

## Приложение за логическа обработка на данните към играта HEARTHSTONE

Валентин Великов, Илия Мутафов

**Abstract:** *Application for a logical database processing to the game HEARTHSTONE. In this paper is present an application for a preliminary database processing, connected with the game HEARTHSTONE. The main part of time for the game is booked for a difficult logical preparation, with "pencil and paper", which can be assisted or replaced with a computer application.*

**Key words:** *computer application, game HEARTHSTONE, preliminary database processing.*

### ВЪВЕДЕНИЕ

Много игри на карти се състоят от два или повече етапа, като само финалният е самото разиграване на картите. В предварителните се извършват различни предварителни подготовки, чрез които се доуточняват правила на текущата игра, и по пътя на логиката прави се опит за проучване на противника, за установяване на неговите карти, на неговия опит, на начина му на мислене. Пример за подобно поведение са чистият бридж, бридж-белотът и др. Поради своята атрактивност част от игрите са пресъздадени и като компютърни.

Една известна игра на карти, с над милион и половина регистрирани в мрежата потребители, е HEARTHSTONE. Има множество нива, като основната част от времето е свързано с огромна предварителна логическа подготовка, където най-често с лист и молив се извършва обработка на известните данни. Оттук възниква и идеята за създаване на различни помощни средства и инструменти, с които да играчът да бъде подпомогнат.

### ИЗЛОЖЕНИЕ

#### 1. Базови правила на играта

Hearthstone: Heroes of Warcraft по всички показатели следва класическия принцип на Blizzard: играта е лесна за усвояване и трудна за усъвършенстване. Правилата са прости и бързи за усвояване [2][1]. На бойното поле се срещат два героя (класа), като целта е да се сваля противниковото здраве от 30 на 0. Изборът от герои е достатъчно голям: druid, hunter, mage, paladin, priest, rogue, shaman, warlock, warrior. Участието в двубой дава опит, чрез който героят вдига ниво. Вдигането на няколко нива отключва нови карти. Класовете се различават както по своето специално умение, така и по набор от карти, които могат да бъдат поставени само в тестето на съответния герой. Всяко тесте съдържа 30 карти, които играчът избира сам преди да влезе в битка. Различните герои предполагат и различен стил на игра, но не е задължително това да се използва. Пример: за уменията на rogue е характерна необходимостта от combo, т.е. да се изиграе поне още една карта, преди да се изиграе умението.

• В тестето си играчът може да постави до две копия от една и съща карта. Картите се делят в 5 групи, според своята рядкост: базисни, често срещани, редки, епични и легендарни. Особеност: няма карта, срещу която да няма десетки противодействия, независимо от нейната сила. Всяка карта спада към една от трите категории: оръжия, магии и minions.

#### 2. Проектиране на системата

Приложението (наречено "Deck Builder") е създадено за потребители и фенове на играта. Информацията и данните, които то ползва, са тясно свързани с нея или са взети директно от нея. Средно-статистическият потребител е човек, който играе играта от средно-голям период от време (няколко месеца), запознат е с механиките

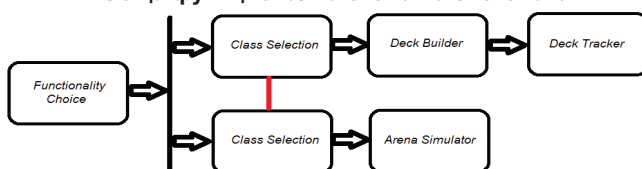
й, интересува се от иновативност или подобряване на игралната среда и иска да извлече всичко от наличните му средства за постигане на победа.

Предимствата на “Deck Builder” пред вградените функции на играта е, че то е доста „леко“ като приложение в сравнение с оригиналната игра, като има на разположение цялата база от данни с карти, съществуващи в играта, и всяка една от тях може да бъде използвана за създаване на тесте. За сравнение: в играта могат да бъдат използвани карти, които играчът вече притежава (т.е. или е закупил, или е отключил с предложени от играта средства) и не се предлага симулатор за създаване на бъдещо тесте. Друго ограничение, което е решено: авторите са поставили таван на максималния брой карти, които могат да се намират в тестето – 30. Това е проблем, тъй като най-добрите играчи изграждат стратегията си като добавят всички нужни според тях карти, и след това започват да премахват най-малко нужните според дадената ситуация, докато не достигнат нужния брой от 30.

