

UDC 004.624

Kotlov V. N., Firsova S. A., Ryabukhina E. A. Creation of a software and information system for collecting and analyzing statistics of professional matches in the Dota 2 esports discipline

Создание программно-информационной системы для сбора и анализа статистики профессиональных матчей по киберспортивной дисциплине Dota 2

Kotlov Vyacheslav Nikolaevich

bachelor student

Federal State Budgetary Educational Institution of Higher Education

"N. P. Ogarev Mordovian State University"

Firsova Svetlana Anatolyevna

Candidate of Physical and Mathematical Sciences, Associate Professor

Federal State Budgetary Educational Institution of Higher Education

"N. P. Ogarev Mordovian State University"

Ryabukhina Elena Aleksandrovna

Candidate of Pedagogical Sciences, Associate Professor

Federal State Budgetary Educational Institution of Higher Education

"N. P. Ogarev Mordovian State University"

Котлов Вячеслав Николаевич

студент-бакалавр

Федеральное государственное бюджетное образовательное

учреждение высшего образования

"Мордовский государственный университет им. Н. П. Огарёва"

Фирсова Светлана Анатольевна

кандидат физико-математических наук, доцент

Федеральное государственное бюджетное образовательное

учреждение высшего образования

"Мордовский государственный университет им. Н. П. Огарёва"

Рябухина Елена Александровна

кандидат педагогических наук, доцент

Федеральное государственное бюджетное образовательное

учреждение высшего образования

"Мордовский государственный университет им. Н. П. Огарёва"

Abstarct. *The article discusses the creation of a prototype of a software and information system for collecting and analyzing statistics of professional matches in the Dota 2 esports discipline, which is a fully functional system for multi-level and detailed search for structured information obtained using the Stratz API.*

The presented prototype provides an opportunity to obtain data on the effectiveness of the team's game and its individual players, selected by them to perform game tasks of heroes and objects, as well as a retrospective analysis of this information, which may be useful to team sponsors and investors in esports, as it contributes to the development of new game strategies. In addition, the data provided by the system can be used in scientific research related to the study of various aspects of the gameplay, such as psychological and pedagogical, neuro-linguistic, economic, etc.

Keywords: *Software and information system, computer game, esports discipline, software interface, game statistics, Dota 2, Stratz*

Аннотация. В статье рассматривается создание прототипа программно-информационной системы для сбора и анализа статистики профессиональных матчей по киберспортивной дисциплине Dota 2, который является полнофункциональной системой по многоуровневому и детализированному поиску структурированной информации, полученной с помощью Stratz API.

Представленный прототип обеспечивает возможность получения данных об эффективности игры команды и её отдельных игроков, выбранных ими для выполнения игровых задач героев и предметов, а также ретроспективного анализа указанной информации, что может быть полезно спонсорам команд и инвесторам в киберспорт, поскольку способствует разработке новых игровых стратегий. Кроме того, предоставляемые системой данные могут быть использованы в научных исследованиях, связанных с изучением различных аспектов игрового процесса, таких как психолого-педагогические, нейролингвистические, экономические и т.п.

Ключевые слова: Программно-информационная система, компьютерная игра, киберспортивная дисциплина, программный интерфейс, игровая статистика, Dota 2, Stratz

Рецензент: Торопцев Василий Владимирович - кандидат технических наук, доцент.
ФГБОУ ВО «РГАУ-МСХА им. К.А. Тимирязева»

Введение

С каждым годом компьютерные игры приобретают все большую популярность по всему миру. В частности, исключительной популярностью среди подростковой и юношеской аудитории пользуется компьютерная игра «Dota 2» (англ. «Defence of the Ancients 2»), что связано со следующими обстоятельствами:

- присутствием в игре большого числа игровых персонажей с разнообразными игровыми качествами;
- наличием значительных призовых фондов в разыгрываемых киберспортивных турнирах по Dota 2 The International;
- возможностью получения доступа к истории игровых матчей с целью анализа особенностей поведения игроков на протяжении длительного периода времени на различных профильных сайтах.

В связи с этим многие современные исследования посвящены исследованию разных аспектов, связанных с игровым процессом – психолого-педагогические, нейролингвистические, экономические и др. Например, в статье [1] проведено исследование взаимосвязи личностных особенностей игроков подросткового и юношеского возраста с их поведением в групповой компьютерной игре Dota 2), в [2] рассматривались социально-педагогические условия повышения коммуникативного взаимодействия детей и подростков с помощью указанной компьютерной игры. Комплексный анализ сленга участников игры приведен в [3], лексики киберспортивного комментатора и геймера Dota 2 – в [4]; экономические показатели и перспективы развития киберспортивных соревнований были рассмотрены в [5], [6]. Также большое количество работ посвящено прогнозированию результатов игры [8,10] и оптимальному подбору персонажей [9].

Большинство проведенных исследований базировались на анализе данных о каждом матче, проведенном пользователем, которые находятся в свободном доступе на официальном сервере игры Dota 2. Однако подобные исследования акцентировали внимание на индивидуальных особенностях игроков и их результатах, не уделяя должного внимания командной игре, в частности, игре профессиональных киберспортивных команд.

