

РАМКА ЗА МОБИЛНИ ЛОГИЧЕСКИ ИГРИ

Николай Павлов¹, Ангел Голев², Асен Рахнев³

^{1,2,3} Факултет по математика и информатика, Пловдивски университет, България
² angelg@uni-plovdiv.bg

FRAMEWORK FOR MOBILE LOGIC GAMES

Nikolay Pavlov¹, Angel Golev², Asen Rahnev³

^{1,2,3} Faculty of Mathematics and Informatics, University of Plovdiv, Bulgaria
² angelg@uni-plovdiv.bg

Abstract. The specific of the mobile games market establishes a model where the main goal is implementation of multiple ideas for a minimal period of time and with minimal resources. This paper presents a software framework for development of logic games for the Microsoft Windows Phone 7 mobile platform. A prototype of the framework is developed. The applicability of the framework in real life is proven by a set of five games created and already available on Windows Phone Marketplace.

Key words: framework, logic game, applications for mobile devices

Въведение

Рамка за разработка на приложения е преизползваемо, „полуготово” приложение, което може да се специализира в производството на потребителски приложения (Johnson, 1988).

Преизползваемостта е основно качество на софтуерните рамки. Състои се в изграждането на устойчиви интерфейси, които подобряват преизползваемостта, като дефинират компоненти с широко приложение, които могат да се използват за създаване на нови приложения. Преизползваемостта на рамките за разработка на приложения позволяват избягването на повторно създаване и тестване на решения с общо приложение за повтарящи се изисквания и предизвикателства при проектиране. Преизползваемостта на компоненти може да доведе до значителни подобрения в продуктивността на програмистите, както и да повиши качеството, производителността, надеждността и възможностите за връзка с външни системи.

Рамките за фирмени приложения се използват в широк спектър от приложни области (финансово инженерство, телекомуникации, производство (Bitter, 1993)) и са крайъгълен камък в структурата на бизнес организацията и бизнес процесите (Fayad, Natu, 1997). Те са по-скъпи за изграждане от първите два вида рамки за разработка на приложения, но се характеризират с високо ниво на възстановяване на направените инвестиции, тъй като чрез тях се разработват приложения за крайните клиенти, а рамките за системна инфраструктура и междинна интеграция са насочени изцяло към вътрешните проблеми на разработката на софтуер. Макар че последните са изключително важни за бързото разработване на висококачествен софтуер, обикновено

те не създават преки ползи за големите фирми. Затова често е по-изгодно закупуването им, отколкото да бъдат разработени вътрешно (Fayad, Hamu, 1997).