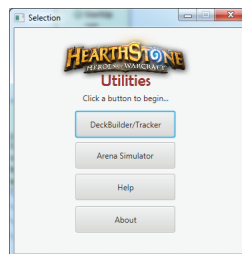
**2.1. Избор на среда за програмиране:** За разработване на приложението се използва JavaFX - набор от графични и медия пакети, предназначени за създаване на Rich Internet Applications (Богати Интернет Приложения), и е доразвитие на Java Swing. JavaFX SDK осигурява JavaFX Script APIs и компилатор за десктоп, браузър и мобилни приложения. Бъдещите издания на JavaFX ще осигуряват същите възможности и за TV платформи (например BlueRay дисковете)[3].

Това прави JavaFX подходящ избор за изграждане на системата.

## 2.2. Обща функционална схема на системата:



фиг. 1 Обща схема на системата



фиг. 2 Начален прозорец

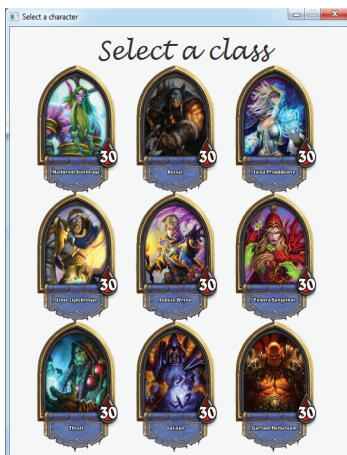
## 3. Потребителски интерфейс на системата

**3.1. Начален прозорец** - При стартиране на приложението, на екрана се появява малък прозорец (фиг.2), съдържащ само два бутона за избор между двете основни функционалности на програмата – Deck Builder/Tracker и Arena Simulator. Обликът на този прозорец се прочита от файла choice.fxml, и се управлява от класа ChoiceController.

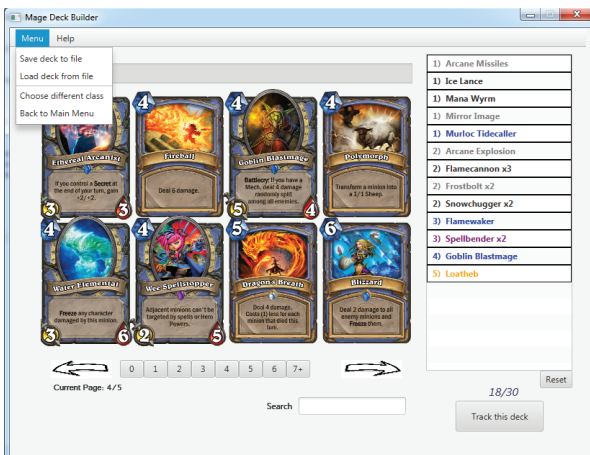
**3.2. Прозорец за избор на игрови клас** - Независимо от избора, направен в главния прозорец, се отваря прозорецът за избор на игрови клас (фиг.3). Той съдържа деветте класа „герои“, от които играчът има право на избор в Hearthstone: Druid, Hunter, Mage, Paladin, Priest, Rogue, Shaman, Warlock, Warrior. Всеки герой има свои собствени карти, които другите герои не могат да използват, но има и така наречените „неутрални“ карти, които са еднакви за всички герои. Оттук направеният избор определя до кои карти ще има достъп потребителят. Обликът на прозореца е във файла main.fxml и се управлява от класа CharController.

**3.3. Прозорец на функционалност Deck Builder** - При избор на опция Tracker в главния прозорец на приложението, и след избор на клас, се отваря прозорецът на функционалност Deck Builder. В случая, изобразен на фиг.4, е избран клас Mage.

В този прозорец се конструира тесте от наличните карти, което се подготвя за



фиг. 3 Прозорец за избор на игрови клас



фиг. 4 Прозорец на функционалност Deck Builder

използване от функционалността Deck Tracker, или се записва в текстов файл за бъдещо ползване (чрез Menu/Save deck to file). Потребителят може да се върне и в един от предишните два прозореца.

Конструирането става чрез избор на карти, представени като изображения в левия край на прозореца. Те са подредени по свойство „mana cost“ в няколко страници. Общият брой на страниците и номерът на текущата страница са изписани под левия долен ъгъл на албума с карти.