В настоящее время большой популярностью пользуются различные веб-сайты и приложения, которые предоставляют более подробные данные, чем официальный сайт Dota 2. Рассмотрим некоторые из них.

Онлайн-платформа Dotabuff, специализирующаяся на анализе и отслеживании статистики игроков в Dota 2, даёт пользователю возможность просматривать статистику по своим играм, в том числе количество побед, поражений, средний уровень, продолжительность матчей и другие показатели эффективности в игре. Кроме того, у Dotabuff есть дополнительные функции, позволяющие проводить более подробный анализ игры, например, предоставлена возможность просмотра статистики по каждому выбранному игроком герою, оценки прогресса при их использовании, сравнения своих показателей эффективности игры с показателями других игроков (как противников, так и союзников).

Платформа OpenDota предоставляет возможность проведения более глубокого анализа игровых данных для Dota 2. Она собирает информацию из официального API игры и обеспечивает доступ к ней пользователям с целью её дальнейшего исследования и анализа. С помощью OpenDota можно получить информацию о различных показателях и статистике игроков, команд, героев и матчей. Важной особенностью OpenDota является разрешение доступа к API разработчикам, чтобы они могли использовать данные в своих проектах и создавать собственные приложения для анализа игровых данных.

Главная особенность платформы Stratz — это возможность увидеть детальную статистику каждой игры с подробной информацией о количестве убийств, смертей, помощи, а также другие ключевых игровых показателях, таких как, золото за минуту, опыт за минуту и т. д. Кроме того, в Stratz при просмотре игровых данных команд и игроков, можно узнать лучшего игрока по определенному герою, его победные и проигрышные рекорды, а также его статистику в разных режимах игры – казуальном, рейтинговом и соревновательном.

В отличие от общедоступных API Dota (Valve, OpenDota), которые предоставляют разработчикам данные в формате REST, платформа Stratz – единственный API Dota, который также доступен в формате GraphQL. Данный формат предлагает ряд

преимуществ, включая возможность сбора широкого спектра различной информации в одном запросе. Кроме того, Stratz предлагает расширенную поминутную статистику по многим ключевым моментам матча.

В связи с вышеперечисленным представляется актуальной разработка программно-информационной системы для сбора и анализа статистики профессиональных матчей по киберспортивной дисциплине Dota 2, позволяющей обрабатывать данные, предоставляемые разработчикам посредством общедоступного API портала Stratz.

Цель исследования

Целью исследования является разработка прототипа программно-информационной системы для сбора и анализа статистики профессиональных матчей по киберспортивной дисциплине Dota 2, позволяющей получать исчерпывающую информацию по каждому профессиональному матчу, а также по каждому игроку и команде в целом. Скомпилированная информация дает возможность применять её при проведении научных исследований в различных предметных областях. С другой стороны, на основе собранной статистики представляется возможным более точно определять уровень подготовки команды и оценивать её соревновательный потенциал, выявлять слабые места в её подготовке, находить оптимальный состав игроков с целью повышения успешности выступлений команды.

Материалы и методы исследования

Данное исследование базируется на материалах, предоставляемых открытым API портала Stratz. Методами исследования, необходимыми для проектирования и реализации прототипа программно-информационной системы (ППИС), являются:

- UML-диаграммы, с помощью которых описана архитектура ППИС;
- технология ASP.NET Core MVC;
- язык программирования C#;
- СУБД Microsoft SQL Server.

Результаты исследования и их обсуждение

Разработанный авторами прототип программно-информационной системы (ППИС) для сбора и анализа статистики профессиональных матчей по киберспортивной дисциплине Dota 2 является полнофункциональной системой по многоуровневому поиску структурированной информации, предоставляемой Stratz API – порталом, целью которого является сбор статистических данных всех матчей в Dota 2.

Предполагается, что данная разработка будет применяться в исследовательских целях для получения необходимых данных с целью последующего проведения разнопланового статистического анализа в различных предметных областях. Также

созданный ППИС может быть полезным для инвесторов в киберспорт информационным ресурсом, который позволяет отследить последние результаты команды, её место в общем рейтинге и статистику очных встреч, получить больше информации о предстоящем турнире, например, о размере призового фонда, правилах, списке участников, турнирной таблице или сетке.

Описание архитектуры ППИС проводилось с помощью UML-диаграмм. В качестве примера рассмотрим некоторые из них.

На диаграмме вариантов использования, представленной на рисунке 1, показано, что в ППИС существуют 2 профиля пользователей: «Администратор», «Посетитель сайта», каждый из которых имеет доступ к определенному набору функций:

