

Uniwersytet Kardynała Stefana Wyszyńskiego  
Wydział Matematyczno-Przyrodniczy



*Symulacja Mistrzostw Świata w Piłkę Koszykową*

*Damian Ubowski*

Warszawa, 2020

## Spis treści

1	Wstęp .....	3
2	Opis gry .....	4
2.1	Opis przebiegu mistrzostw .....	4
2.2	Opis algorytmów .....	4
2.2.1	Algorytm wyboru drużyn .....	4
2.2.2	Algorytm tworzenia grup po fazie eliminacji .....	5
2.2.3	Algorytm generowania akcji podczas meczu .....	6
2.2.4	Algorytm zliczania punktów w podsumowaniu grupy .....	7
2.2.5	Algorytm tworzenia grup do drugiej fazy grupowej .....	7
2.2.6	Algorytmy tworzenia par w fazie pucharowej .....	7
3	Opis programu .....	9
3.1	Opis struktury programu .....	9
3.2	Schemat blokowy aplikacji .....	17
3.3	Schematy blokowe algorytmów .....	18
3.3.1	Algorytm generowania akcji podczas meczu .....	18
3.4	Kod źródłowy wybranych elementów programu .....	22
3.4.1	Kod źródłowy losowania akcji podczas meczu .....	22
3.4.2	Kod źródłowy tworzenia grup dla Pierwszej Fazy Grupowej .....	24
3.5	Interfejs aplikacji .....	25
3.5.1	Wybór drużyn do mistrzostw .....	26
3.5.2	Losowanie .....	28
3.5.3	Pierwsza Faza Grupowa .....	33
3.5.4	Druga Faza Grupowa .....	36
3.5.5	Faza Pucharowa .....	37
4	Instrukcja obsługi .....	41
4.1	Uruchomienie aplikacji .....	41
4.2	Kompilacja aplikacji .....	46
4.3	Przygotowanie środowiska .....	47
5	Bibliografia .....	51
6	Zawartość płytki CD .....	52



## 1 Wstęp

Projekt dotyczący utworzenia aplikacji, która umożliwi użytkownikowi przeprowadzenie symulacji przebiegu mistrzostw świata w piłce koszykowej w ramach federacji FIBA (fr. Fédération Internationale de Basketball).

Mistrzostwa składają się z eliminacji, pierwszej fazy grupowej, drugiej fazy grupowej i fazy finałowej. Najlepsze drużyny z poprzedniej fazy przechodzą do następnej [1] [2].

Piłka koszykowa jest szybką, dynamiczną grą kontaktową, w której do zdobycia punktów lub wymian piłki dochodzi często. Każdy z meczy trwa zazwyczaj 40 minut, a podczas niego może dojść do zdobycia punktów przez zawodnika jednej z drużyn lub do utraty piłki, oraz do popełnienia faulu [3].

Program został napisany przy użyciu dwóch technologii. Warstwa prezentacji została wykonana w języku JavaScript [4] w oparciu o framework Angular [5]. Natomiast cała logika aplikacji utworzona została w języku C# [13] z użyciem framework-u ASP.NET Core [6] oraz z bazą danych MSSQL.

Aplikacja została wykonana z myślą o systemie operacyjnym Microsoft Windows 10, na którym jest zainstalowana platforma Microsoft .NET w wersji Core 3.1.

## 2 Opis gry

### 2.1 Opis przebiegu mistrzostw

Do mistrzostw przystępuje łącznie 32 drużyn z czterech konfederacji FIBA, wybranych przez użytkownika w sposób ręczny lub losowy. Każda z drużyn posiada przypisaną do siebie wartość Klasy, która ustala procentową szansę na wygraną meczu. Klasy mają wartości od 1 do 4. Klasa 1 jest przypisywana do najlepszych drużyn, a Klasa 4 do najgorszych.

Podczas fazy eliminacji są wybierane drużyny mające wziąć udział w mistrzostwach. Drużyny są porządkowane pod względem Klasy od najlepszej (Klasa 1) do najgorszej (Klasa 4) i wkładane do 8 koszyków. W koszyku pierwszym znajdują się drużyny najlepsze i w kolejnych drużyny coraz gorsze. Oznacza to, że w ostatnim koszyku znajdują się drużyny najgorsze pod względem Klasy.

Po rozlosowaniu drużyn do koszyków następuje pierwsza faza grupowa, w której drużyny z koszyków są przekładane do grup A-H, gdzie będą toczony rozgrywki. W celu utworzenia grup A-H drużyny z koszyków pierwszego, czwartego, piątego i ósmego rozlosowane zostają do grup A, C, E i G. Natomiast reprezentacje z koszyków drugiego, trzeciego, szóstego i siódmego do grup B, D, F i H. Następnie rozgrywane są mecze na zasadzie każdy z każdym w ramach grup. Dwie najlepsze drużyny z każdej grupy awansują do drugiej fazy grupowej, a dwie najgorsze przystępują do rywalizacji o miejsca 17-32.

W drugiej fazie grupowej 16 zwycięskich drużyn są układane w grupy I-L w których będą rywalizowały o wstęp do fazy pucharowej. Do fazy pucharowej dostaną się dwa najlepsze zespoły z każdej grupy I-L. Drużyny zajmujące miejsca trzy i cztery w grupach I-L zakończą mistrzostwa na miejscach odpowiednio 9-12 i 13-16.

Po rozegraniu meczy w ramach drugiej fazy grupowej następuje faza pucharowa, która jest podzielona na ćwierćfinały, półfinały i finały. Do tej fazy podchodzi osiem najlepszych drużyn z Drugiej Fazy Grupowej. Drużyny, które przegrały w ćwierćfinałach będą walczyć o miejsca 5-8. Drużyny, które przegrały w półfinałach rozegrają grę o miejsca 3-4, a te które wygrały o miejsca pierwsze i drugie. Jeśli po którymś meczu fazy pucharowej trafi się sytuacja, w której obie drużyny rozgrywające mecz mają tyle samo punktów następuje dogrywka. Dogrywka rozszerza możliwy czas gry o 5 minut, pozwalając na zagranie dodatkowych minut meczu. W koszykówce wynik meczu musi zostać rozstrzygnięty. Dogrywek będzie odbywało się tak wiele, aż wynik na zakończenie dogrywki będzie rozstrzygnięty (jedna drużyna będzie miała więcej punktów od drugiej) [11].

Rozgrywka o miejsca 17–32 również toczyła się w czterech grupach (M–P) po cztery zespoły. Reprezentacje, które zajęły pierwsze miejsce w grupie zostały sklasyfikowane na miejscach 17–20, miejsca 21–24 zajęły drużyny z drugich miejsc w grupach, miejsca 25–28 ekipy z trzecich a 29–32 z czwartych miejsc.

### 2.2 Opis algorytmów

#### 2.2.1 Algorytm wyboru drużyn

Wybór drużyn, które wezmą udział w mistrzostwach odbywa się z pomocą użytkownika. Użytkownik ma możliwość ręcznego lub losowego wybrania 32 spośród ponad 100 drużyn w ramach czterech konfederacji FIBA. Wybór jest przeprowadzany z podziałem na konfederacje i użytkownik musi wybrać osiem drużyn w każdej z nich. Dodatkowo każda z konfederacji musi posiadać dokładnie dwie drużyny o Klasie 1 co łącznie będzie skutkowało posiadaniem ośmiu drużyn z Klasą 1 w mistrzostwach. Wybór drużyn w ramach konfederacji odbywa się po kolei. Oznacza to, że użytkownik

na początku wybiera drużyny z jednej konfederacji, a następnie z kolejnej i tak aż nie wybierze drużyn z każdej z czterech konfederacji.

Oprócz ręcznego wyboru drużyn użytkownik jest w stanie wybrać je w sposób losowy z zachowaniem zasad mówiących o ilości drużyn w konfederacji. Jeśli użytkownik zdecyduje się wybrać opcję losową to aplikacja w ramach danej konfederacji wybierze losowo dwie drużyny o Klasie 1, a następnie wybierze sześć drużyn o losowych klasach.

#### *2.2.1.1 Algorytm krokowy losowego wyboru drużyn w konfederacji*

Na wejściu algorytm przyjmuje tablicę ze wszystkimi drużynami w ramach jednej konfederacji, a na wyjściu zwraca losowo wybrane drużyny. Podczas wyboru została zachowana zasada wyboru dwóch drużyn z Klasą 1.

1. Do tablicy A przypisz drużyny z Klasą 1
2. Do tablicy B przypisz drużyny, które mają Klasę różną od 1
3. Utwórz ośmioelementową tablicę R oraz zmienną  $i = 0$
4. Dla każdego elementu w tablicy R:
  - 4.1. Jeżeli  $i = 0$  lub  $i = 1$  to:
    - 4.1.1. Wylosuj drużynę z tablicy A i włóż do R(i)
  - 4.2. W przeciwnym wypadku:
    - 4.2.1. Wylosuj drużynę z tablicy B i włóż do R(i)
  - 4.3.  $i = i + 1$
  - 4.4. Jeśli  $i > 8$  to zakończ pętlę
5. Zwróć tablicę R

#### *2.2.2 Algorytm tworzenia grup po fazie eliminacji*

Do mistrzostw dostały się 32 drużyny. Docelowo każda z tych drużyn musi się znaleźć w jednej z 8 grup A-H. Podział na grupy odbywa się w dwóch częściach. Pierwsza to podział na koszyki, a druga podział na faktyczne grupy A-H.

##### *2.2.2.1 Algorytm krokowy tworzenia grup po fazie eliminacji*

Na wejściu algorytm przyjmuje tablicę o wielkości 32. W każdym elemencie tablicy znajduje się jedna drużyna wybrana przez użytkownika. Wynikiem algorytmu są grupy A-H wypełnione drużynami wybranymi podczas eliminacji.

1. Posortuj tablicę drużyn  $D$  rosnąco według ich siły
2. Utwórz 8 koszyków  $K$  oraz zmienną  $i=0$
3. Dla każdego koszyka  $K(i)$ :
  - 3.1. Weź elementy od  $i*4$  do  $i*4+4$  i włóż do koszyka  $K(i)$
  - 3.2.  $i = i + 1$
  - 3.3. Jeśli  $i \geq 7$  to zakończ pętlę.
4. Utwórz osiem grup  $G$  i zmienną  $j=0$ ,  $l=0$
5. Dla każdej grupy  $G(l)$ :
  - 5.1.  $j=0$
  - 5.2. Dla każdego koszyka  $K(j)$ :
    - 5.2.1. Wybierz losowo drużynę  $X$  z koszyka
    - 5.2.2. Włóż drużynę  $X$  do aktualnej grupy  $G(l)$
    - 5.2.3. Usuń drużynę  $X$  z koszyka  $K(j)$
    - 5.2.4.  $j = j + 1$
    - 5.2.5. Jeśli  $j \geq 7$  zakończ pętlę z kroku 5.2

- 5.3.  $I = I + 1$
- 5.4. Jeśli  $I \geq 7$  zakończ pętlę z kroku 5
6. Zwróć grupy G

### 2.2.3 Algorytm generowania akcji podczas meczu

Podczas symulowania przebiegu meczu niezbędny jest algorytm, który będzie losowo ustalał to co działo się podczas meczu. Algorytm powinien dla każdego z zespołów na zmianę generować akcje takie jak: zdobycie punktu, utrata piłki lub faul. Na wejściu algorytm przyjmuje Klasę drużyny i zwraca losową akcję.

#### 2.2.3.1 Algorytm krokowy generowania akcji podczas meczu

Wartość  $P_a$  informuje nas o szansie na wystąpienie akcji ataku co może przełożyć się na zdobycie dwóch lub trzech punktów, oraz na chybie i nie zdobycie żadnego punktu. Wartość zmiennej  $P_f$  informuje nas jakie są szanse na wystąpienie akcji faulu co daje szansę na zdobycie punktów. Wartość zmiennej  $P_k$  jest prawdopodobieństwem na utratę piłki w danej akcji, przekładając się na nie zdobycie punktów w tej akcji.

Liczba  $P1$  steruje podstawową akcją przewodzącą podczas aktualnej akcji meczu. Ma ona wpływ na wybór kierunku w jakim potoczy się akcja. Jeśli  $P1$  wpadnie w zakres odpowiedzialny za akcję ataku to aktualna drużyna rozpocznie atak na kosz przeciwnika. Jeśli  $P1$  wpadnie w zakres odpowiedzialny za faul to aktualna drużyna popełniła faul, który zostanie rozstrzygnięty. Jeśli  $P1$  nie zostanie zliczone do poprzednich zakresów to drużyna na początku akcji utraci piłkę. Liczba  $P2$  rozstrzyga, czy drużynie uda się oddać poprawny rzut do kosza lub też popełni faul w ataku. Liczba  $P3$  mówi, czy poprawny rzut do koszy odbył się z bliższej odległości, czy też dalszej – w jednym wypadku drużyna otrzyma 2 punkty, a w drugim 3 punkty. Liczba  $P4$  i  $P5$  pomaga ustalić, czy faul popełniony w ataku zakończył się punktem dla przeciwnika wskutek rzutu osobistego. Rzut osobisty daje 1 punkt dla przeciwnika.

1. Sprawdzenie Klasy drużyny
2. Przypisanie do zmiennej  $P_a$  prawdopodobieństwa wystąpienia akcji ataku
3. Przypisanie do zmiennej  $P_f$  prawdopodobieństwa wystąpienia akcji faulu
4. Przypisanie do zmiennej  $P_k$  prawdopodobieństwa wystąpienia akcji utraty piłki
5. Losowanie  $P1$  z przedziału  $[0, 1]$
6. Jeśli  $0 < P1 \leq P_a$  to:
  - 6.1. Losowanie  $P2$  z przedziału  $[0, 1]$
  - 6.2. Jeśli  $0 < P2 \leq P_a$  to:
    - 6.2.1. Losowanie  $P3$  z przedziału  $[0, 1]$
    - 6.2.2. Jeśli  $P3 \leq P_a$  to:
      - 6.2.2.1. Drużyna zdobywa trzy punkty
    - 6.2.3. W przeciwnym razie:
      - 6.2.3.1. Drużyna zdobywa dwa punkty
  - 6.3. Jeśli  $P_a < P2 \leq P_a + P_f$  to:
    - 6.3.1. Losowanie  $P4$  z przedziału  $[0, 1]$
    - 6.3.2. Jeśli  $1 - P_f < P4$  to:
      - 6.3.2.1. Faul z udanym rzutem osobistym dla przeciwnika
    - 6.3.3. W przeciwnym wypadku:
      - 6.3.3.1. Faul bez udanego rzutu osobistego dla przeciwnika
  - 6.4. W przeciwnym razie
    - 6.4.1. Drużyna utraciła piłkę

7. Jeśli  $P_a < P1 \leq P_a + P_f$  to:
  - 7.1. Losowanie **P5** z przedziału [0, 1]
  - 7.2. Jeśli  $1 - P_f < P5$  to:
    - 7.2.1. **Faul z udanym rzutem osobistym dla przeciwnika**
  - 7.3. W przeciwnym razie:
    - 7.3.1. **Faul bez udanego rzutu osobistego dla przeciwnika**
8. W przeciwnym razie:
  - 8.1. **Drużyna utraciła piłkę**

#### 2.2.4 Algorytm zliczania punktów w podsumowaniu grupy

Po rozegraniu wszystkich meczów w ramach jednej grupy podliczane są punkty wyników. Za przystąpienie do meczu drużyna dostaje jeden punkt, a za wygranie go jeden dodatkowy. Za przegraną nie dostaje się żadnych punktów. Więc drużyna z trzema zwycięstwami ma punktów 6, a drużyna z trzema porażkami ma ich 3.