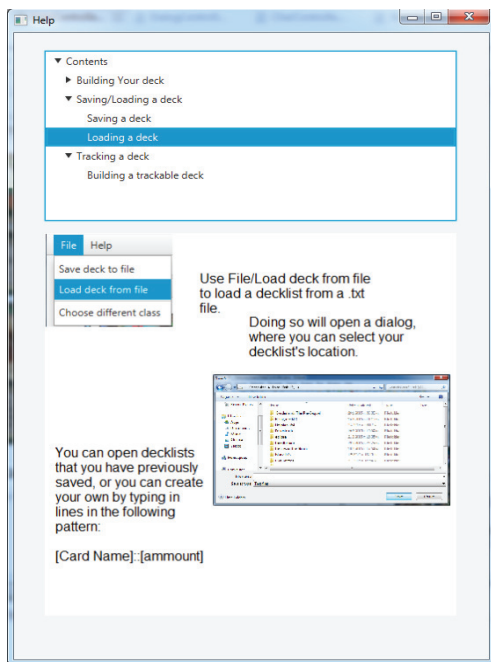
Смяната на страница на албума става чрез двата бутона с вид на стрелки, съответно наляво и надясно. Потребителят може да сменя между албума с картите на избрания клас и албума с неутралните карти по всяко време, и те са независими един от друг. Тъй като двата албума са сравнително дълги, съдържанието им може да се филтрира по 2 начина: чрез един от бутоните под албума по свойството „mana cost“ или чрез текстовото поле - по име на картата.

С щракване върху карта тя се добавя в списъка с карти в дясната страна на екрана. В него се съдържат имената на картите, техният mana cost и колко пъти се съдържат отделните карти, изобразени с цвят, съответстващ на тяхното свойство „rarity“. За премахване на карта от този списък се натиска върху името ѝ с ляв бутон на мишката, а за премахване на всички карти може да се използва бутон Reset. Под този списък е изписан броят на картите, добавени до момента. За да е валидно конструираното тество, то трябва да съдържа 30 карти.

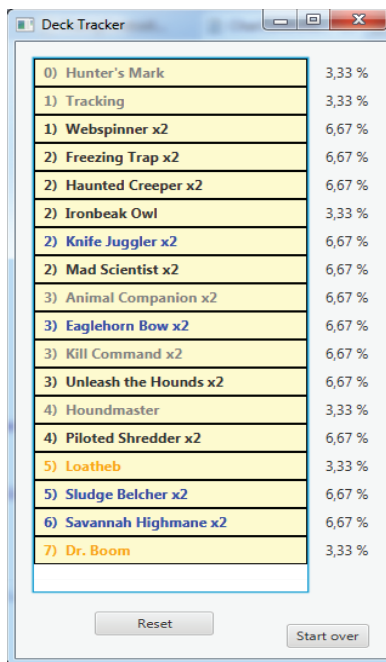
Освен по описания начин, тество може да бъде добавено и чрез отваряне на съществуващо от текстов файл, чрез използване на Menu/Load deck from file.

При наличност на валидно тество може да бъде използвана функционалност Deck Tracker за това тество, чрез натискане на бутон Track This Deck

**3.4. Прозорец на помощника** - Създаден е да подпомогне и да улесни работата на потребителя с приложението (фиг.5). Съдържа дървовидна структура, в която са разположени инструкциите, подредени по тяхната функционалност. Натискане с бутон на мишката върху родителски елемент от дървото активира неговите функции expand и collapse. Натискане върху дъщерен елемент показва изображение, съдържащо кратки инструкции за работа по съответната тема. Обликът на този прозорец се прочита от файл help.fxml и се управлява от клас HelpController.



фиг. 5 Прозорец на помощника



фиг. 6 Прозорец на функционалност Deck Tracker

**3.5. Прозорец на функционалност Deck Tracker** - Съдържа списък с картите на конструираното тесте от функционалността Deck Builder. Елементите на новия списък са копия на взетите елементи, за да бъдат независими от тях, и тяхната промяна да не влияе върху първичните данни.

До всяка карта от списъка има етикет (фиг.6.), показващ вероятността тя да бъде следващата изтеглена карта от играча по време на дуел в играта Hearthstone. Тази вероятност е представена като процент, преизчислява се след всяко изтегляне.

