

Выбор универсального средства разработки игровых приложений

Иванов Е. В.¹, Зарипов Б. М.², Захаров В. А.³

¹Иванов Егор Витальевич / Ivanov Egor Vital'evich – студент;

²Зарипов Булат Маратович / Zaripov Bulat Maratovich – студент;

³Захаров Виктор Альбертович / Zaharov Viktor Albertovich – старший преподаватель,
кафедра машиностроения и информационных технологий,

Зеленодольский институт машиностроения и информационных технологий (филиал)
Казанский национальный исследовательский технический университет имени А. Н. Туполева,
г. Зеленодольск, Республика Татарстан

Аннотация: в статье освещается проблема выбора средств разработки игровых приложений. Рассматриваются достоинства и недостатки одной из популярных существующих систем Monogame и приводится пример собственной разработки в этой среде. Также кратко приводятся примеры известных программ других разработчиков, которые использовали данную среду.

Ключевые слова: программирование, компьютерные игры, мобильные приложения, Monogame, фреймворк.

В наше время количество ноутбуков, компьютеров, приставок, планшетов и смартфонов у людей непрерывно растет. Осмотритесь вокруг себя – можно поспорить, что Вы заметите хотя бы одно такое устройство. И не то чтобы с помощью него люди обязательно делают сложные расчеты, нет, чаще всего они просто играют. Именно поэтому речь пойдет об играх и технологиях, с помощью которых они разрабатываются.

Если Вы, как программист, когда-нибудь были заинтересованы в создании собственной игры, то перед Вами наверняка вставал вопрос – с чего начать разработку будущей игры? Какие средства разработки выбрать? Каким должен быть их порог освоения? Важно ли наличие открытого исходного кода? Должно ли быть игровое приложение кроссплатформенным? Есть ли активное сообщество разработчиков, с которыми можно проконсультироваться? Возможна ли продажа будущего приложения? Вопросов появляется множество и важно сделать правильный выбор.

В настоящее время хорошим решением может быть фреймворк Monogame, который ведет свою историю от Microsoft XNA.

XNA – набор инструментов с управляемой средой времени выполнения (.NET), который был выпущен Microsoft в 2006 году. Он облегчает разработку и управление компьютерными играми. XNA стремилась освободить разработку игр от написания «повторяющегося шаблонного кода» и объединить различные аспекты разработки игр в одной системе. Основывается на нативной реализации .NET Compact Framework 2.0 для разработки игр для Xbox 360 и .NET Framework 2.0 на Windows. Он включает обширный набор библиотек классов, специфичных для разработки игр, поддерживающий максимальное повторное использование кода на всех целевых платформах. Фреймворк выполняется на модификации Common Language Runtime, оптимизированной для игр, чтобы предоставить управляемую среду выполнения. Среда времени выполнения доступна только для Windows XP, Windows Vista, Windows 7 и Xbox 360. Так как игры XNA пишутся для среды времени выполнения, они могут быть запущены на любой платформе, поддерживающей XNA Framework с минимальными изменениями или вообще без таковых. Игры официально разрабатывались на языке C#.

XNA оказался довольно удобным инструментарием для разработки игр с высококачественной графикой и обеспечивал существенное снижение стоимости конечной продукции.

XNA был единственной технологией, официально поддерживаемой для игр на Windows Phone 7, а также был единственной возможностью для инди-разработчиков разрабатывать игры на Xbox. Но позже, с выходом Windows Phone 8, сообщество разработчиков изъявило желание разрабатывать игры не на XNA, а на C++ DirectX. Так базовой игровой технологией Windows Phone 8 стал C++ DirectX, а поддержка XNA прекратилась.

Однако XNA успел накопить большое сообщество людей, которым он пришелся по вкусу. И когда в Microsoft заявили, что поддержка XNA будет прекращена, это сообщество очень огорчилось. И тогда разработчики обратили взгляд на такой продукт, как Monogame.

Monogame – это open-source реализация XNA. Она кроссплатформенна – доступна на платформах iOS, Android, MacOS, Linux, OUYA, PS4, PSVita, Xbox One и на всех платформах Windows. На данный момент в ней есть все, что было в XNA и даже больше.

Разработка Monogame началась в 2009 году под названием XNA Touch, целью которой было портирование простых 2D-игр с XNA на мобильные устройства. Она основывалась на базе кода Mono.XNA и SilverSprite. В том же 2009 разработка увидела свет для устройств iPhone.