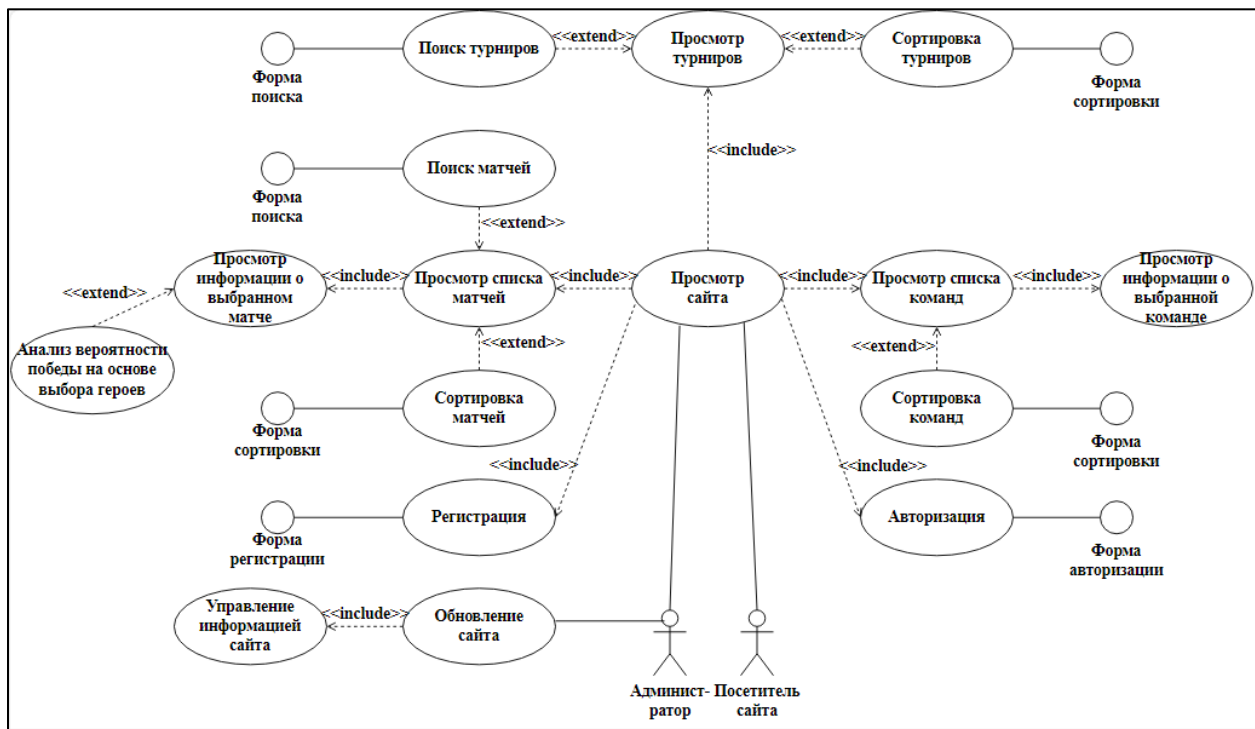


Рисунок 1. Диаграмма вариантов использования

Варианты использования:

Просмотр сайта – это отображение главной страницы сайта, на которой реализованы возможности просмотра списков матчей, турниров, профессиональных команд, а также просмотр информации об отдельном матче, статистики отдельной профессиональной команды и т. д.

Просмотр информации об отдельном матче – одна из основных функциональных возможностей сайта, которая позволяет получить детальную статистику отдельного

матча, включающую, в том числе, время матча и общий счёт, состояние карты, разницу во внутриматчевой экономике команд и их опыте, выбранных игроками персонажей, предметов, которые игроки приобретают во время матча.

Анализ вероятности победы команды на основе данных о выборе персонажей – функционал, который даёт возможность оценить вероятность победы каждой из команд, основываясь на данных о выбранных персонажах каждой из команд.

Просмотр статистики команды – это отображение информации о количестве побед и поражений отдельной профессиональной команды, сыгранных ею матчах, текущем составе и рейтинге относительно других команд.

При разбиении программной системы на структурные компоненты можно выделить четыре их типа:

- 1) Файлы, содержащие представления страниц в формате cshtml, например, Home.cshtml – файл для отображения главной страницы сайта;
- 2) Файлы-контроллеры формата cs, отвечающие за представления соответствующих страниц формата cshtml, например, HomeController.cs – контроллер, отвечающий за представление страницы Home.cshtml;
- 3) База данных – упорядоченный набор структурированной информации по персонажам, матчам, турнирам, командам, игрокам и пользователям.
- 4) Stratz API – открытое API по Dota 2, предоставляющее полную информацию обо всех турнирах, матчах, персонажах и игроках.

Для организации и управления базой данных выбрана СУБД Microsoft SQL Server. Создание базы данных и таблиц происходит с помощью технологии Entity Framework Core (MS SQL Server). Для этого в языке C# сначала создаются классы (сущности), которые впоследствии преобразуются в таблицы базы данных. Поля классов преобразуются в столбцы базы данных. Набор классов и пример класса для последующего преобразования представлен на рисунке 2:



Рисунок 2. Набор классов и пример класса для преобразования в таблицы базы данных

Далее для создания таблиц в классе Context.cs организируются поля DbSet<T>, имена которых соответствуют названиям таблиц:



Рисунок 3. Создание полей DbSet<T> в классе Context.cs

Далее для установления первичных ключей в классе Context.cs, отвечающем за управление базой данных, необходимо переопределить метод OnModelCreating(). При этом явно указывают ключи для каждого класса с помощью метода HasKey():