##### 2.2.4.1 Algorytm krokowy zliczania punktów

1. Ustal zmienną  $P=3$
2. Dla każdego meczu  $M$  rozegranego przez drużynę  $D$ :
  - 2.1. Jeśli drużyna  $D$  ma więcej punktów niż jej przeciwnik w meczu  $M$  to:
    - 2.1.1. Zwiększ  $P$  o 1
  - 2.2. Jeśli to ostatni mecz dla drużyny  $D$  to zakończ pętlę

#### 2.2.5 Algorytm tworzenia grup do drugiej fazy grupowej

W drugiej fazie grupowej najlepsze drużyny z grup A-H są umieszczane w grupach I-L. Odbyna się to poprzez wybór dwóch najlepszych drużyn z każdej z grup A-H i przeniesienie ich do odpowiedniej grupy I-L zgodnie z poniższym schematem:

- Dwie najlepsze drużyny z grupy A i dwie najlepsze drużyny z grupy B umieszczane są w grupie I.
- Dwie najlepsze drużyny z grupy C i dwie najlepsze drużyny z grupy D umieszczane są w grupie J.
- Dwie najlepsze drużyny z grupy E i dwie najlepsze drużyny z grupy F umieszczane są w grupie K.
- Dwie najlepsze drużyny z grupy G i dwie najlepsze drużyny z grupy H umieszczane są w grupie L.

#### 2.2.6 Algorytmy tworzenia par w fazie pucharowej

Po rozegraniu meczów w ramach drugiej fazy grupowej jest dostępna informacja, które drużyny z grup I-L przejdą do fazy pucharowej. Faza pucharowa odbywa się w systemie turniejowym co oznacza, że składa się z ćwierćfinałów, półfinałów i finałów. Do ćwierćfinałów dostają się dwie najlepsze drużyny z grup I-L gdzie łączą się ze sobą w pary. Każda para rozgrywa ze sobą tylko jeden mecz, a wygrana drużyna przechodzi do półfinałów.

##### 2.2.6.1 Algorytm tworzenia par w ćwierćfinałach

Tworzenie par walczących w ćwierćfinałach odbywa się poprzez łączenie pierwszych miejsc w grupach z drugimi miejscami w innych grupach. Tak więc jeśli do ćwierćfinałów dociera osiem drużyn (po dwie na każdą grupę z zakresu I-L) to schemat wygląda następująco:

Pierwsza para ćwierćfinałów	Pierwsze miejsce z grupy I	Drugie miejsce z grupy J
-----------------------------	----------------------------	--------------------------



Druga para ćwierćfinałów	Drugie miejsce z grupy I	Pierwsze miejsce z grupy J
Trzecia para ćwierćfinałów	Pierwsze miejsce z grupy K	Drugie miejsce z grupy L
Czwarta para ćwierćfinałów	Drugie miejsce z grupy K	Pierwsze miejsce z grupy L

Tabela 1

Jak widać na tabeli powyżej: pierwsze miejsca z grup I, K łączą się z drugimi miejscami w grupach J, L i odwrotnie tzn., że drugie miejsca z grup I, K łączą się też w pary z pierwszymi miejscami z grup J, L.

#### 2.2.6.2 Algorytm tworzenia par w półfinałach

W celu utworzenia par walczących o dostanie się do finałów niezbędne jest przeprowadzenie meczy w ćwierćfinałach co da odpowiedź na pytanie, które z drużyn dostaną się na początku do półfinałów. W ćwierćfinałach znajdują się cztery pary i czterech zwycięzców, którzy przechodzą do półfinałów i którzy w ramach półfinałów stoczą ze sobą mecze. Mecze te dopierane są na zasadzie przynależności do konkretnej pary w ćwierćfinałach, a to oznacza, że zwycięzca z pierwszej pary ćwierćfinałów łączy się w nową parę ze zwycięzcą z trzeciej pary ćwierćfinałów. Druga para półfinałów składa się natomiast z pierwszego miejsca drugiej pary ćwierćfinałów i pierwszego miejsca czwartej pary ćwierćfinałów.

Pierwsza para półfinałów	Pierwsze miejsce z pierwszej pary ćwierćfinałów	Pierwsze miejsce z trzeciej pary ćwierćfinałów
Druga para półfinałów	Pierwsze miejsce z drugiej pary ćwierćfinałów	Pierwsze miejsce z czwartej pary ćwierćfinałów

Tabela 2

#### 2.2.6.3 Algorytm tworzenia par w finałach

W celu utworzenia pary zespołów walczących o tytuł mistrza niezbędne jest rozegranie rozgrywek w parach z półfinałów. Zwycięzcy z każdej pary utworzą parę finałową.

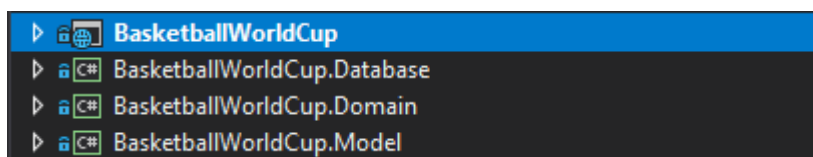
Para finałowa	Pierwsze miejsce z pierwszej pary półfinałów	Pierwsze miejsce z drugiej pary półfinałów
---------------	--	--

Tabela 3

## 3 Opis programu

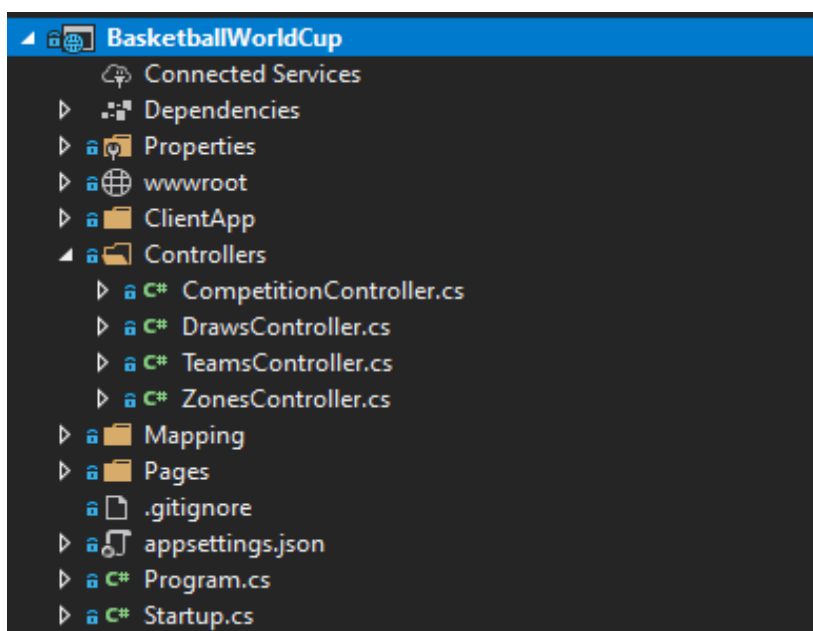
### 3.1 Opis struktury programu

Aplikacja składa się z czterech bibliotek utożsamiających trzy warstwy aplikacji układające się w architekturę trójwarstwową [7]. Do warstwy prezentacji należy biblioteka *BasketballWorldCup*, a do warstwy danych należą biblioteki *BasketballWorldCup.Database* i *BasketballWorldCup.Model*. Warstwą logiki biznesowej jest w tym wypadku biblioteka *BasketballWorldCup.Domain*.



Rysunek 1

W bibliotece *BasketballWorldCup* znajdują się głównie kontrolery do obsługi zapytań do serwisu REST API. Dzięki nim konkretne zapytania HTTP są przechwytywane i zmieniane na konkretne akcje programu. Dla przykładu zapytanie HTTP z użyciem metody GET na adres <http://localhost/api/competition/firstRound/1> zwróci rezultaty rozgrywek Pierwszej Fazy Grupowej dla losowanego oznaczonego w bazie danych identyfikatorem o wartości 1.



Rysunek 2

Aplikacja zawiera cztery kontrolery: *CompetitionController*, *DrawsController*, *TeamsController*, *ZonesController*. Najważniejsze są pierwsze dwa, które służą do obsługi symulacji. Klasa *CompetitionController* (Rysunek 3) służy do obsługi rozgrywek w symulacji i podsumowania ich wyników. Posiada metody *FirstRound*, *SecondRound* i *FinalRound* mające za zadanie przeprowadzić mecze w każdej z faz mistrzostw z pomocą metod z klasy *ICompetitionService* (metody *FirstRound*, *SecondRound*, *QuarterFinals*, *SemiFinals*, *FinalRound*), a następnie podsumować wyniki. Natomiast klasa *DrawsController* (Rysunek 4) obsługuje proces losowania drużyn zakwalifikowanych do mistrzostw. Dzięki niej drużyny wybrane przez użytkownika są umieszczane w koszykach (metoda *Get*) i następnie wrzucane są do grup A-H (metoda *Post*).

```

[Route("api/[controller]")]
[ApiController]
public class CompetitionController : ControllerBase
{
    private readonly ICompetitionService _competitionService;
    private readonly IMapper _mapper;

    public CompetitionController(ICompetitionService competitionService, IMapper mapper)
    {
        _competitionService = competitionService;
        _mapper = mapper;
    }

    [HttpGet]
    [Route("firstRound/{drawId}")]
    public IActionResult FirstRound(int drawId)
    {
        // Przeprowadzenie rozgrywek Pierwszej Rundy Drużynowej
        var groupResult = _competitionService.FirstRound(drawId);
        // Podsumowanie wyników: podliczenie zwycięstw, porażek, punktów, etc.
        var resultWithSummaries = _competitionService.GroupsSummaries(groupResult);
        var dto = _mapper.Map<IEnumerable<GroupResultDto>>(resultWithSummaries);
        return Ok(dto);
    }

    [HttpGet]
    [Route("secondRound/{drawId}")]
    public IActionResult SecondRound(int drawId)
    {
        // Przeprowadzenie rozgrywek Drugiej Rundy Drużynowej
        var groupResult = _competitionService.SecondRound(drawId);
        // Podsumowanie wyników: podliczenie zwycięstw, porażek, punktów, etc.
        var resultWithSummaries = _competitionService.GroupsSummaries(groupResult);
        var dto = _mapper.Map<IEnumerable<GroupResultDto>>(resultWithSummaries);
        return Ok(dto);
    }

    [HttpGet]
    [Route("finalRound/{drawId}")]
    public IActionResult FinalRound(int drawId)
    {
        var result = new List<GroupResult>();

        // Przeprowadzenie rozgrywek ćwierćfinałów
        var quarterResult = _competitionService.QuarterFinals(drawId);
        // Podsumowanie wyników: podliczenie zwycięstw, porażek, punktów, etc.
        var quarterWithSummaries = _competitionService.GroupsSummaries(quarterResult);
        result.AddRange(quarterWithSummaries);

        // Przeprowadzenie rozgrywek półfinałów
        var semiResult = _competitionService.SemiFinals(drawId);
        // Podsumowanie wyników: podliczenie zwycięstw, porażek, punktów, etc.
        var semiWithSummaries = _competitionService.GroupsSummaries(semiResult);
        result.AddRange(semiWithSummaries);

        // Przeprowadzenie rozgrywek finałów
        var finalsResult = _competitionService.FinalRound(drawId);
        // Podsumowanie wyników: podliczenie zwycięstw, porażek, punktów, etc.
        var finalsWithSummaries = _competitionService.GroupsSummaries(finalsResult);
        result.AddRange(finalsWithSummaries);

        var dto = _mapper.Map<IEnumerable<GroupResultDto>>(result);
        return Ok(dto);
    }
}

```

Rysunek 3

```

[Route("api/[controller]")]
public class DrawsController : Controller
{
    private readonly IMapper _mapper;
    private readonly IDrawsService _drawsService;

    public DrawsController(IMapper mapper, IDrawsService drawsService)
    {
        _mapper = mapper;
        _drawsService = drawsService;
    }

    [HttpGet("{drawId}")]
    public IActionResult Get(int drawId)
    {
        return Ok();
    }

    [HttpPost]
    public IActionResult Post([FromBody]TeamDto[] teamsDtos)
    {
        var teams = _mapper.Map<IEnumerable<Team>>(teamsDtos);
        var teamsIds = teams.Select(t => t.Id);
        // Losowanie do koszyków
        var draw = _drawsService.SeedPots(teamsIds);
        var drawDto = _mapper.Map<DrawDto>(draw);
        return Ok(drawDto);
    }

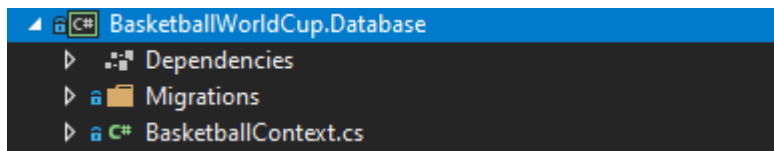
    [HttpPatch]
    [Route("{drawId}")]
    public IActionResult Patch(int drawId)
    {
        // Tworzenie grup A-H na podstawie koszyków
        var draw = _drawsService.AssignGroups(drawId);
        var drawDto = _mapper.Map<DrawDto>(draw);
        return Ok(drawDto);
    }
}

```

Rysunek 4

Biblioteka *BasketballWorldCup.Database* zawiera schemat definicji bazy danych. Ustalany jest on w klasie *BasketballContext* co oznacza, że zaglądając do tej klasy jest możliwość zobaczenie jakie dane mogą zostać pobrane z bazy. Natomiast sama klasa definiuje tabele w bazie danych. Każde pole, takie jak np. *Draws*, jest jedną tabelą.

Poza możliwością definiowania tabel *BasketballContext* posiada jeszcze umiejętność definiowania relacji pomiędzy tabelami bazy danych z użyciem przeciążonej metody *OnModelCreating*. Na przykład pierwsze trzy linie tej metody definiują relację jeden-do-wielu pomiędzy tabelą *Draw* i tabelą *Pots*.



Rysunek 5

```

public class BasketballContext : DbContext
{
    public DbSet<Draw> Draws { get; set; }

    public DbSet<Pot> Pots { get; set; }

    public DbSet<Group> Groups { get; set; }

    public DbSet<Team> Teams { get; set; }

    public DbSet<TeamSummary> TeamSummaries { get; set; }

    public BasketballContext(DbContextOptions<BasketballContext> options) : base(options)
    {
    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<Draw>()
            .HasMany(d => d.Pots)
            .WithOne(p => p.Draw);
        modelBuilder.Entity<Draw>()
            .HasMany(d => d.Groups)
            .WithOne(g => g.Draw);

        modelBuilder.Entity<TeamPot>()
            .HasKey(tp => new { tp.TeamId, tp.PotId });
        modelBuilder.Entity<TeamPot>()
            .HasOne(tp => tp.Pot)
            .WithMany(p => p.TeamPots)
            .HasForeignKey(tp => tp.PotId);
        modelBuilder.Entity<TeamPot>()
            .HasOne(tp => tp.Team)
            .WithMany(t => t.TeamPots)
            .HasForeignKey(tp => tp.TeamId);

        modelBuilder.Entity<TeamGroup>()
            .HasKey(tg => new { tg.TeamId, tg.GroupId });
        modelBuilder.Entity<TeamGroup>()
            .HasOne(tg => tg.Group)
            .WithMany(g => g.TeamGroups)
            .HasForeignKey(tg => tg.GroupId);
        modelBuilder.Entity<TeamGroup>()
            .HasOne(tg => tg.Team)
            .WithMany(t => t.TeamGroups)
            .HasForeignKey(tg => tg.TeamId);

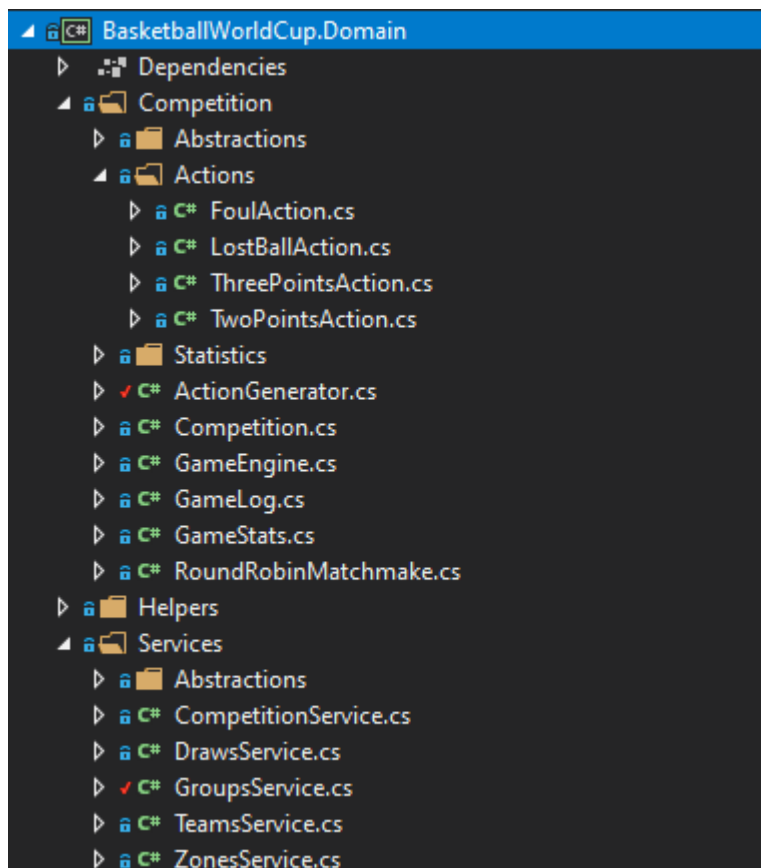
        modelBuilder.Entity<TeamSummary>()
            .HasOne(s => s.Group)
            .WithMany(g => g.Summaries);
        modelBuilder.Entity<TeamSummary>()
            .HasOne(s => s.Team);
    }
}