Полная версия Monogame появилась в марте 2011 года и была опубликована на GitHub. На тот момент она имела поддержку Android, MacOS, Linux и OpenGL на Windows, а позднее, в раннем 2012, стала доступна поддержка Windows Phone 8 на DirectX 11 [1].

Фреймворк имеет множество интересных возможностей, которые позволяют быстро приступить к работе с ним. Вот некоторые из них:

- Управляемый код: используя C# и другие .NET языки на платформах Microsoft и Mono, Вы можете писать современный, быстрый и надежный код игры.
- Все что нужно для разработки игры: готовые классы для работы с текстурами (.jpg, .png, .tiff), с 3D-моделями (.obj, .fbx, .3ds), со звуками (.wav, .wma) и с шейдерами на языках HLSL и GLSL.
- Встроенные препроцессоры игрового контента и возможность написания собственных.
- Работа в MonoDevelop/Xamarin Studio, Visual Studio.
- Бесплатен. Приложения, написанные на этом фреймворке, могут распространяться под лицензией MS-PL.
- Активное сообщество разработчиков.
- Постоянные обновления, а также анонсы будущих.
- Открытый исходный код.

С помощью Monogame разработано немало известных игровых приложений, например:

1. Transistor – научно-фантастическая игра в жанре Action RPG от создателей Bastion, в которой Вам предстоит путешествовать по завораживающему футуристическому городу, защищая свою жизнь при помощи невероятного оружия неизвестного происхождения.
2. Skulls of the Shogun – пошаговая стратегия с ограниченным временем хода, в которой полно сражений и в два раза больше серьезного чувства юмора. Игроку предстоит познать жизнь самурая после смерти, собрать армии неуспокоенных воинов, волшебных монахов и могучих усатых самураев-генералов.
3. Bastion – ролевая игра с принципиально новым взглядом на истории в играх – спасибо рассказчику, реагирующему на все действия игрока. На пути к разгадке тайны Бедствия – странной катастрофы, разорвавшей мир на части, героя ждет более сорока ярких, нарисованных от руки мест [2].

Благодаря использованию Monogame, большинство из этих игр доступны на платформах Windows, Xbox 360, OS X и Linux.

Принципы разработки

Приложение на Monogame отличается от привычного событийного приложения – в играх не следует немедленно реагировать на каждое действие, так как это практически невозможно выполнять в реальном времени. Вместо этого предлагается следующая структура.

Изначально, на этапе компиляции приложения, идет работа контент-процессора – все игровые файлы из проекта конвертируются в удобный для Monogame формат, чтобы не делать это в момент запуска игры.

Затем игровое приложение инициализируется – проводит различные расчеты, устанавливает нужное разрешение, режим отображения и т.д. После проходит загрузка в память скомпилированных ресурсов игры: текстур, звуков и моделей.

Далее выполняется игровой цикл, 60 раз в секунду:

1. Обновление состояния игры – это выполнение метода Update, куда помещается вся логика приложения.
2. Обновление экрана – это выполнение метода Draw, в котором можно написать отрисовку каких-либо игровых объектов.
3. Конец -> К шагу 1.

Если есть необходимость, можно выгрузить неиспользуемый контент, например, при переходе на следующий уровень игры.

Опыт разработки

Для лучшего понимания возможностей фреймворка было решено разработать собственную 2D-игру жанра файтинг.

Рисовать что-либо в игре на Monogame можно довольно простыми способами. Для начала инициализируется класс прямоугольник, равный размерам окна, а затем он заполняется текстурой, загруженной в методе LoadContent. Выглядит это так:

```
private Texture2D TEXTURE_BACKGROUND; // объявление текстуры
private Rectangle m_rectWallpaper; // объявление прямоугольника
В методе LoadContent:
```

```
m_rectWallpaper = new Rectangle(0, 0, graphics.PreferredBackBufferWidth, graphics.PreferredBackBufferHeight);
// прямоугольник с размерами окна
TEXTURE_MENU = Content.Load<Texture2D>("Background"); // фоновая текстура.
```

И в методе Draw:

```
spriteBatch.Begin(); // открыть блок шейдера
spriteBatch.Draw(TEXTURE_BACKGROUND, m_rectWallpaper, Color.White); // рисовать текстуру на фоне
spriteBatch.End(); // закрыть
```

Теперь в игре появился фон. Остается нарисовать игроков и части интерфейса, например, показатели здоровья игроков.