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Hero>().HasKey(h => h.HeroId);
    modelBuilder.Entity<GamePickedHero>().HasKey(gph => new { gph.HeroId, gph.GameId });
    modelBuilder.Entity<Game>().HasKey(g => g.GameId);
    modelBuilder.Entity<League>().HasKey(l => l.LeagueId);
    modelBuilder.Entity<LeagueGame>().HasKey(lg => new { lg.LeagueId, lg.GameId});
    modelBuilder.Entity<LeagueTeam>().HasKey(lt => new { lt.LeagueId, lt.TeamId});
    modelBuilder.Entity<Player>().HasKey(p => p.PlayerId);
    modelBuilder.Entity<PlayerTeam>().HasKey(pt => new { pt.PlayerId, pt.TeamId });
    modelBuilder.Entity<Team>().HasKey(t => t.TeamId);
    modelBuilder.Entity<User>().HasKey(u => u.Id);
}
```

Рисунок 4. Установление первичных ключей классов

При вызове метода Database.EnsureCreated() класса Context.cs будет создана база данных и таблицы, входящие в нее:

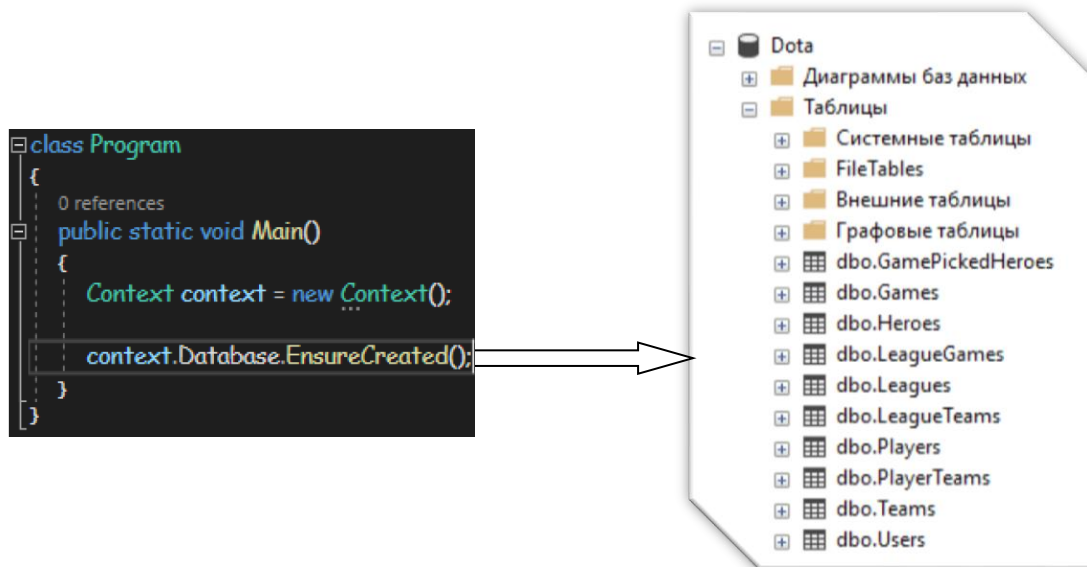


Рисунок 5. Функция Database.EnsureCreated() и результат её выполнения

Таблицы базы данных заполняются посредством выполнения запросов в C# к Open API и дальнейшего заполнения через Entity Framework Core. Для этого определяется класс DotaOpenAPIProvider.cs, который будет получать информацию из Dota Open API, реализация класса показана на рисунке 6.


```
internal class DotaOpenApiProvider
{
    private static string apiMatchInfoUrl = "https://api.opendota.com/api/matches/";
    private static string apiHeroesUrl = "https://api.opendota.com/api/constants/heroes";
    private static string apiLeaguesUrl = "https://api.opendota.com/api/leagues";
    private static string apiTeamsUrl = "https://api.opendota.com/api/teams";
    private static string apiPlayersByTeamIdUrl = "https://api.opendota.com/api/teams/{0}/players";
    private static string apiTeamsByLeagueIdUrl = "https://api.opendota.com/api/leagues/{0}/matches";
    private static string apiLeagueMatchesByLeagueIdUrl = "https://api.opendota.com/api/leagues/{0}/matches";

    2 references
    public static League[] GetLeagues()
    {
        return Query<League[]>(apiLeaguesUrl);
    }

    2 references
    public static Team[] GetTeams()
    {
        return Query<Team[]>(apiTeamsUrl);
    }
}
```

Рисунок 6. Реализация класса DotaOpenAPIProvider.cs

Далее в классе Context.cs необходимо создать методы заполнения базы данных. На рисунке 7 представлена реализация данных методов на примере заполнения таблицы Leagues.