```

Rysunek 6

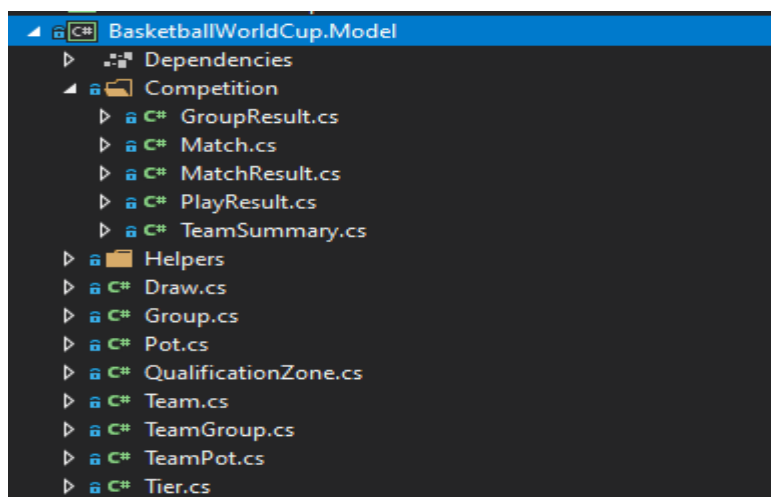
Biblioteka *BasketballWorldCup.Domain* zawiera kod logiki aplikacji stojącej za generowaniem grup do mistrzostw, rozgrywaniem rozgrywek przez drużyny lub tworzeniem podsumowań rozgrywek. Folder *Competition* posiada klasy w których jest zaimplementowana logika silnika rozgrywek (klasa *GameEngine* i jej klasy pomocnicze *GameLog*, *GameStats*), oraz algorytm określający sposób

rozgrywania meczy w ramach drużyn i później w ramach systemu pucharowego (klasa *RoundRobinMatchmake*). Folder *Actions* posiada klasy określające akcje dostępne podczas przeprowadzania rozgrywki (rzuty za punkty, utrata piłki, etc.).



Rysunek 7

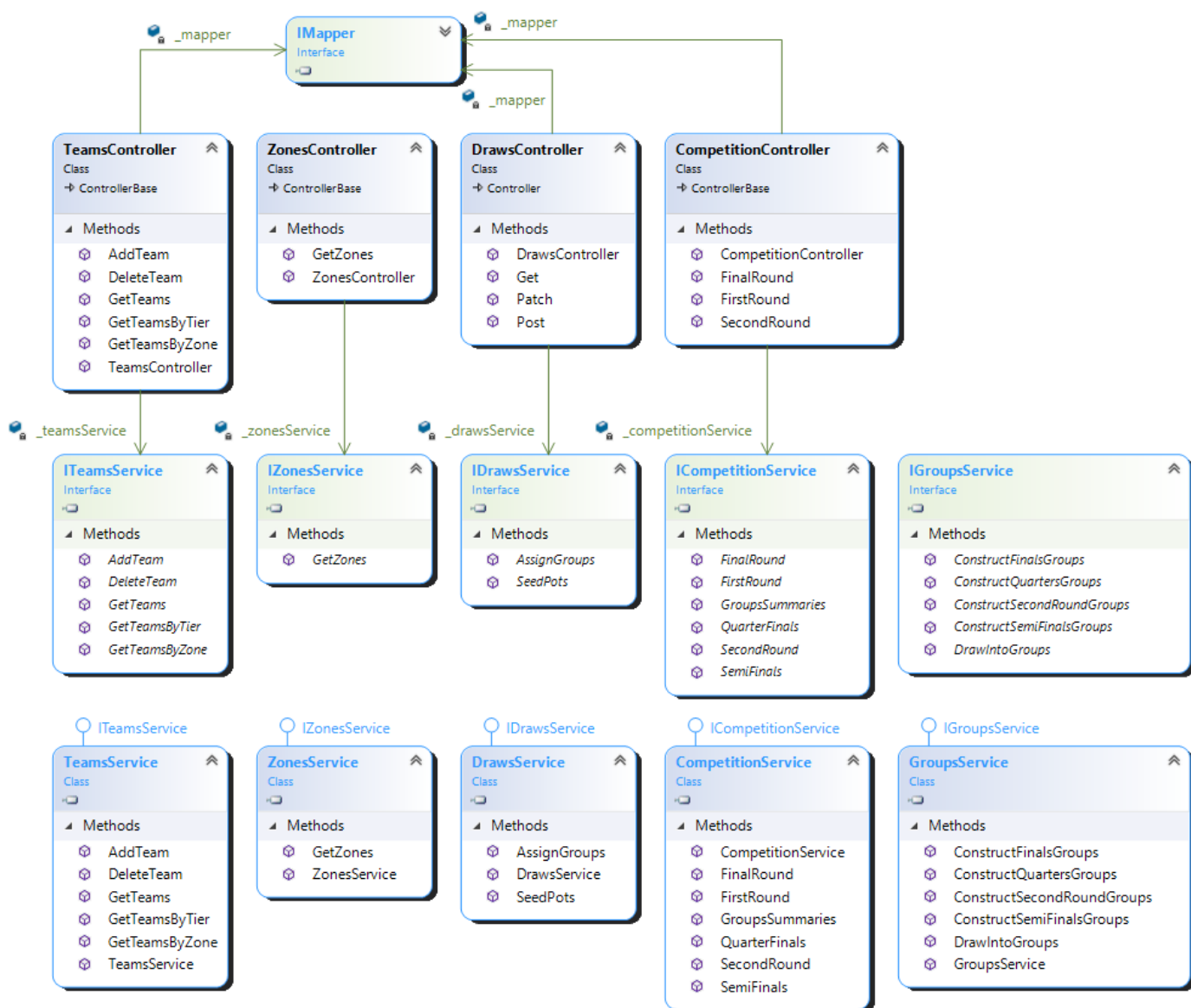
Biblioteka *BasketballWorldCup.Model* posiada obiekty POCO [8] używane na przestrzenie aplikacji.



Rysunek 8

Diagram klas przedstawiony poniżej pokazuje główne klasy, którymi są kontrolery sterujące aplikacją, wraz z serwisami, których używają do wykonania swoich zadań.

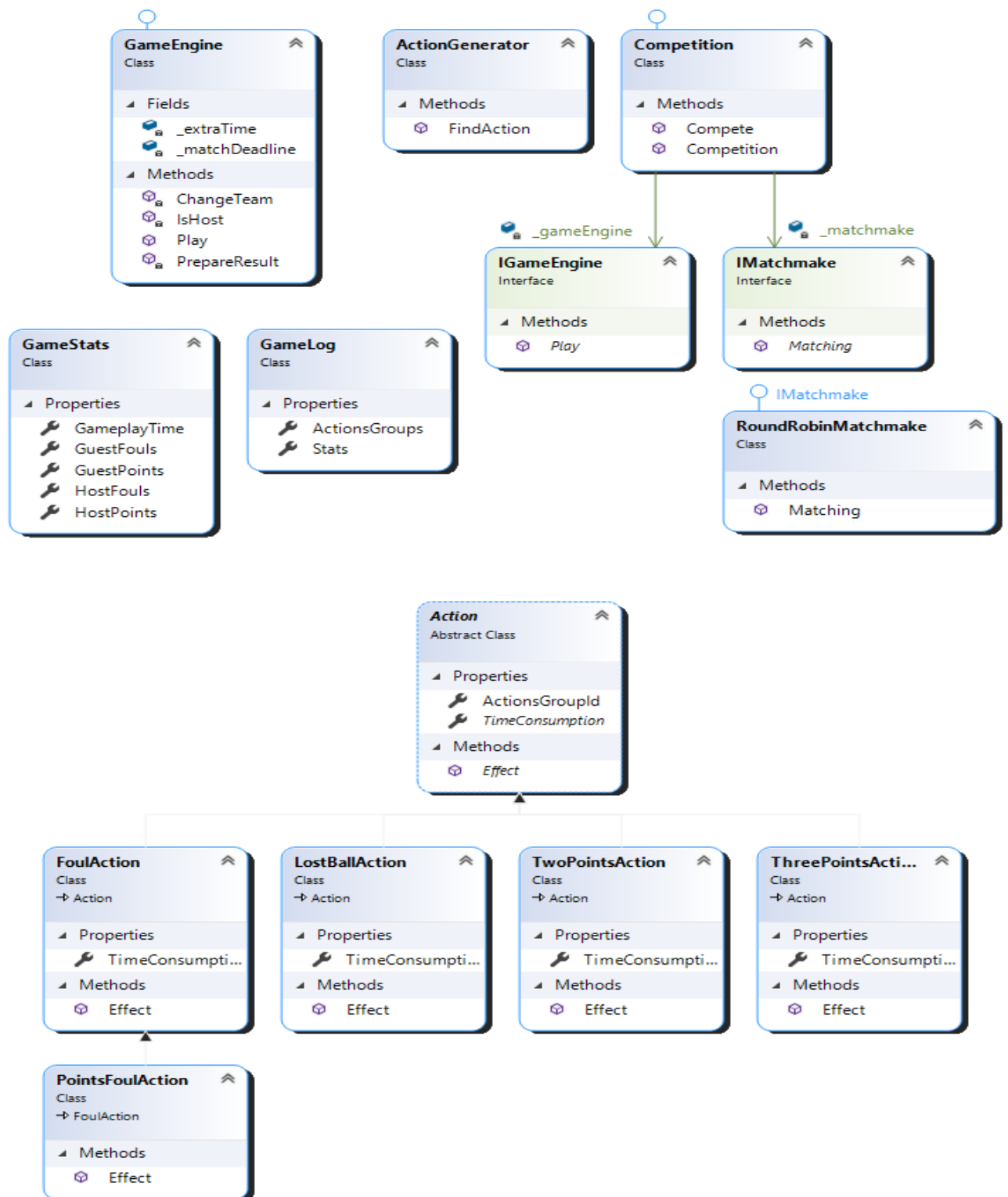
Klasa *TeamsController* wraz z przylegającą jej klasą *ITeamsService* mają za zadanie obsługiwać zapytania do systemu odnośnie drużyn. Najczęstszym sposobem użycia tych klas jest użycie metod mających zwrócić zestawy drużyn podzielone według Klasy (metoda *GetTeamsByTier*) lub według konfederacji (metoda *GetTeamsByZone*). Klasa *ZonesController* wraz z przylegającą jej klasą *IZonesService* mają za zadanie tylko zwrócić informacje o dostępnych konfederacjach. Klasa *DrawsController* wraz z przylegającą jej klasą *IDrawsService* mają za zadanie przeprowadzić losowanie do koszyków (metoda *Get* wywołuje później metodą *SeedPots*), a następnie losowanie do grup A-H (metoda *Patch* wywołuje później metodą *AssignGroups*). Klasa *CompetitionController* wraz z przylegającą jej klasą *ICompetitionService* mają za zadanie przeprowadzić rozgrywki w ramach Pierwszej Fazy Drużynowej (metoda *FirstRound*), Drugiej Fazy Drużynowej (metoda *SecondRound*), oraz Fazy Pucharowej (metoda *FinalRound*).



Rysunek 9



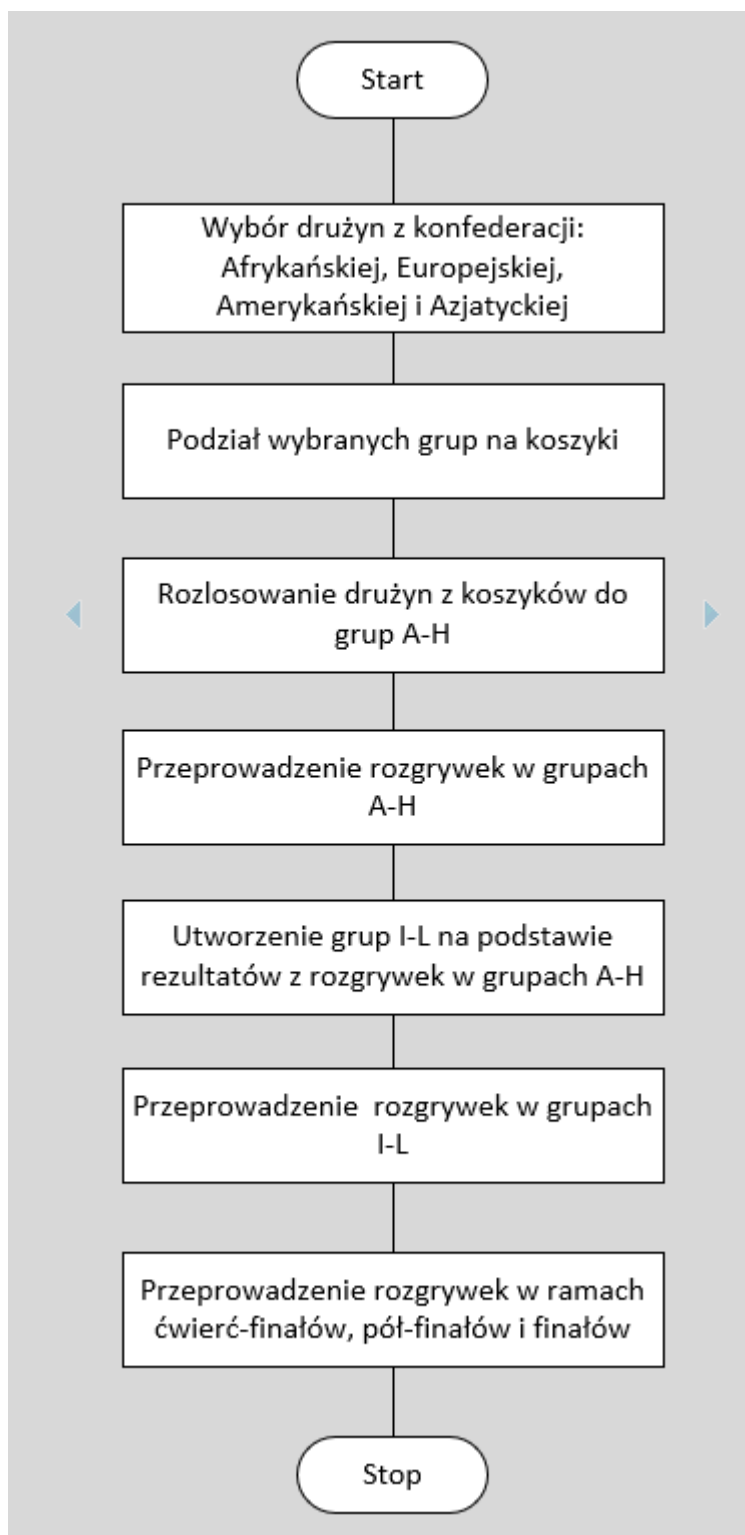
Oprócz klas służących do zarządzania systemem w kodzie aplikacji możemy również wyszczególnić klasy tworzące silnik gry. Główną klasą silnika jest *GameEngine* z jej metodą *Play* służącą do rozpoczynania symulacji. Reszta metod jest pomocnicza i ma za zadanie tylko wspomóc metodę *Play*.