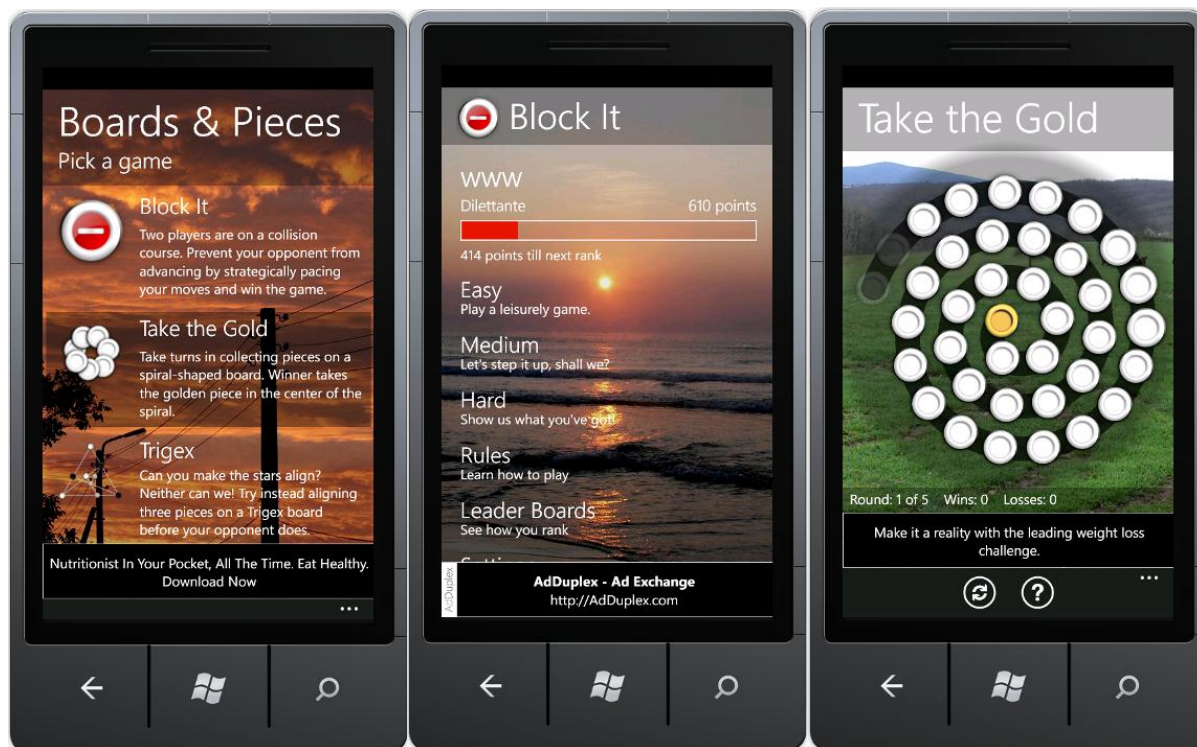
Прототип на софтуерна рамка

Създаден е прототип на софтуерна рамка за разработка на логически игри за мобилната платформа Microsoft Windows Phone 7 (Petzold, 2010). За конкретната реализация на програмите се използва Microsoft Visual Studio 2010 Express for Windows Phone и платформата Microsoft Silverlight.

Целта на софтуерната рамка е максимално бързо и удобно да се програмират логически игри (Касаткин, Владыкина, 1984) за игрална дъска. С използването на рамката конкретно са разработени:

- Две игри на Баше: **Block It**, в която при движение на двамата противници един към друг побеждава този, който направи последния ход и **Take the Gold**, в която целта е да се вземе последния златен жетон;
- Играта **Trigex**, която се играе на специална дъска и в която за победа трябва първи да подредим три жетона на линия;
- **Catch me if you can** (залови противника) е по-скоро игра закачка, в която на специална дъска трябва за заловим жетона на противниковия играч.
- Играта **Queens** – задачата за n-дами (царици). Потребителят трябва да разположи дамите на шахматна дъска, така че да не се заплашват.

Във всичките игри без **Queens** играчът се състезава против компютъра. За тази цел в логиката на конкретните игри са реализирани алгоритми, позволяващи на компютъра да избере ход, когато дойде неговият ред за игра.



Фигура 1. Изображения от пакета с игри

За реализацията на рамката са дефинирани абстрактни класове и интерфейси, включващи предполагаемите общи свойства на разработваните игри.

```
public abstract class BaseGameData
```

```

{
    public BaseGameScore FirstPlayerScore { get; set; }
    public BaseGameScore SecondPlayerScore { get; set; }
    public GameDifficulties GameDifficulty { get; set; }
    public int WinningPlayer { get; set; }
    public int PlayerNumberTurn { get; set; }
    public int CurrentRoundNumber { get; set; }
    ...
}
public class BaseGameScore : IComparable<BaseGameScore>
{
    readonly static List<Type> knownTypes = new List<Type>();
    ...
}
public interface IGameLogic
{
    GameStates GameState { get; set; }
    BaseGameData Data { get; }
    BaseGameData PreviousData { get; }
    IEnumerable<GameDifficulties> SupportedDifficulties { get; }
    void ResetGame(GameDifficulties difficulty);
    void RestoreGame(BaseGameData data);
    IGameMove GetNextMoveHint(int playerNumber);
    IEnumerable<IGameMove> GetPossibleMoves(int playerNumber);
    IGameMove GetNextComputerMove();
    bool CanMoveTo(int playerNumber, IGameMove nextMove);
    bool MoveTo(int playerNumber, IGameMove nextMove);
    void StartNextRound();
}
public interface IGameMove
{
}