```
public void FillLeagues(bool deleteDataAtStart)
{
    if (deleteDataAtStart)
    {
        Leagues.ExecuteDelete();
    }

    League[] leagues = DotaOpenApiProvider.GetLeagues();

    Leagues.AddRange(leagues);

    SaveChanges();

    Console.WriteLine("Leagues filled");
}
```

Рисунок 7. Реализация методов заполнения базы данных

После вызова методов все таблицы базы данных будут заполнены данными, поступающими из Open API. На рисунке 8 показан вызов методов из функции Main() и фрагмент заполненной таблицы Leagues.



The image shows a code editor on the left with the following C# code:

```
class Program
{
    public static void Main()
    {
        Context context = new Context();
        context.FillHeroes(true);
        context.FillLeagues(true);
        context.FillUsers(true);
        context.FillTeams(true);
        context.FillPlayersAndPlayerTeams(0, 20, false);
    }
}
```

On the right, there is a table with the following data:

LeagueId	Name	Tier	BannerUrl
0	JetsetPro Amateur League 1x1 Season 1	amateur	NULL
2	Wild Cards West	premium	econ/leagues/subscription_logo_the_wildcards
3	Wild Cards East	premium	econ/leagues/subscription_logo_the_wildcards
4	The Defense	professional	econ/leagues/the_defense
5	It's Gosu Monthly Madness Asia	NULL	econ/leagues/subscriptions_mmasia_ingame
6	Star Series Season II Lan Final	professional	econ/leagues/subscriptions_starladder2_ingame
7	BeyondTheSummit World Tour	professional	econ/leagues/subscriptions_beyondthesummit_ingame
8	Prodota 2 Worldwide League	professional	econ/leagues/subscriptions_prodota2_ingame
9	Machinima ECAL: Americas	NULL	econ/leagues/subscriptions_americas_ingame
10	Razer Mini Madness	NULL	econ/leagues/subscriptions_razeminimadness_ingame

Рисунок 8. Вызов методов заполнения таблиц и результат для таблицы Leagues

Основной функционал программно-информационной системы состоит в получении и выводе по запросу пользователя структурированной информации об игроках, командах, матчах, турнирах и т.д., хранящейся в базе данных.

Рассмотрим в качестве примера вывод списка команд, реализация которого основана на создании двух сущностей – контроллера TeamsController.cs и представления списка команд Teams.cshtml. Контроллер TeamsController.cs, в свою очередь, состоит из одного метода Index, который в качестве входных параметров принимает параметры пагинации и поиска.

```
public class TeamsController : Controller
{
    private Context _context;
    public TeamsController(Context context)
    { _context = context; }
    [HttpGet]
    public IActionResult Index
    (
        int limit = 10,
        int offset = 0,
        string? partOfName = null)
    {
        var teams = partOfName is null? _context.Teams
        : _context.Teams.Where(t => t.Name.Contains(partOfName));
        var orderedTeams = teams
            .OrderByDescending(x => x.Rating)
            .Skip(offset)
            .Take(limit)
            .ToArray();
        ViewBag.teams = teams;
        return View();
    }
}
```

Рисунок 9. Контроллер TeamsController.cs

Данный метод возвращает полный список команд, или, если пользователь указал в строке поиска часть названия команды (переменная **partOfName**), только те команды, который содержат в себе данную подстроку. Список команд сортируется по рейтингу и обрабатывается согласно параметрам пагинации (**limit** и **offset**).

Для представления списка команд создаётся html-таблица, которая в цикле заполняется значениями, переданными контроллером TeamsController:

```
<div class="table-responsive">
  <table class="table table-hover table-bordered border border-2">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Logo</th>
        <th scope="col">Name</th>
        <th scope="col">Rating</th>
      </tr>
    </thead>
    <tbody>
      @foreach (var team in teams)
      {
        <tr class="">
          <td scope="row">
            <p>#@GetIndex()</p>
          </td>
          <td>
            <img src=@team.Key.LogoUrl class="teamLogo"
style="height:40px;">
          </td>
          <td>@team.Name</td>
          <td>@team.Rating</td>
        </tr>
      }
    </tbody>
  </table>
</div>
```

Рисунок 10. Представление списка команд Teams.cshtml

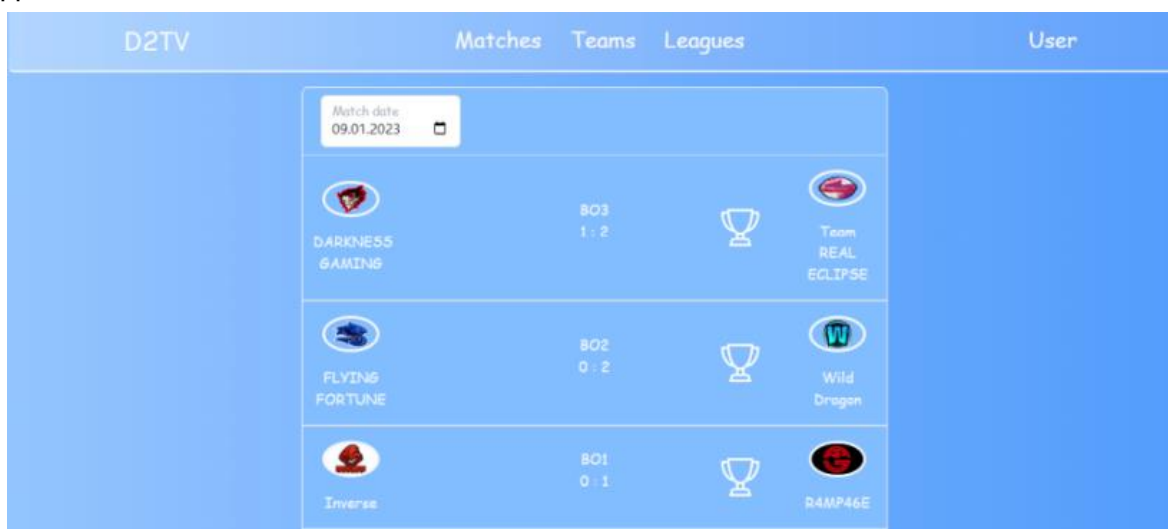
Результат вызова метода Index контроллера представлен на рисунке 11:



#	Logo	Name	Rating
#1		Team Tundra Exports	1624.91
#2		Team Liquid	1532.73
#3		Team Secret	1530.73
#4		P50.L6D	1523.04
#5		CyberBench-1	1520.12
#6		Team Aster	1487.27
#7		Team Spirit	1459.07
#8		OG	1439.16
#9		Gaimin Gladiators	1433.01
#10		Shopify Rebellion	1413.61

Рисунок 11. Результат вызова метода Index

Аналогично создаются контроллеры и представления для отображения других списков. Например, на главной странице ППИС представлен список матчей, сыгранных на заданную дату. Данный список содержит информацию о командах, участвовавших в матче, о количестве игр в матче и счете по данным играм, а также информацию о победителе матча.



Match date	Match	Score	Winner
09.01.2023	DARKNESS GAMING vs TEAM REAL ECLIPSE	BO3 1 : 2	TEAM REAL ECLIPSE
	FLYING FORTUNE vs Wild Dragon	BO2 0 : 2	Wild Dragon
	Inverse vs RAMP46E	BO1 0 : 1	RAMP46E

Рисунок 12. Главная страница ППИС

При выборе конкретного матча выводится более детальная информация, например, команды, участвовавшие в матче, количество сыгранных карт, игровое время, счет каждой команды, суммарное количество заработанного за всю игру золота у каждой из команд, опыт и вероятности победы команд за период игрового времени, а также информация о каждом игроке-участнике матча: его имя, выбранные герои и внутриигровые предметы, уровни персонажей, вклад в общеконандное золото.

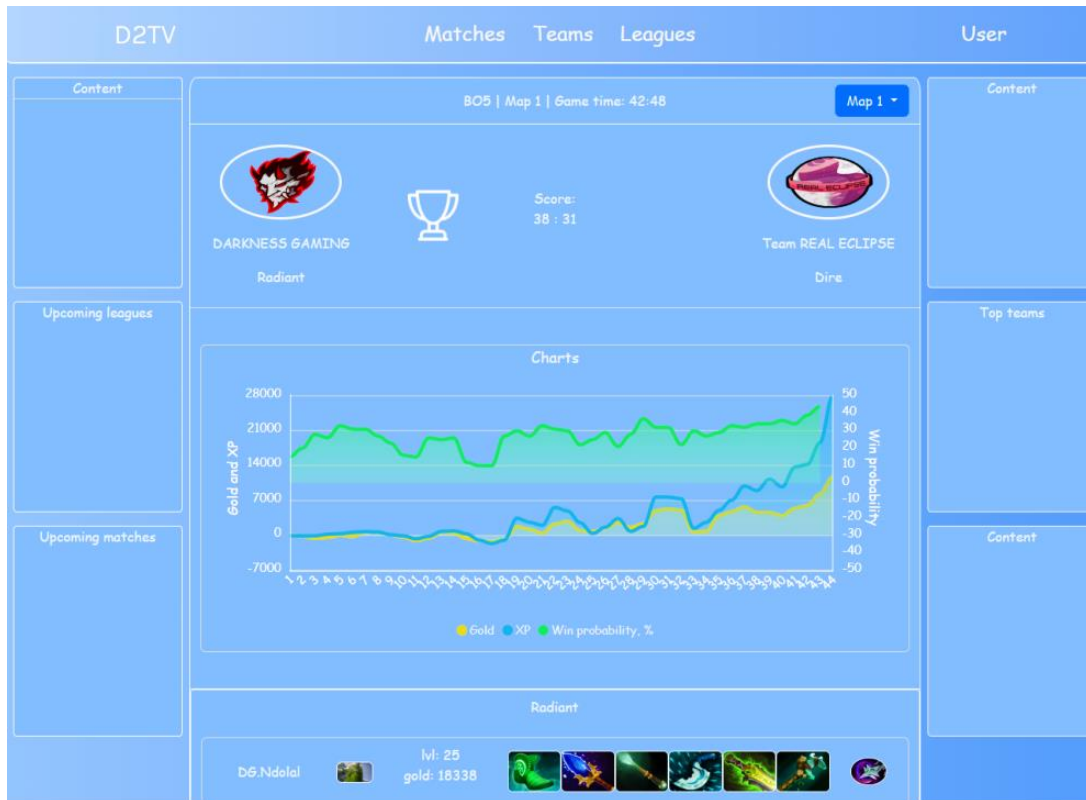


Рисунок 13. Меню матча

В верхней секции окна можно увидеть серию, входящую в данную игру, текущую карту, игровое время:

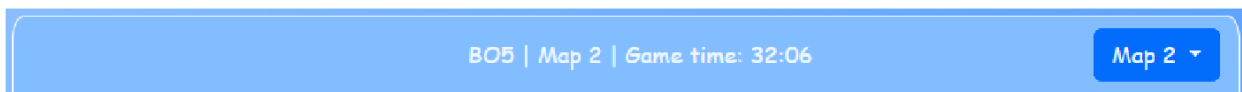


Рисунок 14. Первая секция меню матча

Ниже в секции 2 представлены логотипы и названия команд, счёт команды в данной серии, а также значок победителя карты:

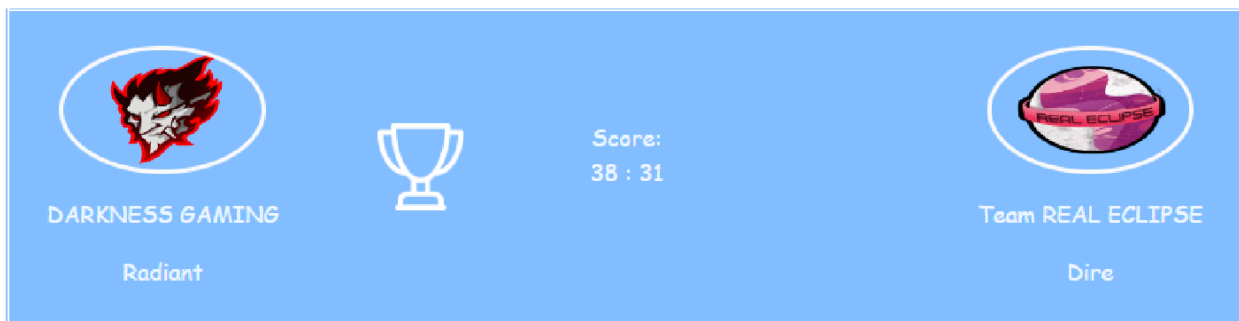


Рисунок 15. Вторая секция меню матча

В третьей секции представлен график, на котором отображено превосходство (положительное или отрицательное) первой команды – сил света (Radiant) над второй – силами тьмы (Dire) заработанному золоту (Gold, желтая линия), заработанному опыту (XP, синяя линия), а также вероятность её победы (Win probability, зеленая). Значение вероятности победы является результатом анализа таких факторов, как вероятность победы героев сил света против сил тьмы, их превосходства по золоту, опыту, купленным и нейтральным предметам.