Rysunek 10

### 3.2 Schemat blokowy aplikacji

Aplikacja składa się z szeregu kroków. Pierwszym krokiem jest wybranie przez użytkownika drużyn, które będą ze sobą rywalizować w mistrzostwach i jest to jedyny tak angażujący dla użytkownika proces w aplikacji. Następne kroki polegają na wyświetlaniu danych w taki sposób, aby użytkownik miał możliwość przyjrzenia się результатам poszczególnych faz i rozgrywek. W skład kroków prezentujących rezultaty znajduje się krok podziału wybranych przez użytkownika drużyn na koszyki, z których następuje rozlosowanie do pierwszych grup A-H. Kolejnym krokiem jest przeprowadzenie rozgrywek dla drużyn z grup A-H i wyświetlenie ich wyników. Po czym następuje podział wygranych zespołów z grup A-H na nowe grupy I-L w ramach, których odbywają się rozgrywki w Drugiej Fazie Grupowej. Na koniec po dwa zwycięskie zespoły z każdej z grup I-L rozgrywają ze sobą mecze w ćwierćfinałach, półfinałach i w finałach.



Rysunek 11

### 3.3 Schematy blokowe algorytmów

#### 3.3.1 Algorytm generowania akcji podczas meczu

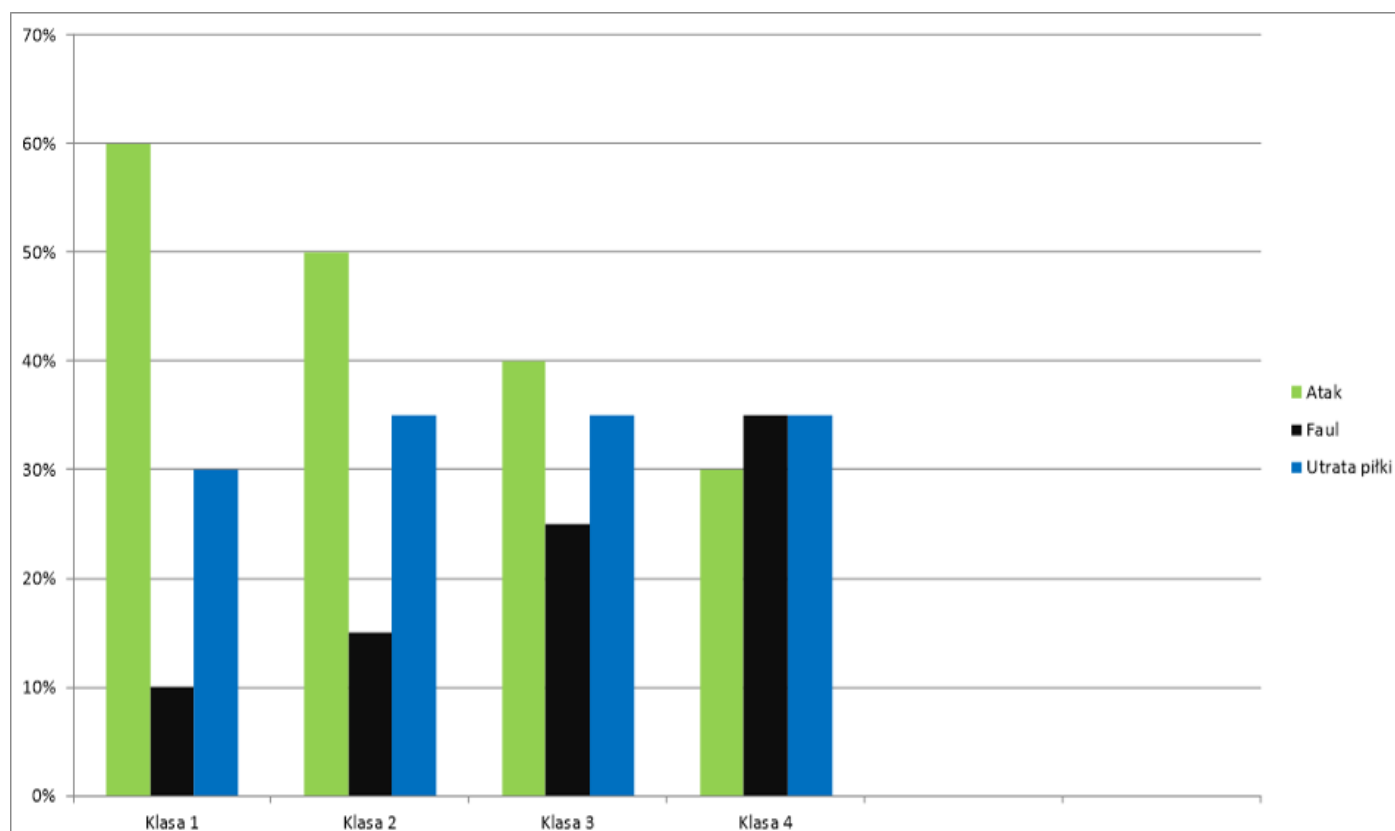
Aplikacja dla każdej Klasy przypisuje odpowiednie statystyki takie jak: prawdopodobieństwo ataku, prawdopodobieństwo obrony przed utratą piłki i prawdopodobieństwo popełnienia faulu.

Każdemu z prawdopodobieństw przypisuje się wartość liczbową z zakresu (0, 1), tak aby ich suma dla siły wynosiła 1 (100%). Liczba ta odpowiada za procentową szansę wystąpienia zdarzenia danego typu.

Zdarzenia *Ataku* odpowiadają za umiejętność zdobywania punktów. Im wyższa ta wartość tym większa szansa na zdobycie punktów.

Akcje *Faulu* służą do karania drużyny za niesportowe zachowanie. Wystąpienie akcji *Faulu* oznacza utratę piłki i podniesienie licznika fauli dla drużyny, u której ta akcja wystąpiła. Jeśli licznik fauli będzie miał wartość większą niż 5 to przy każdym kolejnym faulu drużyna przeciwna dostanie możliwość zdobycia kosza w rzucie osobistym.

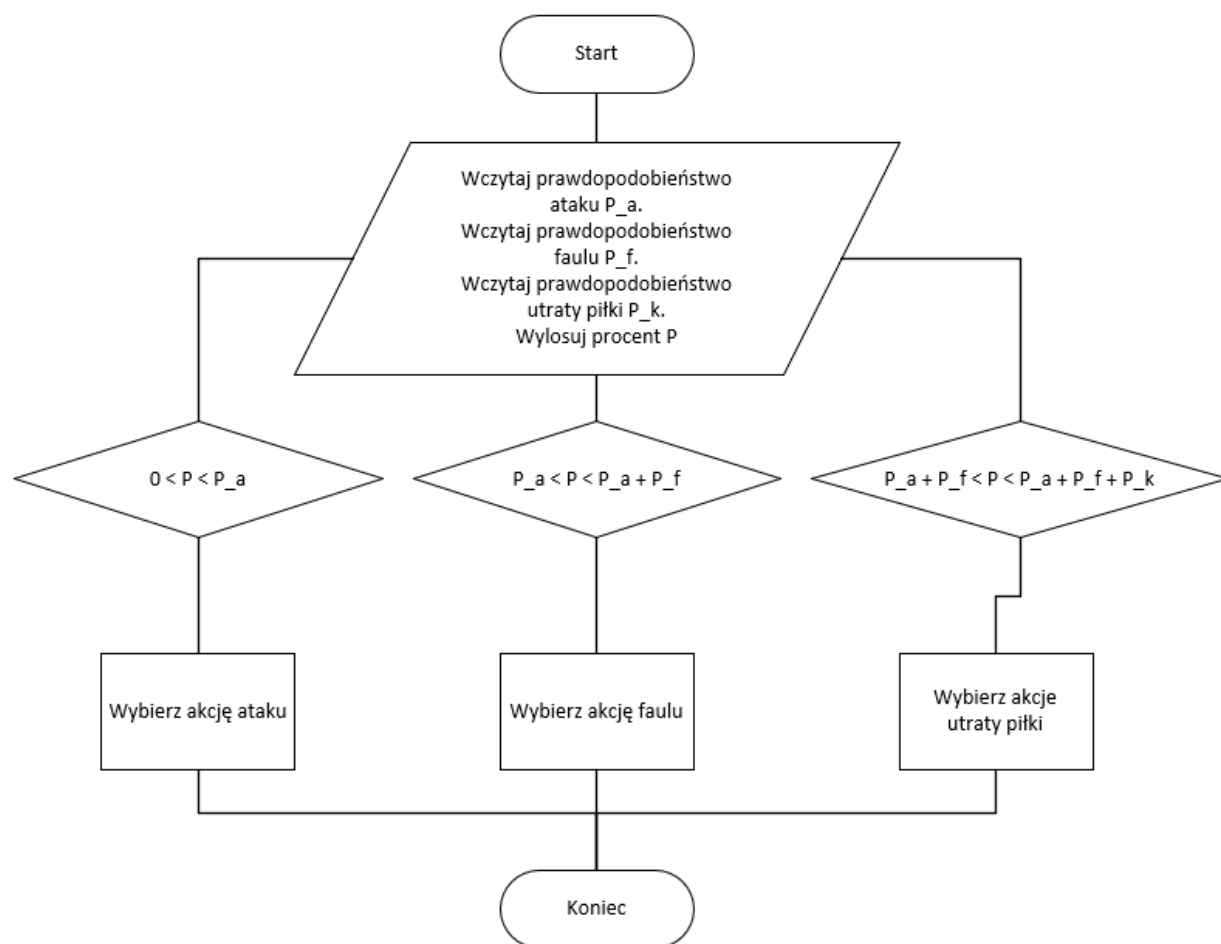
Zdarzenia *Utraty Piłki* oznaczają, że aktualnie grająca drużyna dała sobie ukraść piłkę, więc przechodzi ona do drużyny przeciwnej.



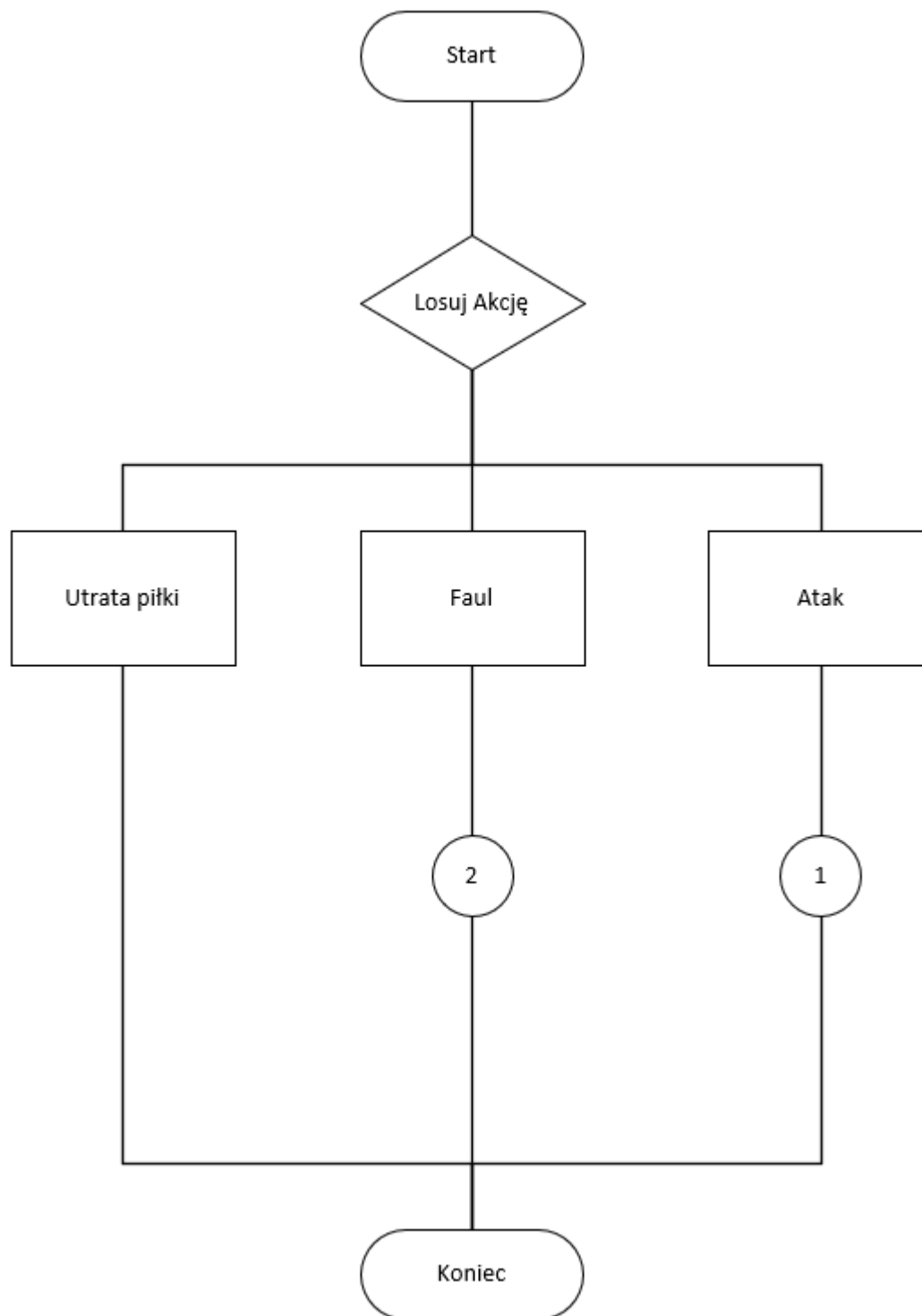
Rysunek 12

Każde z prawdopodobieństw wpływa na różne wydarzenia podczas meczu. Na początku generowania akcji aplikacja sprawdza jaką *Klasę* ma drużyna aktualnie trzymająca piłkę. Na tej podstawie losuje akcję, która wystąpi podczas meczu (Rysunek 14). Do takich akcji zaliczamy atak drużyny, faul lub utratę piłki.

Losowanie akcji odbywa się poprzez wygenerowanie pseudolosowej liczby zmiennoprzecinkowej z zakresu 0-1, która symbolizuje procent. Jeśli wylosowany procent jest mniejszy niż procent *Ataku* dla aktualnej *Klasy* to wybierana jest akcja *Ataku*. Jeśli jest większy niż procent *Ataku*, ale mniejszy niż suma procentu *Ataku* i *Faulu* to wybierana jest akcja *Faulu*. W wypadku, gdy procent jest większy niż suma procentu *Ataku* i *Faulu* to wybierana jest akcja *Utraty Piłki* (Rysunek 13).