```

Тези абстрактни класове и интерфейси се наследяват и предефинират от конкретните класове на всяка една игра. Например, за ***Catch me if you can*** имаме:

```

public class CatchMeGameData: BaseGameData
{
    public int[] GameBoard { get; set; }
    public const int BoardCells = 16;
    public List<BoardPosition> BoardPositions { get; private set; }
    public int PlayerPosition { get; set; }
    public int OpponentPosition { get; set; }
    ...
}
public class CatchMeGameMove: IGameMove
{
    public int PiecePosition { get; private set; }
    public int NewPiecePosition { get; private set; }
}

```

Някои от основните методи, които трябва да се реализират за всяка игра, са проверка дали играчът може да направи ход на конкретна позиция:

```

public bool CanMoveTo(int playerNumber, IGameMove nextMove)
{
    ...
}

```

В метода, в който се копира текущото състояние, се извършват някои приключващи операции за текущия ход и може да се промени статусът на играта преди да се предаде управлението на другия играч:

```

public bool MoveTo(int playerNumber, IGameMove nextMove)
{
    if (!CanMoveTo(playerNumber, nextMove)) return false;
    gameData.CopyTo(previousData);
    ...
}

```

Със следващия метод се реализира логиката на играта на компютъра:

```

public IGameMove GetNextComputerMove()
{
    ...
    return new CatchMeGameMove(..., ...);
}

```

Методът, с който се приключва играта при победа на единия от играчите или при равенство, е:

```

private bool SetGameWinner()
{
    bool result = false;
    ...
}

```

С използването на класовете на рамката се унифицира, улеснява и ускорява работата по създаването на нови логически игри. Софтуерната рамка не е сложна и позволява екипна работа при реализацията на игрите.

Изграждане на графичен потребителски интерфейс

За разработването на графичния интерфейс е използвана технологията Silverlight с eXtensible Markup Language (XAML). Приложена е комбинация от стандартни за Silverlight контроли и собствени графични ресурси. Анимациите са осъществени отново чрез Silverlight (Bugnion, 2010). За възпроизвеждане на звуковите ефекти е употребена техника от XNA, тъй като инструментът на Silverlight MediaPlayer има ограничена поддръжка на кратки звукови ефекти.

Графичното изображение е изградено на няколко нива. Най-отдолу имаме фон, след това е нивото на прозрачната игрална дъска, а отгоре се разполагат изображенията на обектите: жетони или фигури. За всеки обект имаме изображение и координати на екрана.

За посочването и преместването на изображения по екрана са предекларирани манипулаторите за обработка на всички движения с пръста по екрана на устройството.

Включени са и различни анимации при избор на обекти и групи от обекти.

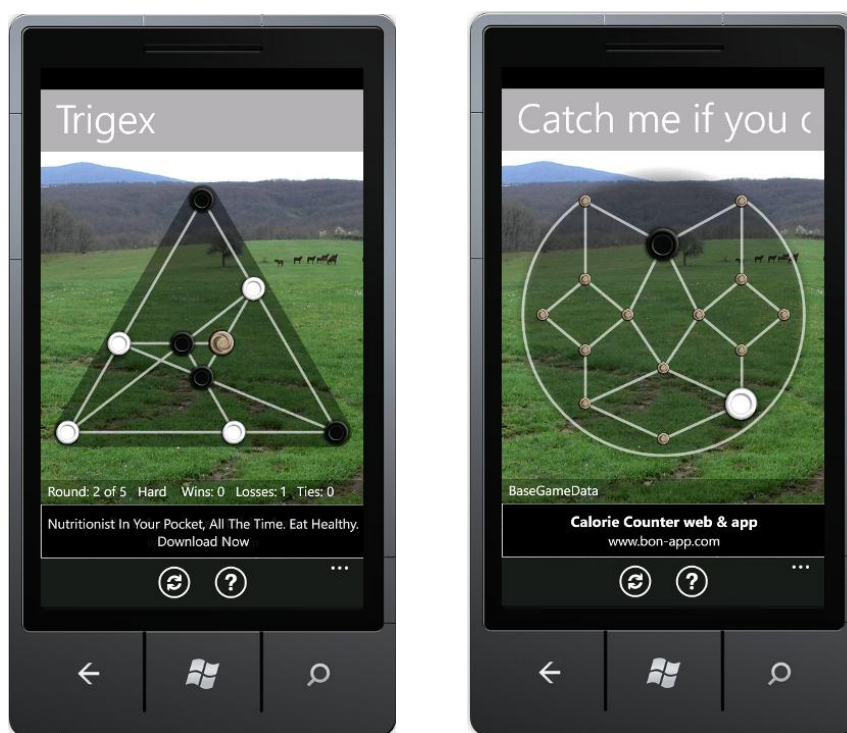
За реализацията на логиката на конкретна игра се използват различни структури от данни за описанието ѝ, така че имаме междинна връзка между логическата структура на играта и списъкът, описващ игралните позиции на дъската.