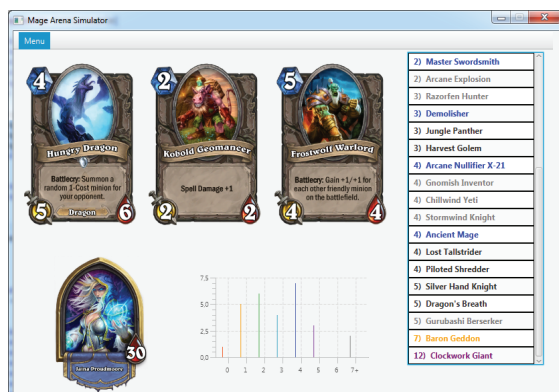
Играчът маркира изтеглена карта ръчно, чрез щракване с бутон на мишката върху името на изтеглената карта.

В началото всяка карта в списъка е оцветена с жълт цвят. След изтегляне на всички нейни копия, цветът се променя в тъмно червен за избягване на объркване.

След края на дуела, играчът може да нулира данните на тракера, като за целта използва бутон Reset, и да използва отново тракера в нов дуел със същото тесте.

Ако играчът смени тестето си, той трябва да го зареди или конструира наново през Deck Builder-a, чрез бутона Start Over. Обликът на този прозорец се прочита от файла list.xml и се управлява от класа ListController.

**3.6. Прозорец на функционалност Arena Simulator** -Тази функционалност симулира конструирането на тесте, по системата в играта Hearthstone на режим арена (фиг. 7). На играча се предоставят по 3 карти с еднаква рядкост (свойство „rarity“), от които той избира една и я добавя в тестето си. Това се повтаря 30 пъти до конструирането на цяло тесте. След това, в играта играчът трябва да играе с това тесте срещу опоненти, минали през същата процедура, докато загуби 3 мача, или не спечели 12, като получава награда в зависимост от броя на победите, които направи. Това приложение симулира елемента на конструиране на тесте, и се използва в подготвителни сценарии.



фиг. 7 Прозорец на функционалност Arena Simulator

до 1000, като всяко число попада в определен от горните числа интервал. Генерирането става до конструиране на пълно тестване, след което на играча се дава възможност за генериране на ново тестване със същия клас. Ако той предпочете, може да смени игровият клас или да се върне в главното меню чрез опциите от Menu в горния край.

Обликът на прозореца е в arena.fxml и се управлява от класа ArenaController.

## ЗАКЛЮЧЕНИЕ

Под формата на десктоп система с игрови елементи е създадено приложение, улесняващо логическата част при предварителната подготовка на картите за играта **HEARTHSTONE**. Приятният и интуитивен интерфейс, както и отчетените недостатъци при други реализации правят това приложение атрактивно и конкурентноспособно и може да бъде в помощ на над милион и половина регистрирани потребители на играта.

## ЛИТЕРАТУРА

- [1]. Adler, Alexander U., Hearthstone Card Rarity in the Arena (Original and Naxxramas), Alexadler.net, 25.03.2015, <http://blog.alexadler.net/?p=5>, 24.05.2015
- [2]. Митев, Георги, Hearthstone: Heroes of WarCraft - PCM2.0, 23.05.2014, <http://pcmania.bg/>, 24.05.2015
- [3]. Национална академия по разработка на софтуер, JavaFX – технология за създаване на RIA, HAPC, <http://nars.bg/newsletter/javafx/>, 24.05.2015

## За контакти:

гл. ас. д-р. Валентин Петров Великов, катедра "Информатика и ИТ", РУ "Ангел Кънчев", тел.: 082/888 326; GSM: 0886 011 544; e-mail: val@ami.uni-ruse.bg

бак. Илия Иванов Мутафов, кат. Информатика и информационни технологии", Русенски университет "Ангел Кънчев", GSM: 0884050191 e-mail: xchokleet@gmail.com

Трите генерирани карти имат определен процент вероятност да са от различна рядкост. За приложението са взети данни, изчислени от Алекс Адлер [1]. След направени експерименти той установява приблизителни шансове за всяка рядкост:

**Common:** 88.7%, **Rare:** 9.1%, **Epic:** 1.7%, **Legendary:** 0.4% [1]. За допълване до 100% шансът на рядкост Common е завишен до 88,8% (в арена Common се състои както от Common, така и от Basic карти). На практика това се постига чрез генериране на число от 1