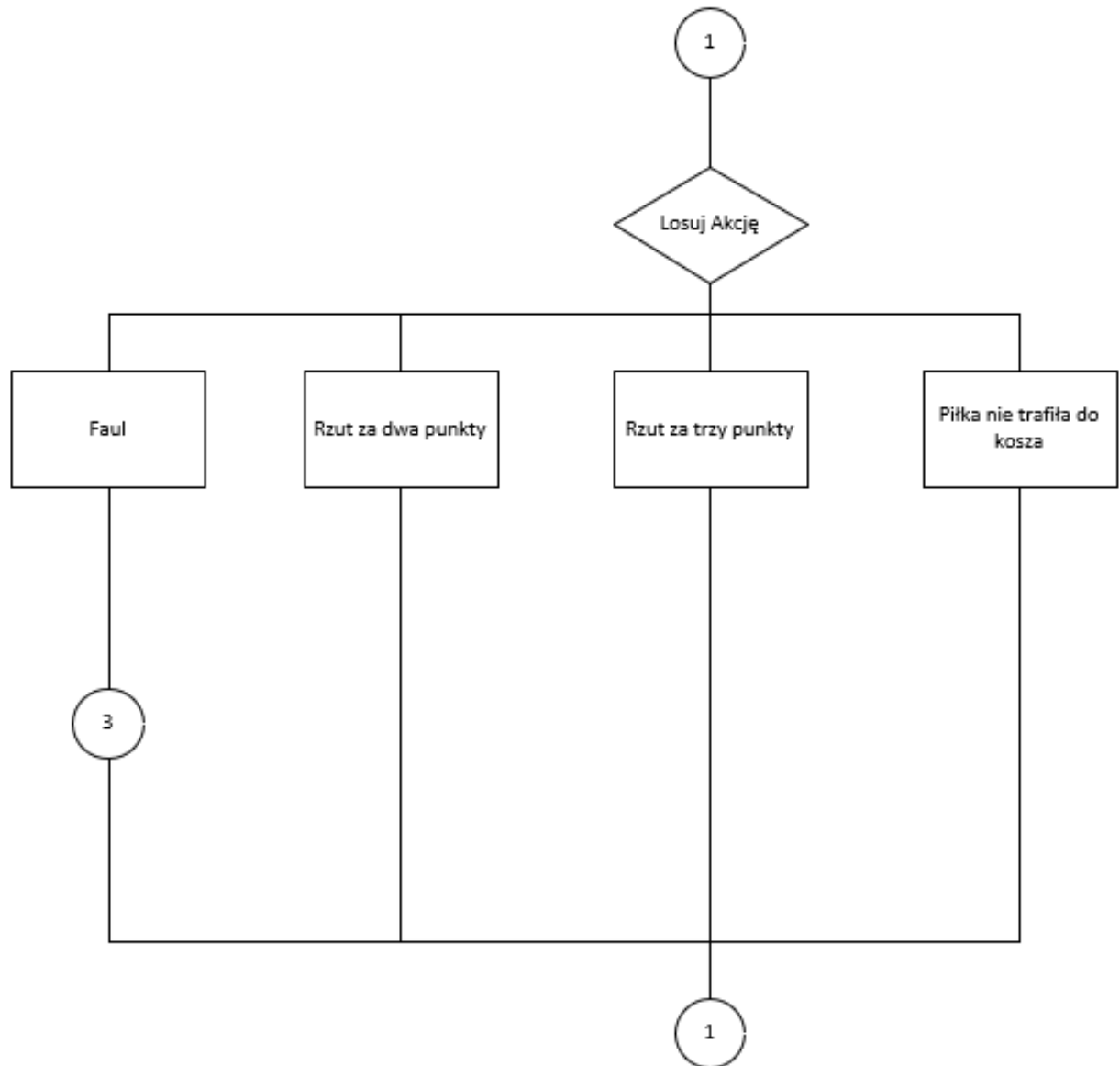


Rysunek 13



Rysunek 14

Wylosowanie akcji *Ataku* oznacza, że drużyna próbuje wykonać rzut do kosza. W tej sytuacji znowu wykonywane jest losowanie akcji, ale teraz gdy zostanie wylosowana akcja *Ataku* sprawdzana jest szansa na zdobycie dwóch lub trzech punktów. Oprócz tego może się też zdarzyć, że piłka nie trafi do kosza lub zostanie popełniony faul w ataku przez atakującego.



Rysunek 15

Po wygenerowaniu akcji dla aktualnej drużyny doliczany jest czas jej trwania do ogólnego czasu trwania meczu. Upłynięcie czasu 40 minut (4 kwarty po 10 minut) jest sygnałem do zakończenia meczu. Każda akcja trwa równo 30 sekund.

Generowanie akcji odbywa się na zmianę dla każdej drużyny.

### 3.4 Kod źródłowy wybranych elementów programu

#### 3.4.1 Kod źródłowy losowania akcji podczas meczu

W celu wygenerowania akcji podczas meczu niezbędne jest podanie wartości *Tier* do metody *FindAction()*. Wartość *Tier* jest reprezentacją *Klasę* danej drużyny w ramach aplikacji. Natomiast metoda *FindAction()* na podstawie podanej *Klasy* pobiera prawdopodobieństwa zaistnienia poszczególnych akcji podczas meczu, takich jak akcja *Ataku*, *Utraty piłki*, lub *Faulu*. Następnie losuję liczbę z przedziału od 0 do 1 i w ramach kilku instrukcji warunkowych *if* sprawdza, czy zaistniała szansa na wystąpienie danej akcji.

```

public Action FindAction(Tier tier)
{
    var random = new Random();
    // Pobranie prawdopodobieństw na zaistnienie poszczególnych akcji
    var statistics = tier.GetStatistics();
    // Losowanie liczby z przedziału od 0 do 1
    var value = random.NextDouble();
    if (value <= statistics.Attack)
    {
        value = random.NextDouble();
        if (value <= statistics.Attack)
        {
            value = random.NextDouble();
            if (value <= statistics.Attack)
            {
                // Wybranie akcji ataku za trzy punkty
                return new ThreePointsAction();
            }
            else
            {
                // Wybranie akcji ataku za dwa punkty
                return new TwoPointsAction();
            }
        }
        else if (statistics.Attack < value && value <= statistics.Attack + statistics.Steal)
        {
            // Wybranie akcji utracenia piłki
            return new LostBallAction();
        }
        else
        {
            value = random.NextDouble();
            if (1 - statistics.Foul < value)
            {
                // Akcja faulu z punktem dla przeciwnika - rzut osobisty udany
                return new PointsFoulAction();
            }
            else
            {
                // Akcja faulu bez punktu dla przeciwnika - rzut osobisty nieudany
                return new FoulAction();
            }
        }
    }
    else if (statistics.Attack < value && value <= statistics.Attack + statistics.Steal)
    {
        // Wybranie akcji utracenia piłki
        return new LostBallAction();
    }
    else
    {
        value = random.NextDouble();
        if (1 - statistics.Foul < value)
        {
            // Akcja faulu z punktem dla przeciwnika - rzut osobisty udany
            return new PointsFoulAction();
        }
        else
        {
            // Akcja faulu bez punktu dla przeciwnika - rzut osobisty nieudany
            return new FoulAction();
        }
    }
}

```

Rysunek 16



### 3.4.2 Kod źródłowy tworzenia grup dla Pierwszej Fazy Grupowej

Kod źródłowy do tworzenia grup dla Pierwszej Fazy Grupowej zawiera trzy metody. Pierwsza z nich *DrawIntoGroups()* przyjmuje obiekty typu *Pots*, które są reprezentacją koszyków w aplikacji, i zajmuje się utworzeniem pustych grup A-H z pomocą metody *FreshFirstRoundGroups()* i dzieli koszyki na dwa zbiory, które następnie są przekształcane na grupy A-H dzięki metodzie *DrawSetIntoGroups()*.

```
public IEnumerable<Group> DrawIntoGroups(Pot[] pots)
{
    // Utworzenie pustych grup A-H
    var groups = FreshFirstRoundGroups().ToList();
    // Podział koszyków na pierwszy zbiór
    var firstPotsSet = new List<Pot>
    {
        pots[0],
        pots[3],
        pots[4],
        pots[7]
    };
    // Dodanie do pierwszego zbioru grup zespołów z pierwszego zbioru koszyków
    DrawSetIntoGroups(groups, firstPotsSet, 0);

    // Podział koszyków na drugi zbiór
    var secondPotsSet = new List<Pot>
    {
        pots[1],
        pots[2],
        pots[5],
        pots[6]
    };
    // Dodanie do drugiego zbioru grup zespołów z drugiego zbioru koszyków
    DrawSetIntoGroups(groups, secondPotsSet, 1);

    return groups;
}
```

Rysunek 17

```

private static IEnumerable<Group> FreshFirstRoundGroups()
{
    const string startingLetter = "A";
    const int groupsAmount = 8;

    var freshFirstRoundGroups = new List<Group>();
    for (var i = 0; i < groupsAmount; i++)
    {
        var group = new Group
        {
            Letter = Encoding.ASCII.GetString(new[] { (byte)(startingLetter[0] + i) }),
            TeamGroups = new List<TeamGroup>()
        };
        freshFirstRoundGroups.Add(group);
    }

    return freshFirstRoundGroups;
}

```

Rysunek 18

```

private static void DrawSetIntoGroups(List<Group> groups, List<Pot> firstPotsSet, int set)
{
    foreach (var group in groups.Where(g => groups.IndexOf(g) % 2 == set))
    {
        foreach (var pot in firstPotsSet)
        {
            var teams = pot.TeamPots.Select(tp => tp.Team).ToArray();
            var randomTeam = teams[_random.Next(0, teams.Length)];

            var randomTeamPot = pot.TeamPots.Single(tp => tp.TeamId == randomTeam.Id);
            pot.TeamPots.Remove(randomTeamPot);

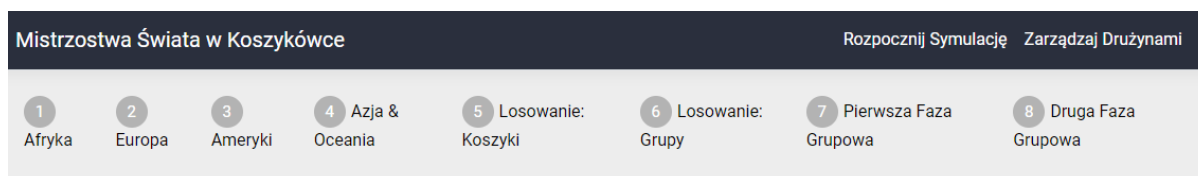
            var teamGroup = new TeamGroup
            {
                Group = group,
                Team = randomTeam
            };
            group.TeamGroups.Add(teamGroup);
        }
    }
}

```

Rysunek 19

### 3.5 Interfejs aplikacji

Główny interfejs aplikacji zawiera menu w postaci paska, na którym znajdują się opcje do rozpoczęcia symulacji lub utworzenia okna do zarządzania dostępnymi drużynami. Tam również znajduje się napis z nazwą aplikacji. Trochę poniżej menu znajdują się ponumerowane kroki służące do łatwiejszego przeprowadzenia użytkownika przez cały proces trwania symulacji.



Rysunek 20

### 3.5.1 Wybór drużyn do mistrzostw

Wybranie drużyn, które wejdą w skład mistrzostw odbywa się ręcznie i jest wykonywane przez użytkownika aplikacji. Zaraz po rozpoczęciu symulacji użytkownik stanie przed wyborem drużyn z różnych konfederacji FIBA. Ich wybór pociągnie za sobą wygenerowanie wyników w ramach kolejnych kroków aplikacji.

#### 3.5.1.1 Rozpoczęcie symulacji

Symulację można rozpocząć wybierając opcję "Rozpocznij Symulację" z menu nawigacyjnego na górze ekranu. Pokaże nam się panel wyboru drużyn dla pierwszej konfederacji.







































1 Afryka 2 Europa 3 Ameryki 4 Azja & Oceania 5 Losowanie: Koszyki 6 Losowanie: Grupy 7 Pierwsza Faza Grupowa

## Kwalifikacje: Africa

☒ Zaznaczono osiem drużyn

☒ Zaznaczono co najmniej dwie drużyny o sile 1

Losuj Drużyny

Zaznaczona	Flaga	Nazwa	Sila				
<input checked="" type="checkbox"/>		Nigeria	1	<input type="checkbox"/>		South Sudan	3
<input type="checkbox"/>		Burundi	3	<input type="checkbox"/>		Angola	1
<input type="checkbox"/>		Guinea	3	<input checked="" type="checkbox"/>		Tunisia	1
<input type="checkbox"/>		Togo	3	<input type="checkbox"/>		Senegal	1
<input type="checkbox"/>		Equatorial Guinea	3	<input type="checkbox"/>		Cote d'Ivoire	1
<input checked="" type="checkbox"/>		Gabon	3	<input type="checkbox"/>		Egypt	1
<input type="checkbox"/>		Burkina Faso	4	<input type="checkbox"/>		Cameroon	1
<input type="checkbox"/>		Zimbabwe	4	<input type="checkbox"/>		Morocco	1
<input checked="" type="checkbox"/>		Zambia	4	<input type="checkbox"/>		Mali	1
<input type="checkbox"/>		Seychelles	4	<input checked="" type="checkbox"/>		CAF	2
<input checked="" type="checkbox"/>		Somalia	4	<input type="checkbox"/>		Dem. Rep. of Congo	2
<input type="checkbox"/>		Benin	4	<input checked="" type="checkbox"/>		Rwanda	2
<input type="checkbox"/>		Botswana	4	<input type="checkbox"/>		Mozambique	2
<input type="checkbox"/>		Mauritania	4	<input type="checkbox"/>		Madagascar	2
<input type="checkbox"/>		Tanzania	4	<input type="checkbox"/>		Cape Verde	2
<input type="checkbox"/>		Eritrea	4	<input type="checkbox"/>		Algeria	2
<input type="checkbox"/>		Libya	4	<input type="checkbox"/>		Uganda	2
<input type="checkbox"/>		South Africa	3	<input checked="" type="checkbox"/>		Chad	2
<input type="checkbox"/>		Kenya	3	<input type="checkbox"/>		Congo	3

← Poprzedni krok


Następny krok →

Rysunek 21

### 3.5.1.2 Wybór drużyny

Następnie należy wybrać drużyny, które wezmą udział w turnieju poprzez kliknięcie na nią.

Wybrana drużyna zostanie zaznaczona. Można również odznaczyć wcześniej zaznaczoną opcję klikając na nią ponownie.

<input checked="" type="checkbox"/>		Poland	1
-------------------------------------	---	--------	---

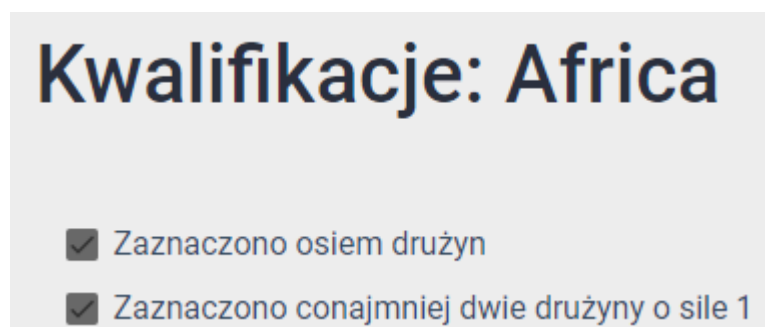
Rysunek 22

### 3.5.1.3 Walidacja

Każdy z panelu wyboru drużyn (jest ich cztery, dla każdej z Konfederacji - Europa, Azja i Oceania, Ameryki oraz Afryka) posiada walidację mówiącą czy podana Konfederacja może przystąpić do turnieju:

1. Każda z Konfederacji musi posiadać dokładnie dwie drużyny o Klasie 1.
2. Każda z Konfederacji musi posiadać łącznie 8 drużyn.

Po spełnieniu tych ograniczeń pola wyboru zostaną zaznaczone jako spełnione.

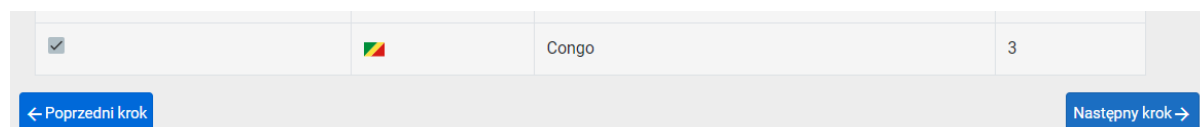


Rysunek 23

### 3.5.1.4 Przejście do następnej Konfederacji

Ponadto po wybraniu drużyn zostanie odblokowany przycisk przekierowujący do wyboru drużyn z następnej Konfederacji.

A w wypadku chęci zmiany poprzedniej Konfederacji przycisk "Poprzedni Krok" pozwolił się cofnąć.



Rysunek 24

## 3.5.2 Losowanie




### 3.5.2.1 Podział na koszyki

Po wybraniu drużyn dla każdej konfederacji aplikacja podzieli nasze drużyny na koszyki na podstawie Klasy każdej z nich.

Najsilniejsze z nich znajdą się w pierwszych koszykach.

1 Afryka 2 Europa 3 Ameryki 4 Azja & Oceania 5 Losowanie: Koszyki 6 Losowanie: Grupy 7 Pierwsza Faza Grupowa

## Losowanie: Koszyki

Koszyk 1			Koszyk 2			Koszyk 3			Koszyk 4		
	Nigeria	1		Jordan	1		Iraq	2		Finland	2
	Spain	1		USA	1		Guam	2		CAF	2
	Serbia	1		Argentina	1		Latvia	2		Rwanda	2
	Japan	1		Tunisia	1		Ukraine	2		Chad	2
Koszyk 5			Koszyk 6			Koszyk 7			Koszyk 8		
	Antigua	2		Portugal	3		Kuwait	4		Zambia	4
	Barbados	2		Gabon	3		Bhutan	4		Somalia	4
	Singapore	3		Bolivia	3		Andorra	4		Peru	4
	Bahrain	3		Nicaragua	3		Malta	4		Guatemala	4

[< Poprzedni krok](#)
[Następny krok >](#)

Rysunek 25

### 3.5.2.2 Podział na grupy A-H

Tworzenie grup odbywa się poprzez wybranie losowego zespołu z każdego koszyka 1, 4, 5, 8 i umieszczenie go do jednej z grup A, C, E, G.