Сам игрок должен быть анимирован – это значит, что нужно рисовать разные кадры текстуры в разных ситуациях. В Monogame, к сожалению, нет средств для загрузки и отображения анимированных текстур, поэтому пришлось разработать собственный класс для этой цели. Данный класс предназначен для того, чтобы менять кадры текстуры по истечении определенного времени.

Поскольку в Monogame, в отличие от XNA, нет каких-либо дополнительных программ, кроме менеджера контента, анимации были разработаны в отдельной программе. На рисунке 1 приведен эпизод игры, где правый игрок атакует, а левый – защищается.

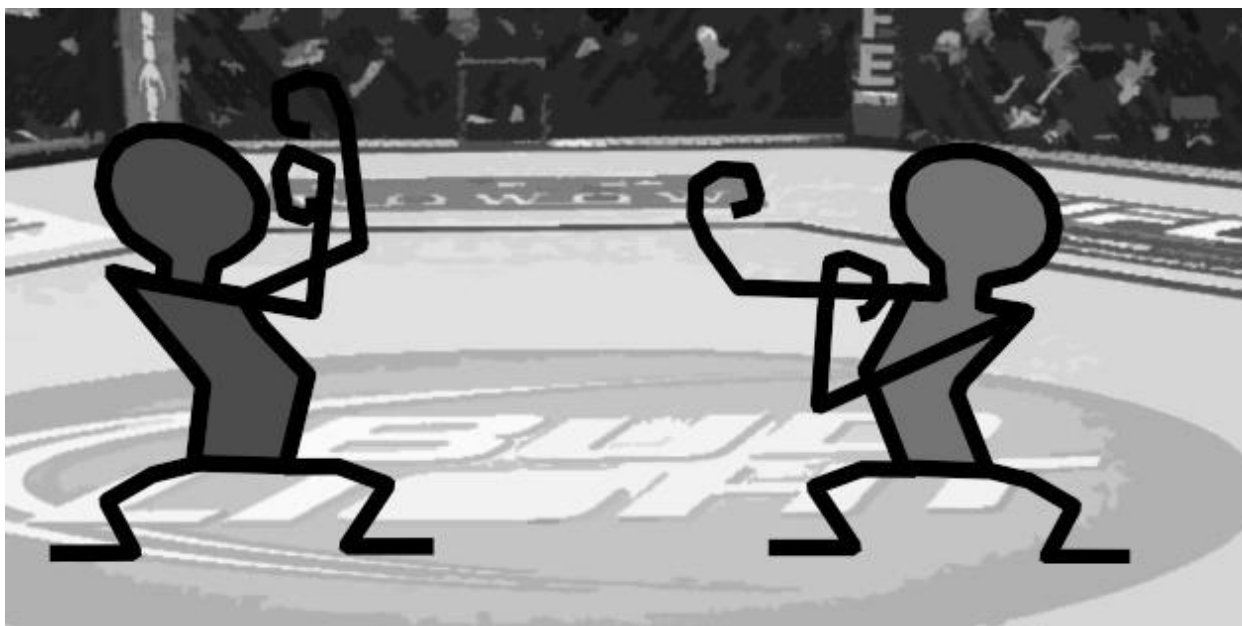


Рис. 1. Момент игры

Также нужно написать код для управления игроками. По нажатию на определенные клавиши каждый из игроков может двигаться вперед-назад, наносить удар или обороняться. В Monogame это делается в функции Update. Например, код по нажатию клавиши A (левый игрок отходит назад):

```
if (newKeyboardState.IsKeyDown(Keys.A)
    && oldKeyboardState.IsKeyUp(Keys.A)) { // код }
```

Было найдено несколько бесплатных звуков и композиций с последующим их добавлением в игру. Предварительно было необходимо переconvertировать их в формат .wav, поскольку .mp3 или .ogg не поддерживаются.

В итоге получилась небольшая игра с использованием несложной логики, музыки, звуков и анимированных текстур. Игра также была протестирована на операционной системе Linux. Опыт написания данной игры подтверждает достоинства фреймворка Monogame.

Литература

1. Официальный сайт Monogame. [Электронный ресурс]. Режим доступа: <http://monogame.com/> (дата обращения: 20.11.2016).
2. Официальный сайт дистрибьютора игровых программ Steam. [Электронный ресурс]. Режим доступа: <http://store.steampowered.com/> (дата обращения: 20.11.2016).