Логика на играта на компютъра

При изграждането и тестването на софтуерната рамка, първоначално са избрани логически игри с точно определена стратегия и такива, при които логиката за игра на компютъра се реализира лесно.

Най-общо, игрите на Баше се описват по следния начин. При всеки ход поредният играч може да взема от 1 до p предмета от една група. В началото на играта групата има N на брой предмета. Победител е този играч, който принуди противника да вземе последния предмет. В конкретната реализация победител е играчът, който взема последния предмет. За всяка игра N и p се задават предварително. Игрите на Баше са логически игри с безпогрешни стратегии. В тях точно може да се определи кой играч ще победи и какви да бъдат ходовете му.

За да не загуби интерес към игрите, на потребителя се предлагат три нива на сложност. Например, за най-слабото ниво печеливш ход от страна на компютъра се избира с вероятност 30%, за средно ниво – 60%, за нивото с най-висока трудност печелившия ход се избира с вероятност 85%. Вероятността не е 100%, понеже и потребителя трябва да има шанс за победа.



Фигура 2. Игралните полета на две от игрите

За играта **Trigex** имаме 9 позиции за жетони, следователно възможният брой на всички конфигурации е точно $9!$. Изборът на ход на компютъра става с помощта на **min/max** алгоритъм по възможните конфигурации. За всяко от свободните полета се прави оценка дали поставянето на жетон на тази позиция ще доведе до победа, равен или загуба. В зависимост от нивото на трудност избираме позицията за игра на компютъра с предварително зададена вероятност.

В *Catch me if you can* потребителят се стреми да залови жетона на компютъра. Лесно се реализира компютърът да „бяга” от жетона на противника към област с повече свободни полета.

При играта *Queens* няма реализирана логика за игра на компютъра, а само се маркират свободните полета, където потребителят може да постави дама.

Заклучение

Създаден е прототип на софтуерна рамка за разработка на логически игри за Microsoft Windows Phone, с помощта на която е реализиран пакет от няколко логически игри, публикуван в Windows Phone маркет. Чрез тази рамка в бъдеще ще бъдат създадени други логически и забавни игри. Една от възможностите за развитие е добавянето на режим мултиплейър, в който няколко играча едновременно ще могат да играят на една игра от различни мобилни устройства.

Благодарности

Това изследване е частично финансирано от проект НИ11-ФМИ004 към НПД на Пловдивски университет „Паисий Хилендарски”.

Литература

Касаткин, В., Владыкина, Л. Алгоритмы и игры. // Издательство „Радянська школа”, Киев, 1984.

Johnson, R., Foote, B. Designing Reusable Classes. // *Journal of Object-Oriented Programming. SIGS*, 1, 5 (June/July. 1988), p. 22-35.

Fayad, M., Hamu, D. Object-Oriented Enterprise Frameworks: Make vs. Buy Decisions and Guidelines for Selection. // *The Communications of ACM*, 1997.

Birrer, E. Frameworks in the Financial Engineering Domain: An Experience Report ECOOP '93 Proceedings. // *Lecture Notes in Computer Science nr. 707*, Springer-Verlag, 1993.

Troelsen, A. Pro C# 2010 and the .NET 4 Platform, 5th Edition. // Apress, 2010.

Petzold, Ch. Programming Windows Phone 7. // Microsoft Press, 2010.

Bugnion, L. Silverlight 4 Unleashed. // SAMS, 2010.