Analogicznie tworzone są grupy B, D, F, H z koszyków 2, 3, 6, 7.

Mistrzostwa Świata w Koszykówce
Rozpocznij Symulację Zarządzaj Drużynami

1 Afryka
2 Europa
3 Ameryki
4 Azja & Oceania
5 Losowanie: Koszyki
6 Losowanie: Grupy
7 Pierwsza Faza Grupowa

## Losowanie: Grupy

Grupa A			Grupa B			Grupa C			Grupa D		
	Japan	1		Argentina	1		Serbia	1		Tunisia	1
	Chad	2		Iraq	2		CAF	2		Guam	2
	Singapore	3		Nicaragua	3		Barbados	2		Bolivia	3
	Somalia	4		Bhutan	4		Peru	4		Malta	4

Grupa E			Grupa F			Grupa G			Grupa H		
	Nigeria	1		USA	1		Spain	1		Jordan	1
	Rwanda	2		Ukraine	2		Finland	2		Latvia	2
	Bahrain	3		Portugal	3		Antigua	2		Gabon	3
	Guatemala	4		Andorra	4		Zambia	4		Kuwait	4

← Poprzedni krok
Następny krok →

Rysunek 26

### 3.5.2.3 Przykład podziału na koszyki i losowania drużyn

Poniżej zaprezentowanego przebieg podziału na koszyki w którym występuje osiem drużyn o *Klasie 1*, szesnaście drużyn o *Klasie 2*, cztery drużyny o *Klasie 3* i również cztery drużyny o *Klasie 4*. Na rysunkach zostanie pokazane krok po kroku na jakiej zasadzie są tworzone grupy w aplikacji. Na Rysunek 27 znajduje się podział na koszyki, a na Rysunek 28 został pokazany podział, na grupy który został wygenerowany na podstawie tych koszyków.

FIBA Basketball World Cup
Start Teams Manager

1 Africa
2 Europe
3 Americas
4 Asia & Oceania
5 The Draw: Pots
6 The Draw: Groups

## The Draw: Pots

Pot 1		
	Nigeria	1
	Spain	1
	Serbia	1
	Chinese Taipei	1

Pot 2		
	Lebanon	1
	USA	1
	Argentina	1
	Tunisia	1

Pot 3		
	Qatar	2
	India	2
	Kazakhstan	2
	Iraq	2

Pot 4		
	Latvia	2
	Ukraine	2
	Finland	2
	Georgia	2

Pot 5		
	Cape Verde	2
	Algeria	2
	Uganda	2
	Chad	2

Pot 6		
	Panama	2
	Virgin Islands	2
	Bahamas	2
	Cuba	2

Pot 7		
	Maldives	3
	Sweden	3
	Congo	3
	Bolivia	3

Pot 8		
	Kuwait	4
	Kosovo	4
	Libya	4
	Haiti	4

Previous Step
Next Step

Rysunek 27

FIBA Basketball World Cup
Start Teams Manager

1 Africa
2 Europe
3 Americas
4 Asia & Oceania
5 The Draw: Pots
6 The Draw: Groups

## The Draw: Groups

Group A		
	Nigeria	1
	Georgia	2
	Chad	2
	Kosovo	4

Group B		
	Lebanon	1
	Iraq	2
	Virgin Islands	2
	Bolivia	3

Group C		
	Serbia	1
	Ukraine	2
	Cape Verde	2
	Haiti	4

Group D		
	USA	1
	India	2
	Bahamas	2
	Congo	3

Group E		
	Spain	1
	Latvia	2
	Algeria	2
	Libya	4

Group F		
	Tunisia	1
	Qatar	2
	Panama	2
	Maldives	3

Group G		
	Chinese Taipei	1
	Finland	2
	Uganda	2
	Kuwait	4

Group H		
	Argentina	1
	Kazakhstan	2
	Cuba	2
	Sweden	3

Previous Step
Next Step

Rysunek 28



W przypadku otrzymania takiego podziału na grupy, stały się następujące rzeczy.

1. Z koszyków 1, 4, 5, 8 (zaznaczone czerwonym prostokątem) wybraliśmy losowo po jednej drużynie (zaznaczone niebieską elipsą) i wrzuciliśmy do Grupy A (Rysunek 29).
2. Następnie z tych samych koszyków, pomijając poprzednio wybrane drużyny (czarne przekreślenie), losujemy po jednej drużynie do Grupy C (Rysunek 30).
3. Operację 2 powtarzamy dla grup E i G
4. Analogicznie jak w przypadku kroków 1-3 działamy z grupami B, D, F, H i koszykami 2, 3, 6, 7

FIBA Basketball World Cup Start Teams Manager

1 Africa 2 Europe 3 Americas 4 Asia & Oceania 5 The Draw: Pots 6 The Draw: Groups

## The Draw: Pots

Pot 1	Pot 2	Pot 3	Pot 4
Nigeria 1	Lebanon 1	Qatar 2	Latvia 2
Spain 1	USA 1	India 2	Ukraine 2
Serbia 1	Argentina 1	Kazakhstan 2	Finland 2
Chinese Taipei 1	Tunisia 1	Iraq 2	Georgia 2

Pot 5	Pot 6	Pot 7	Pot 8
Cape Verde 2	Panama 2	Maldives 3	Kuwait 4
Algeria 2	Virgin Islands 2	Sweden 3	Kosovo 4
Uganda 2	Bahamas 2	Congo 3	Libya 4
Chad 2	Cuba 2	Bolivia 3	Haiti 4

← Previous Step Next Step →

Rysunek 29

FIBA Basketball World Cup
Start Teams Manager

1 Africa
2 Europe
3 Americas
4 Asia & Oceania
5 The Draw: Pots
6 The Draw: Groups

## The Draw: Pots

**Pot 1**

	Nigeria	1
	Spain	1
	Serbia	1
	Chinese Taipei	1

**Pot 2**

	Lebanon	1
	USA	1
	Argentina	1
	Tunisia	1

**Pot 3**

	Qatar	2
	India	2
	Kazakhstan	2
	Iraq	2

**Pot 4**

	Latvia	2
	Ukraine	2
	Finland	2
	Georgia	2

**Pot 5**

	Cape Verde	2
	Algeria	2
	Uganda	2
	Shud	2

**Pot 6**

	Panama	2
	Virgin Islands	2
	Bahamas	2
	Cuba	2

**Pot 7**

	Maldives	3
	Sweden	3
	Congo	3
	Bolivia	3

**Pot 8**

	Kuwait	4
	Kosovo	4
	Libya	4
	Haiti	4

← Previous Step
Next Step →

Rysunek 30

### 3.5.3 Pierwsza Faza Grupowa

W ramach każdej grupy odbywają się sparingi mające na celu wybrać dwóch zwycięzców.








Każda drużyna zagra z każdą inną drużyną w grupie. Łącznie odbędzie się 48 rozgrywek (3 przypadają dla każdej drużyny, 6 w każdej grupie).

Interfejs rezultat rozgrywek w Pierwszej Fazie Grupowej zawiera szereg rzędów tabel. W każdym rzędzie znajdują się dwie tabele odnoszące się do jednej grupy. Tabela po lewej z tytułem „Wynik meczów grupy X” zawiera spis wszystkich meczy w ramach jednej grupy. Są tam informacje jakie drużyny brały udział w meczu i ile punktów udało im się zdobyć. Tabela po prawej z tytułem „Podsumowanie grupy X” zawiera informacje o ilości zwycięstw i punktów przypisanych za nie.

1 Afryka 2 Europa 3 Ameryki 4 Azja & Oceania 5 Losowanie: Koszyki 6 Losowanie: Grupy 7 Pierwsza Faza Grupowa

## Pierwsza Faza Grupowa

### Wynik meczów grupy A

Gospodarz				Gość
 Singapore	22	vs	63	 Japan
 Singapore	38	vs	17	 Somalia
 Singapore	33	vs	47	 Chad
 Japan	80	vs	25	 Somalia
 Japan	80	vs	45	 Chad
 Somalia	29	vs	49	 Chad

### Podsumowanie grupy A

Drużyna	Wynik	Zwycięstwa	Przegrane
 Japan	6	3	0
 Chad	5	2	1
 Singapore	4	1	2
 Somalia	3	0	3

Rysunek 31

## Wynik meczów grupy B

Gospodarz				Gość
 Bhutan	15	vs	45	 Iraq
 Bhutan	18	vs	57	 Argentina
 Bhutan	7	vs	18	 Nicaragua
 Iraq	54	vs	62	 Argentina
 Iraq	43	vs	40	 Nicaragua
 Argentina	73	vs	32	 Nicaragua





## Podsumowanie grupy B

Drużyna	Wynik	Zwycięstwa	Przegrane
 Argentina	6	3	0
 Iraq	5	2	1
 Nicaragua	4	1	2
 Bhutan	3	0	3

## Wynik meczów grupy C

Gospodarz				Gość
 Serbia	66	vs	52	 CAF
 Serbia	51	vs	22	 Peru
 Serbia	68	vs	30	 Barbados
 CAF	43	vs	21	 Peru
 CAF	45	vs	47	 Barbados
 Peru	18	vs	34	 Barbados

## Podsumowanie grupy C

Drużyna	Wynik	Zwycięstwa	Przegrane
 Serbia	6	3	0
 Barbados	5	2	1
 CAF	4	1	2
 Peru	3	0	3

Rysunek 32

### 3.5.4 Druga Faza Grupowa

W ramach każdej grupy odbywają się sparingi mające na celu wybrać dwóch zwycięzców, którzy przejdą do kolejnej fazy.

Każda drużyna zagra z każdą inną drużyną w grupie. Łącznie odbędą się 24 rozgrywki (3 przypadają dla każdej drużyny, 6 w każdej grupie).

Interfejs rezultat rozgrywek w Drugiej Fazie Grupowej zawiera szereg rzędów tabel. W każdym rzędzie znajdują się dwie tabele odnoszące się do jednej grupy. Tabela po lewej z tytułem „Wynik meczów grupy X” zawiera spis wszystkich meczy w ramach jednej grupy. Znajdują się tam informacje jakie drużyny brały udział w meczu i ile punktów udało im się zdobyć. Tabela po prawej z tytułem „Podsumowanie grupy X” zawiera informacje o ilości zwycięstw i punktów przypisanych za nie.

Mistrzostwa Świata w Koszykówce

Rozpocznij Symulację Zarządzaj Drużynami

1  
Afryka

2  
Europa

3  
Ameryki

4  
Azja &  
Oceania

5  
Losowanie:  
Koszyki

6  
Losowanie:  
Grupy

7  
Pierwsza Faza  
Grupowa

8  
Druga Faza  
Grupowa

## Druga Faza Grupowa

### Wynik meczów grupy I

Gospodarz				Gość
Angola	67	vs	38	Bulgaria
Angola	63	vs	53	Netherlands
Angola	64	vs	77	Cote d'Ivoire
Bulgaria	42	vs	42	Netherlands
Bulgaria	30	vs	41	Cote d'Ivoire
Netherlands	51	vs	63	Cote d'Ivoire

### Podsumowanie grupy I

Drużyna	Wynik	Zwycięstwa	Przegrane
Cote d'Ivoire	6	3	0
Angola	5	2	1
Bulgaria	3	0	3
Netherlands	3	0	3

Rysunek 33

### Wynik meczów grupy J

Gospodarz				Gość
Jordan	67	vs	45	Chile
Jordan	66	vs	77	Mexico
Jordan	70	vs	20	Estonia
Chile	38	vs	74	Mexico
Chile	61	vs	28	Estonia
Mexico	48	vs	35	Estonia

### Podsumowanie grupy J

Drużyna	Wynik	Zwycięstwa	Przegrane
Mexico	6	3	0
Jordan	5	2	1
Chile	4	1	2
Estonia	3	0	3

### Wynik meczów grupy K

Gospodarz				Gość
Greece	72	vs	50	Madagascar
Greece	46	vs	66	Iran
Greece	61	vs	42	Mozambique
Madagascar	47	vs	51	Iran
Madagascar	39	vs	35	Mozambique
Iran	61	vs	37	Mozambique

### Podsumowanie grupy K

Drużyna	Wynik	Zwycięstwa	Przegrane
Iran	6	3	0
Greece	5	2	1
Madagascar	4	1	2
Mozambique	3	0	3

Rysunek 34

#### 3.5.5 Faza Pucharowa

Interfejs aplikacji dla Fazy Pucharowej mistrzostw zawiera wyniki meczów rozgrywanych w ramach ćwierćfinałów, półfinałów i finałów. Pary utworzone dla rozgrywek ćwierćfinałowych mają oznaczenia od Q1 do Q4, gdzie Q1 jest pierwszą parą ćwierćfinałów, Q2 drugą parą ćwierćfinałów, Q3 trzecią parą, a Q4 czwartą parą. Takiej samej zasadzie funkcjonują pary z półfinałów, które posiadają znaczniki od S1 do S2. Natomiast para finałowa mistrzostw posiada znacznik F1.

Q1	Pierwsza para ćwierćfinałów
Q2	Druga para ćwierćfinałów
Q3	Trzecia para ćwierćfinałów
Q4	Czwarta para ćwierćfinałów
S1	Pierwsza para półfinałów
S2	Druga para półfinałów
F1	Para finałowa

Tabela 4

Mistrzostwa Świata w Koszykówce
Rozpocznij Symulację
Zarządzaj Drużynami

1 Afryka
2 Europa
3 Ameryki
4 Azja & Oceania
5 Losowanie: Koszyki
6 Losowanie: Grupy
7 Pierwsza Faza Grupowa
8 Druga Faza Grupowa
8 Faza Pucharowa

## Faza Pucharowa

### Wynik meczów pary Q1



Gospodarz				Gość
Nigeria	64	vs	68	Spain

### Podsumowanie pary Q1



Drużyna	Wynik	Zwycięstwa	Przegrane
Spain	2	1	0
Nigeria	1	0	1

Rysunek 35



## Wynik meczów pary Q1

Gospodarz				Gość
 Nigeria	64	vs	68	 Spain

## Podsumowanie pary Q1

Drużyna	Wynik	Zwycięstwa	Przegrane
 Spain	2	1	0
 Nigeria	1	0	1



## Wynik meczów pary Q2

Gospodarz				Gość
 Dominican Republic	57	vs	52	 Korea



## Podsumowanie pary Q2

Drużyna	Wynik	Zwycięstwa	Przegrane
 Dominican Republic	2	1	0
 Korea	1	0	1



## Wynik meczów pary Q3

Gospodarz				Gość
 Cote d'Ivoire	70	vs	78	 Croatia



## Podsumowanie pary Q3

Drużyna	Wynik	Zwycięstwa	Przegrane
 Croatia	2	1	0
 Cote d'Ivoire	1	0	1

## Wynik meczów pary Q4

Gospodarz				Gość
 Uruguay	65	vs	50	 Philippines



## Podsumowanie pary Q4

Drużyna	Wynik	Zwycięstwa	Przegrane
 Uruguay	2	1	0
 Philippines	1	0	1



Rysunek 36





## Wynik meczów pary S1

Gospodarz				Gość
 Spain	77	vs	53	 Dominican Republic



## Podsumowanie pary S1

Drużyna	Wynik	Zwycięstwa	Przegrane
 Spain	2	1	0
 Dominican Republic	1	0	1

## Wynik meczów pary S2



Gospodarz				Gość
 Croatia	59	vs	66	 Uruguay

## Podsumowanie pary S2



Drużyna	Wynik	Zwycięstwa	Przegrane
 Uruguay	2	1	0
 Croatia	1	0	1

Rysunek 37

## Wynik meczów pary F1

Gospodarz				Gość
 Spain	71	vs	58	 Uruguay

## Podsumowanie pary F1


Drużyna	Wynik	Zwycięstwa	Przegrane
 Spain	2	1	0
 Uruguay	1	0	1

Rysunek 38

## 4 Instrukcja obsługi














### 4.1 Uruchomienie aplikacji

Po włożeniu płytki CD do napędu powinniśmy być w stanie odczytać jej zawartość. Jako pierwsze użytkownik dostrzeże folder *BasketballWorldCup*, który powinien przekopiować do dogodnej dla siebie lokalizacji.