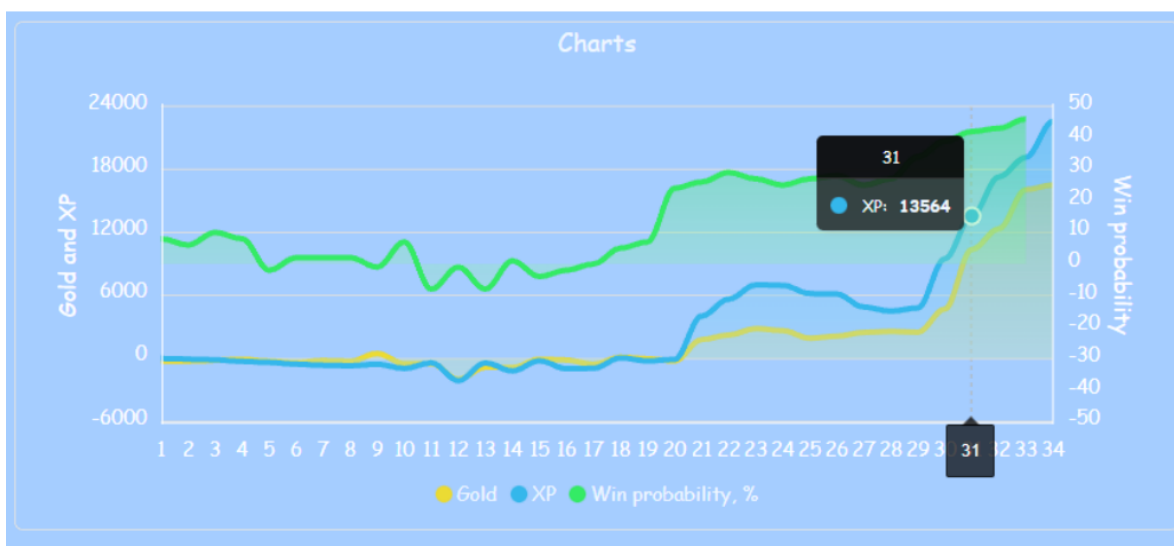


Рисунок 16. Третья секция меню матча

В четвертой секции приведены игроки команды сил света. Здесь мы можем увидеть имена игроков, выбранных ими героев, достигнутые уровни, добытое за время матча золото, купленные предметы и один нейтральный предмет.



Рисунок 17. Четвертая секция меню матча

Аналогично мы можем просмотреть информацию по команде сил тьмы.

Так как Stratz предоставляет доступ к расширенной поминутной статистике по многим ключевым моментам матча, то в базе данных происходит накопление получаемой информации, которую затем можно анализировать подходящими методами математической статистики, исходя из целей исследования.

Заключение

Представленный в статье прототип программно-информационной системы представляет собой инструментарий для получения данных, являющихся предметом статистического анализа профессиональных матчей по киберспортивной дисциплине Dota 2. В частности, с его помощью можно получать сведения об эффективности игры команды и её отдельных игроков, частоте использования различных персонажей, динамике результативности команды, использовании участниками команды нестандартных приемов, связанных, например, с выбором героев и предметов, что является прямым признаком креативности команды. Разработанный прототип является полнофункциональной системой по многоуровневому и детализированному поиску структурированной информации, полученной с помощью Stratz API.

Созданный ППИС может быть полезным для спонсоров команд, так как позволяет оценивать перспективы команды в последующих матчах на основе ретроспективного

анализа, для инвесторов в киберспорт с целью определения наиболее интересных игровых стратегий, а также для болельщиков, желающих, получить больше информации о прошедших и предстоящих турнирах.

Предполагается, что данная разработка также будет применяться и в исследовательских целях как способ получения необходимых данных для последующего проведения разнопланового статистического анализа в различных предметных областях.

References

1. Рубцова О.В., Панфилова А.С., Артеменков С.Л. Исследование взаимосвязи личностных особенностей игроков подросткового и юношеского возраста с их поведением в виртуальном пространстве (на примере групповой компьютерной игры «Dota 2») <https://cyberpsy.ru/articles/lichnostnye-osobennosti-podrostkov-na-primere-dota-2/>
2. Руденков, М. Б. Социально-педагогические условия повышения коммуникативного взаимодействия детей и подростков с помощью компьютерной игры "Dota 2" / М. Б. Руденков // Главные аспекты развития современной науки. Инновационная парадигма развития мировой экономики, Санкт-Петербург, 06–15 февраля 2019 года / Институт управления и социально-экономического развития, Саратовский государственный технический университет. – Санкт-Петербург: Общество с ограниченной ответственностью "Центр профессионального менеджмента "Академия Бизнеса", 2019. – С. 55-57.
