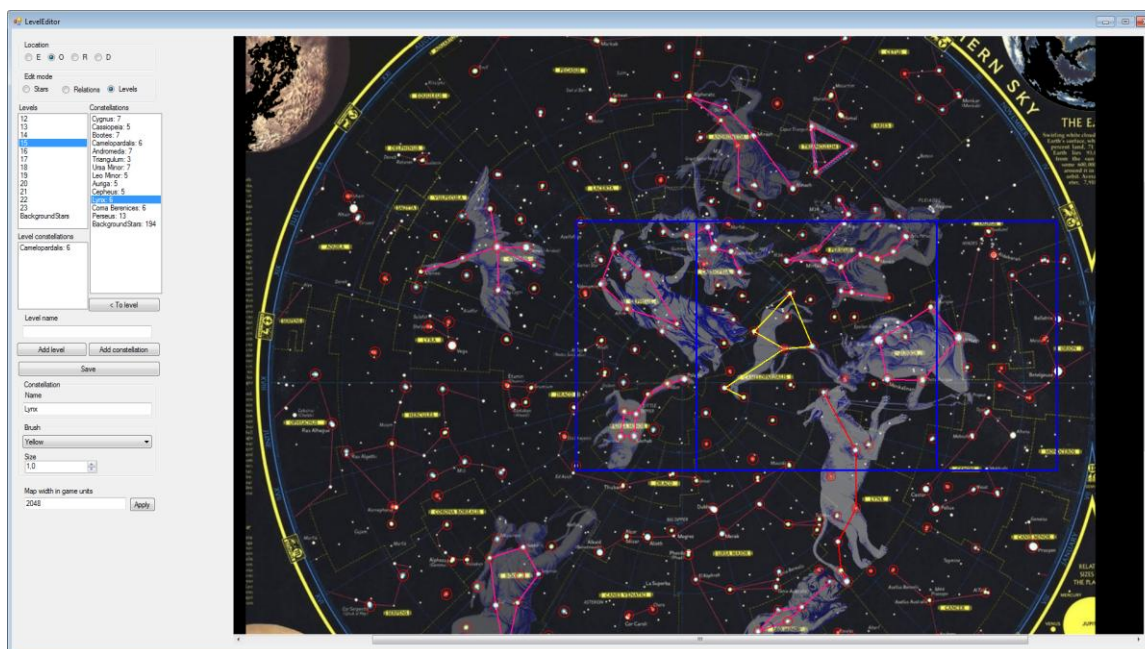


Рисунок 1 – Редактор уровней

Игровой движок управляет процессом игры. Он содержит все игровые объекты и задает логику их обновления и прорисовки. В нашем случае игровой движок содержит локации. Так же у него есть такие поля как камера, фон, определяющий логику прорисовки различных декоративных фоновых объектов, мини-карта, которая отображает задание на текущий уровень, а также загрузчик и поставщик ресурсов для получения текстур всех объектов, которые нам понадобятся. Движок содержит событие GameOver, которое происходит, когда все созвездия отгаданы, а также обработчики событий игровой страницы, которые обрабатывают пользовательский ввод. Движок определяет, что делать при сбросе прогресса, при нажатии на соответствующую кнопку и управляет игровой камерой. В методах движка Update и Draw, вызываемых из соответствующих методов игровой страницы, вызываются эти же методы для камеры, локации, мини-карты и фона.

Остальная логика игры инкапсулирована в классе локации Location. Этот класс содержит список уровней, события, срабатывающие при отгадывании созвездия, окончании уровня и окончании последнего уровня, список звезд и связей всех уровней локации, логику перехода между уровнями и проверку на правильность связей, нарисованных пользователем, а также обработчики пользовательского ввода, которые вызываются из игрового движка. В методах Update и Draw вызываются эти же методы для уровней локации, а также проверяется правильность нарисованных связей между звездами текущего уровня.

Класс уровня, Level, содержит список звезд текущего уровня, список связей, необходимых для прохождения уровня, список связей, нарисованных пользователем, список созвездий данного уровня, список иллюстраций к созвездиям данного уровня. Этот класс содержит внутри себя сущность уровня, считываемую из хранилища, и в свою очередь добавляет к ней игровую логику и прорисовку. В методах Update и Draw

вызываются соответствующие методы для связей, звезд и иллюстраций к созвездиям, появляющимся после открытия заданного созвездия. Во время выполнения программы эти методы вызываются с достаточно высокой частотой, что дает плавное изменение положения объектов.

Аналогично классу Level, классы звезд Star и связей Connection содержат внутри себя сущность, считываемую из XML-файла с данными, и дополняют это прорисовкой. Класс звезды прорисовывает сияние над звездой и управляет его анимацией.

Класс иллюстрации, Illustration, создан в декоративных целях. Иллюстрация появляется, когда отгадано созвездие. Этот класс управляет анимацией иллюстрации – она плавно изменяет форму и цвет.

Также создан такой игровой объект как LevelConstellation. Этот класс содержит внутри себе сущность созвездия и добавляет поле типа bool, которое указывает на то, отгадано ли созвездие. Это сделано для возможности создания уровней с несколькими созвездиями.

К игровым объектам можно отнести класс Background, который управляет фоновыми объектами, такими как туманности и фоновые звезды. Эти классы созданы исключительно для декорирования игрового поля, но они делают игру гораздо более приятной на вид, а значит и более интересной. Класс Background обновляет и прорисовывает все фоновые объекты, которые в свою очередь определяют логику этой прорисовки и собственную анимацию, если таковая имеется.

На рисунке 2 отображены перечисленные игровые объекты, содержащие логику прорисовки: звезды, связи, иллюстрации, туманности фиолетового цвета, фоновые звезды, которые значительно меньше игровых, а также мини-карта с заданием в нижней части экрана.



Рисунок 2 – Игровые объекты

На данный момент приложение собрано под платформу Windows Phone и находится в стадии тестирования, после чего оно будет размещено в Windows Phone Store, откуда его сможет скачать и установить любой пользователь смартфона на соответствующей платформе.