Name	Date modified	Type
 BasketballWorldCup	18.05.2020 16:40	File folder

















Rysunek 39

W głównym folderze jeszcze raz wchodzimy do folderu o nazwie *BasketballWorldCup*.

Name	Date modified	Type
 BasketballWorldCup	12.09.2020 12:47	File folder
 BasketballWorldCup.Database	31.05.2020 13:17	File folder
 BasketballWorldCup.Domain	12.09.2020 12:47	File folder
 BasketballWorldCup.Model	31.05.2020 13:17	File folder
 BasketballWorldCup.Tests	10.04.2020 18:55	File folder
 docs	17.09.2020 18:05	File folder
 images	12.09.2020 12:56	File folder
 SeedingTool	10.04.2020 18:55	File folder
 Tests	15.04.2020 06:21	File folder
 .gitattributes	10.03.2020 19:58	Text Document
 .gitignore	10.03.2020 19:58	Text Document
 BasketballWorldCup.sln	19.04.2020 16:10	Visual Studio Solution
 README.md	17.05.2020 13:36	MD File


Rysunek 40

Następnie do folderu *bin*.


Name	Date modified	Type
 bin	10.03.2020 19:57	File folder
 ClientApp	17.09.2020 17:49	File folder
 Controllers	05.09.2020 14:20	File folder
 Mapping	31.05.2020 13:43	File folder
 obj	17.09.2020 17:48	File folder
 Pages	10.03.2020 19:57	File folder
 Properties	10.03.2020 19:57	File folder
 wwwroot	10.03.2020 19:57	File folder
 .gitignore	10.03.2020 19:57	Text Document
 appsettings.Development.json	10.03.2020 19:57	JSON File
 appsettings.json	10.03.2020 20:09	JSON File
 BasketballWorldCup.csproj	03.09.2020 09:13	Visual C# Project ...
 BasketballWorldCup.csproj.user	13.04.2020 13:06	Per-User Project O...
 MainDiagram.cd	12.09.2020 12:52	Class Diagram File
 Program.cs	10.03.2020 20:19	Visual C# Source f...
 Startup.cs	13.05.2020 20:01	Visual C# Source f...

Rysunek 41

















Następnie folder *Debug*, *netcoreapp3.1*, *publish* i na końcu folder *win10-x64*

Name	Date modified	Type
 Debug	10.03.2020 19:57	File folder
















Rysunek 42

Name	Date modified	Type
 netcoreapp3.1	17.09.2020 17:48	File folder

Rysunek 43

Name	Date modified	Type
 cs	10.03.2020 20:31	File folder
 de	10.03.2020 20:31	File folder
 es	10.03.2020 20:31	File folder
 fr	10.03.2020 20:31	File folder
 it	10.03.2020 20:31	File folder
 ja	10.03.2020 20:31	File folder
 ko	10.03.2020 20:31	File folder
 pl	10.03.2020 20:31	File folder
 Properties	10.03.2020 20:10	File folder
 pt-BR	10.03.2020 20:31	File folder
 ru	10.03.2020 20:31	File folder
 runtimes	14.03.2020 18:16	File folder
 tr	10.03.2020 20:31	File folder
 win10-x64	17.09.2020 17:48	File folder
 zh-Hans	10.03.2020 20:31	File folder
 zh-Hant	10.03.2020 20:31	File folder

Rysunek 44

Name	Date modified	Type
 cs	17.09.2020 17:48	File folder
 de	17.09.2020 17:48	File folder
 es	17.09.2020 17:48	File folder
 fr	17.09.2020 17:48	File folder
 it	17.09.2020 17:48	File folder
 ja	17.09.2020 17:48	File folder
 ko	17.09.2020 17:48	File folder
 pl	17.09.2020 17:48	File folder
 Properties	17.09.2020 17:48	File folder
 pt-BR	17.09.2020 17:48	File folder
 publish	17.09.2020 17:50	File folder
 ru	17.09.2020 17:48	File folder
 tr	17.09.2020 17:48	File folder
 zh-Hans	17.09.2020 17:48	File folder
 zh-Hant	17.09.2020 17:48	File folder

Rysunek 45

Folder *publish* zawiera wszystkie pliki potrzebne do uruchomienia aplikacji. Samo uruchomienie odbywa się poprzez podwójne kliknięcie na plik *BasketballWorldCup.exe*.

Name	Date modified	Type	Size
ClientApp	17.09.2020 17:50	File folder	
wwwroot	17.09.2020 17:50	File folder	
appsettings.Development.json	10.03.2020 19:57	JSON File	1 KB
appsettings.json	10.03.2020 20:09	JSON File	1 KB
BasketballWorldCup.exe	17.09.2020 17:50	Application	112 814 KB
BasketballWorldCup.pdb	17.09.2020 17:48	Program Debug D...	28 KB
web.config	17.09.2020 17:50	XML Configuratio...	1 KB

Rysunek 46

Po uruchomieniu aplikacji pojawi się okno konsoli, które informuje jaki adres należy wpisać w oknie przeglądarki, aby móc skorzystać z aplikacji internetowej. Adres ten to *http://localhost:5000*.

```

C:\Users\Damian\source\repos\BasketballWorldCup\BasketballWorldCup\bin\Debug\netcoreapp3.1\win10-x64\publish\BasketballWorldCup.exe
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Damian\source\repos\BasketballWorldCup\BasketballWorldCup\bin\Debug\netcoreapp3.1\win10-x64\publish

```

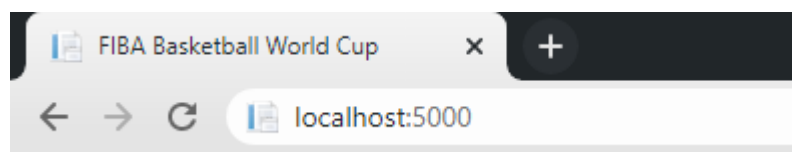
Rysunek 47

```

info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000

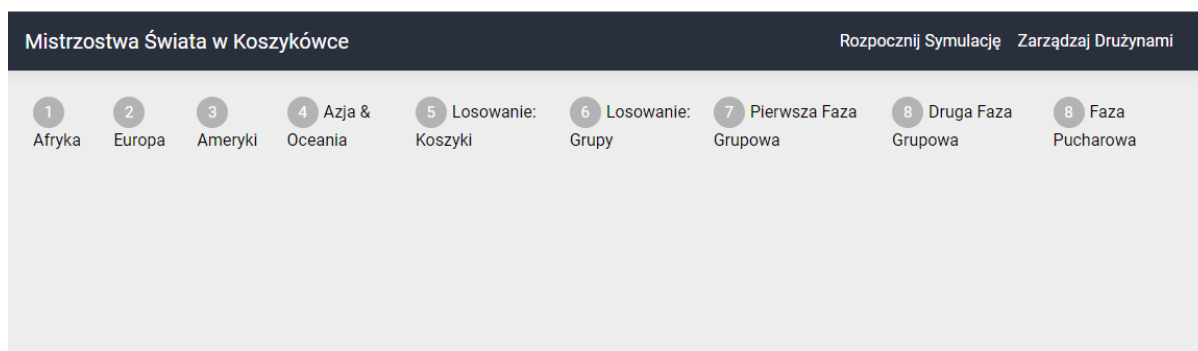
```

Rysunek 48



Rysunek 49

Po pokazaniu się aplikacji w przeglądarce zobaczymy poniższy ekran.

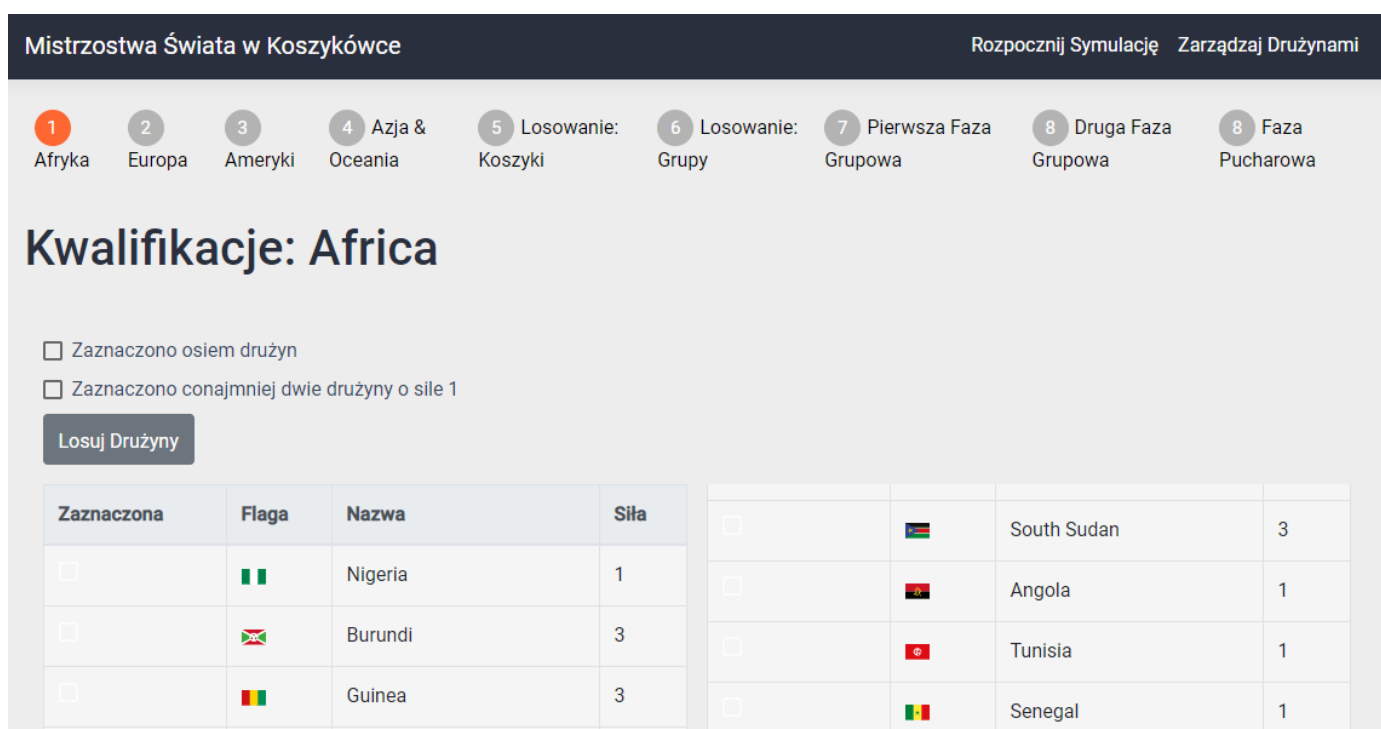


Rysunek 50

W celu rozpoczęcia symulacji i przejścia do wyboru drużyn klikamy na przycisk oznaczony numerem 1 z napisem „Afryka”. Spowoduje to wyświetlenie ekranu wyboru drużyn do mistrzostw.




Rysunek 51



Rysunek 52

Następnie należy wybrać drużyny, które wezmą udział w turnieju poprzez kliknięcie na nią.

Wybrana drużyna zostanie zaznaczona. Można również odznaczyć wcześniej zaznaczoną opcję klikając na nią ponownie.

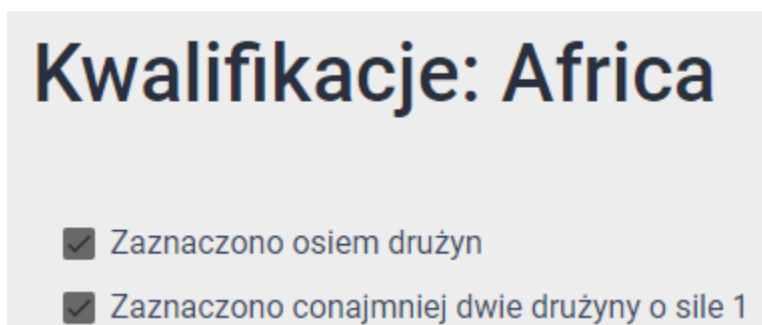
<input checked="" type="checkbox"/>		Poland	1
-------------------------------------	---	--------	---

Rysunek 53

Każdy z panelu wyboru drużyn (jest ich cztery, dla każdej z Konfederacji - Europa, Azja i Oceania, Ameryki oraz Afryka) posiada walidację mówiącą czy podana Konfederacja może przystąpić do turnieju:

1. Każda z Konfederacji musi posiadać dokładnie dwie drużyny o Klasie 1.
2. Każda z Konfederacji musi posiadać łącznie 8 drużyn.


Po spełnieniu tych ograniczeń pola wyboru zostaną zaznaczone jako spełnione.



Rysunek 54

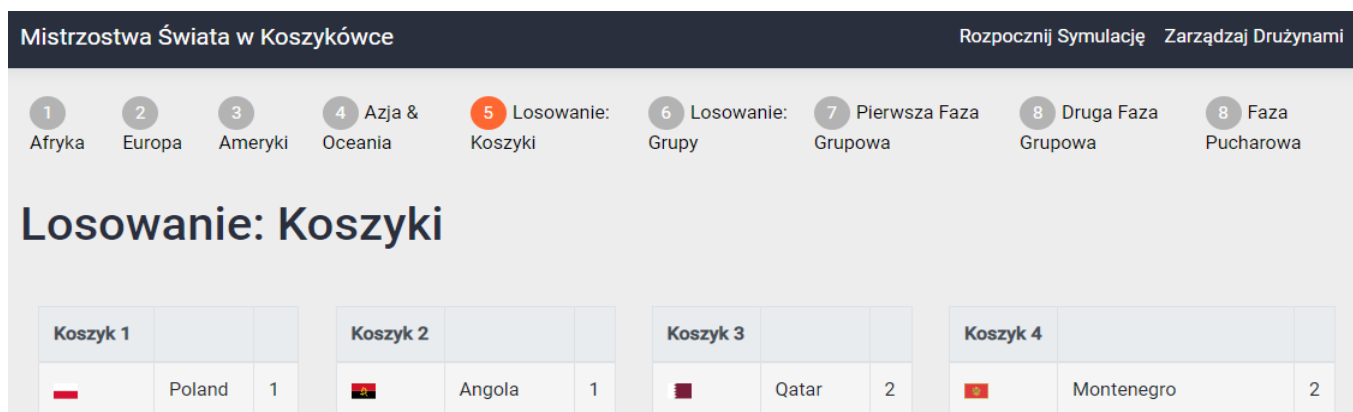
Ponadto po wybraniu drużyn zostanie odblokowany przycisk przekierowujący do wyboru drużyn z następnej Konfederacji. Aby przejść dalej wystarczy kliknąć przycisk „Następny krok”, który zawsze znajduje się na dole ekranu.

W wypadku chęci zmiany poprzedniej Konfederacji przycisk "Poprzedni Krok" pozwolił się cofnąć.

<input checked="" type="checkbox"/>		Congo	3
<a href="#">← Poprzedni krok</a>		<a href="#">Następny krok →</a>	

Rysunek 55

Po wybraniu drużyn z ostatniej konfederacji, używamy przycisku „Następny krok”, aby przenieść się do ekranu losowania drużyn do koszyków.