3. Костина, Е. В. Комплексный анализ сленга участников компьютерной игры Dota 2 / Е. В. Костина, В. А. Мороховцев // Lingua Academica: Актуальные проблемы лингвистики и лингводидактики: Материалы V Всероссийской научно-практической конференции, Ульяновск, 05–09 февраля 2020 года / Под ред. Н.А. Крашенинниковой. – Ульяновск: Ульяновский государственный университете, 2020. – С. 217-223.
4. Микулинский, А. Д. Сравнительный анализ лексики киберспортивного комментатора и геймера Dota 2 / А. Д. Микулинский // Теоретические и прикладные аспекты изучения речевой деятельности. – 2021. – Т. 14, № 7. – С. 89-98.
5. Ратников, К. С. Экономические показатели и перспективы развития киберспортивных соревнований на примере компьютерной игры Dota 2 / К. С. Ратников // Сборник статей преподавателей, обучающихся на кафедре студентов и магистрантов, а также студентов и магистрантов РГУФКСМиТ и работников сферы образования из Болгарии по результатам проведенной 31 января 2019 года Конференции посвященной теме "Экономика - спорту" и 10-летию кафедры Экономики, Москва, 31 января 2019

года. – Москва: Общество с ограниченной ответственностью "КОНВЕРТ", 2019. – С. 94-99.

6. Тарасенко, В. А. Финансово-экономические аспекты компьютерного спорта / В. А. Тарасенко // Имущественные отношения в Российской Федерации. – 2019. – № 11(218). – С. 37-49.

7. Кузнецов, Д. В. Командное взаимодействие в игре Dota 2 / Д. В. Кузнецов, И. Н. Третьяк, М. Н. Власов // Научный взгляд в будущее. – 2020. – Т. 2, № 19. – С. 114-118. – DOI 10.30888/2415-7538.2020-19-02-026.

8. Молдавский, М. И. Подбор оптимальной модели прогнозирования результатов матчей Dota 2 / М. И. Молдавский, Ю. А. Степанов // Фундаментальные и прикладные исследования в физике, математике и информатике: Материалы симпозиума в рамках XVII (XLIX) Международной научной конференции студентов, аспирантов и молодых ученых, Кемерово, 21 апреля 2022 года / Сост. Ю.А. Степанов, С.Ю. Завозкин, В.В. Илькевич. Том Выпуск 23. – Кемерово: Кемеровский государственный университет, 2022. – С. 247-249.

9. МЭС по подбору команды персонажей в многопользовательской игре Dota 2 / Н. С. Муравьев, О. И. Панков, Е. А. Захаренко [и др.] // Мивар'23: Сборник студенческих статей. – Москва: Издательский Дом "Инфра-М", 2023. – С. 138-147.

10. Плаксин, В. А. Прогнозирование победителя в матче DOTA 2 с помощью нейронных сетей / В. А. Плаксин, Н. В. Быстров, В. А. Пятунина // Концепции устойчивого развития науки в современных условиях: сборник статей Международной научно-практической конференции: в 2 частях, Екатеринбург, 28 июня 2017 года. Том часть 2. – Екатеринбург: Общество с ограниченной ответственностью "ОМЕГА САЙНС", 2017. – С. 55-57.