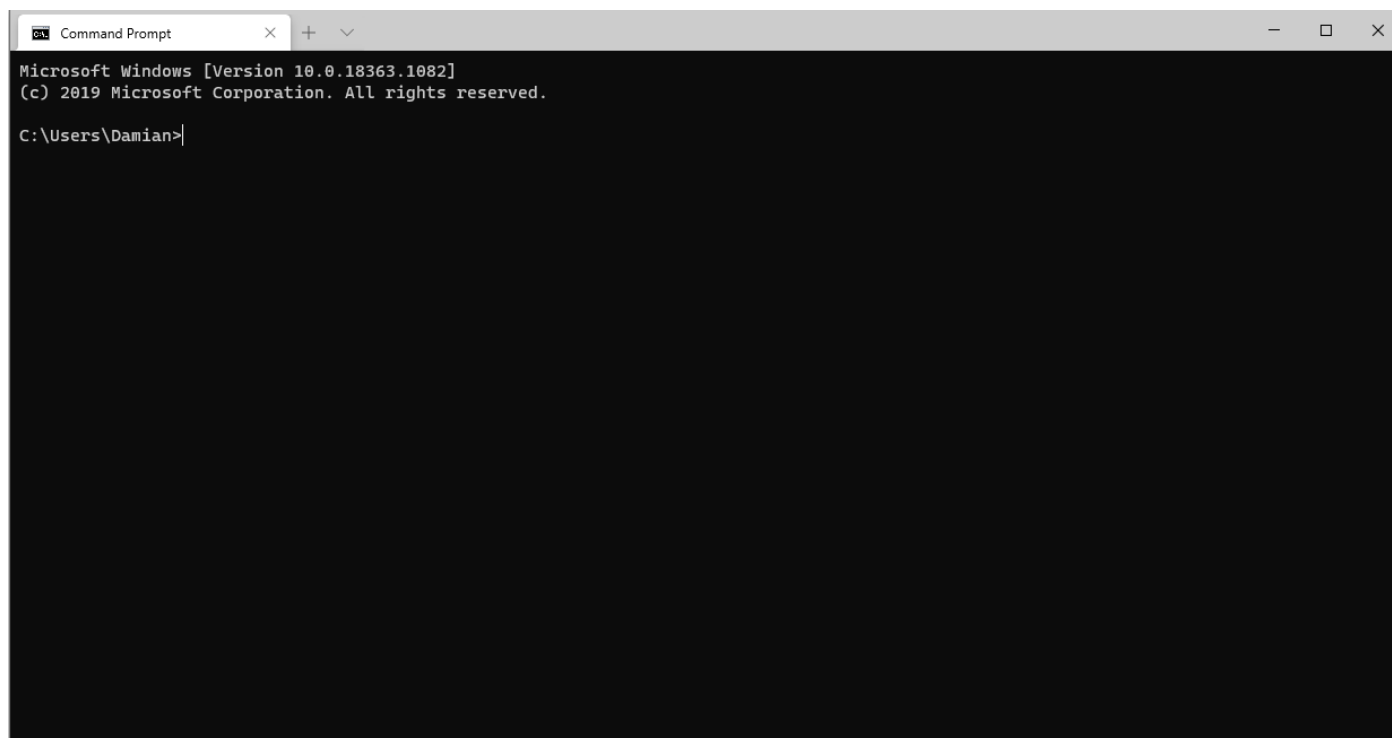


Rysunek 56

Od tego momentu użytkownik musi tylko naciskać przycisk „Następny krok” i oglądać wyniki poszczególnych kroków.

## 4.2 Kompilacja aplikacji

W celu kompilacji programu i otrzymania pliku wykonywalnego .exe należy otworzyć konsolę systemu Windows (CMD).



Rysunek 57

Następnie przechodzimy do folderu *BasketballWorldCup* skopiowanego z płytki CD. Używamy do tego komendy *cd*. Po przejściu do odpowiedniego folderu używamy do kompilacji polecenia:

**dotnet publish -c Debug -r win10-x64 /p:PublishSingleFile=true**

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Damian>cd C:\Users\Damian\source\repos\BasketballWorldCup

C:\Users\Damian\source\repos\BasketballWorldCup>dotnet publish -c Debug -r win10-x64 /p:PublishSingleFile=true|
```

Rysunek 58

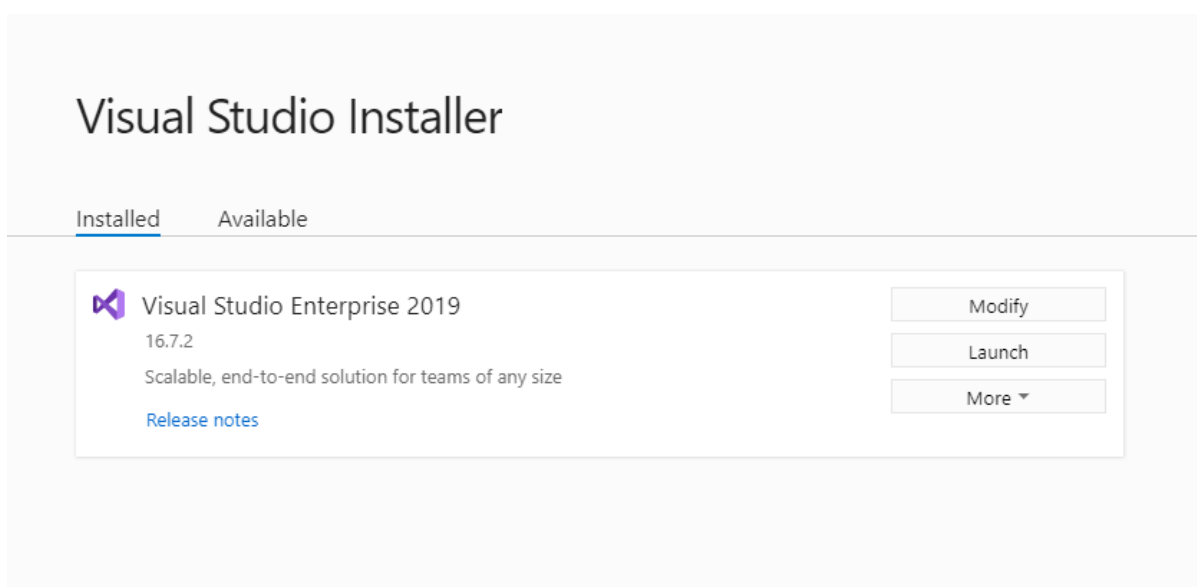
Spowoduje to utworzenie pliku wykonywalnego w ścieżce:

**`.\BasketballWorldCup\BasketballWorldCup\bin\Debug\netcoreapp3.1\win10-x64\publish`**

### 4.3 Przygotowanie środowiska

Przed pierwszym uruchomieniem aplikacji należy przygotować środowisko dla niej. W tym celu niezbędne jest na początku przeprowadzenie kilku czynności umożliwiających działanie aplikacji. Na początek wymagane jest zainstalowanie programu SQL Server Management Studio [9], który jest dostępny na stronie producenta i który posiada darmową wersję wystarczającą do działania aplikacji. Instalacja nie wymaga żadnych nadmiarowych konfiguracji, wystarczy cały czas naciskać przycisk „Dalej”. SSMS pozwoli na ustawienie bazy danych, która przetrzymuje informacje o drużynach pomiędzy uruchomieniami aplikacji.

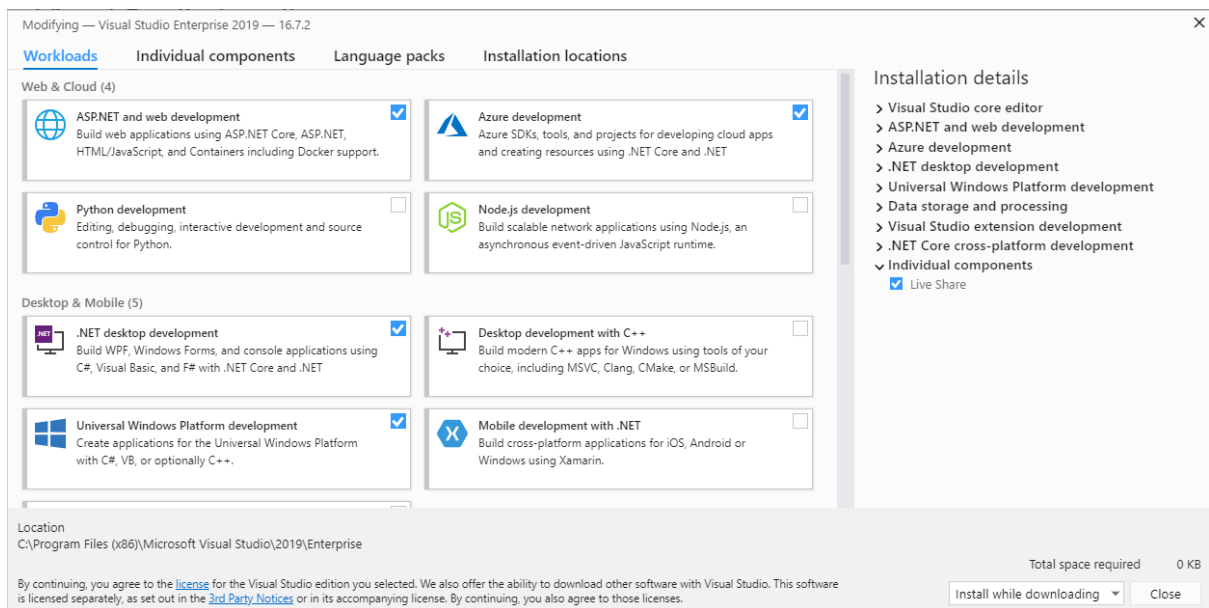
Gdy program SSMS jest już gotowy do użycia w następnym kroku uruchamiamy program Visual Studio Installer w celu doinstalowania dodatkowych funkcjonalności pozwalających na działanie aplikacji napisanej w technologii ASP.NET Core.



Rysunek 59

Po uruchomieniu zobaczymy ekran z dostępnymi instancjami programu Visual Studio. Naciskamy na przycisk Modyfikuj (*ang. Modify*).



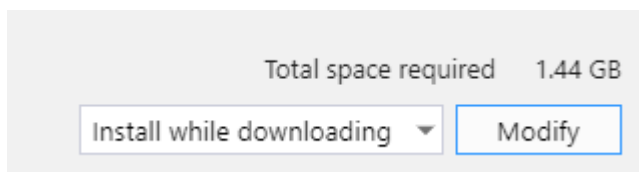


Rysunek 60

W panelu wyboru dodatkowych funkcjonalności upewniamy się, że mamy zaznaczone opcje:

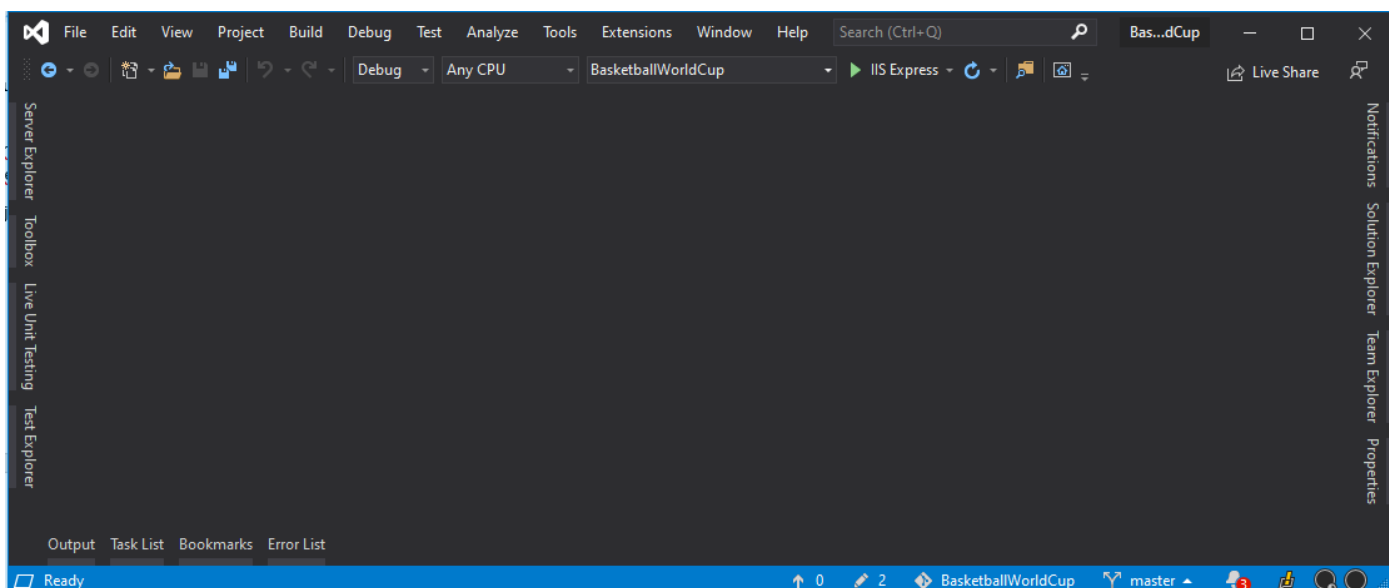
- ASP.NET and web development
- Data storage and processing
- .NET Core cross-platform development

Jeśli któraś z opcji jest odznaczona należy ją zaznaczyć i zatwierdzić zmiany przyciskiem Modyfikuj (ang. *Modify*).



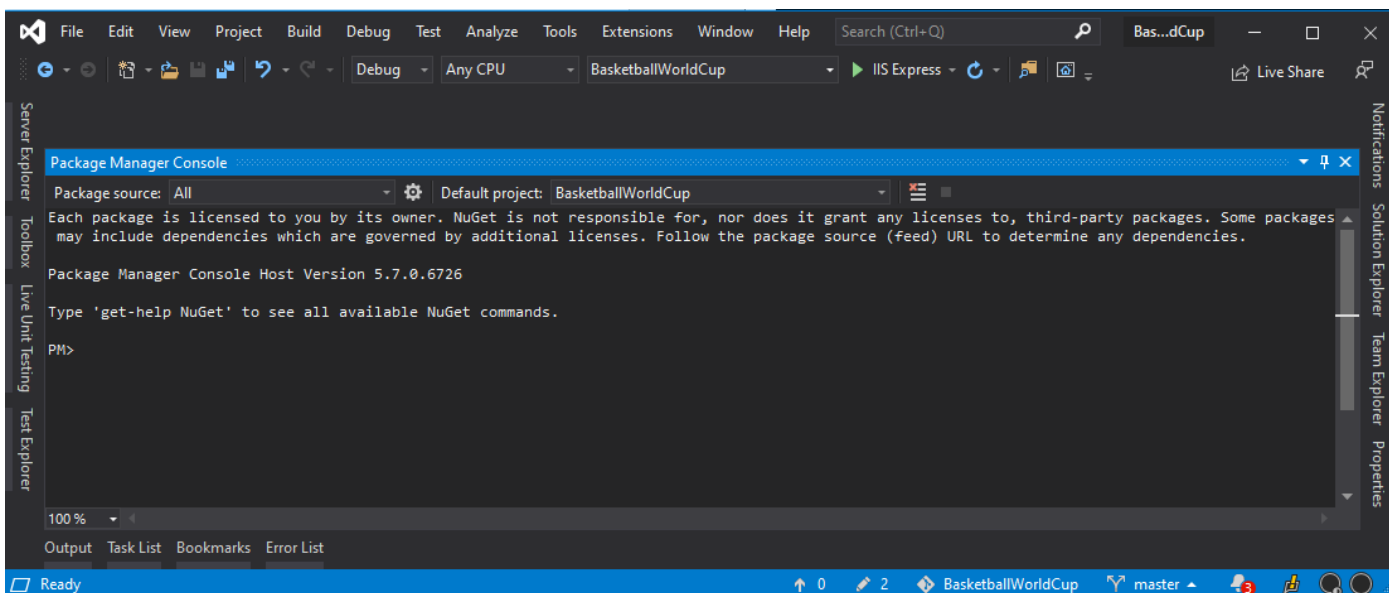
Rysunek 61

Po zainstalowaniu SSMS i VS2019 można przystąpić do dalszej konfiguracji bazy danych. Odbywa się to poprzez wczytanie pliku solucji o nazwie *BasketballWorldCup.sln* w programie Visual Studio.

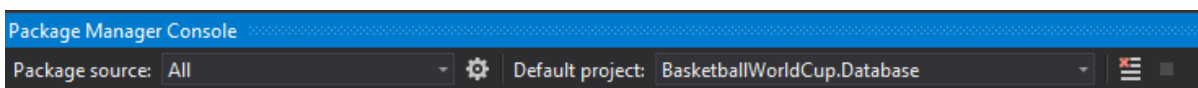


Rysunek 62

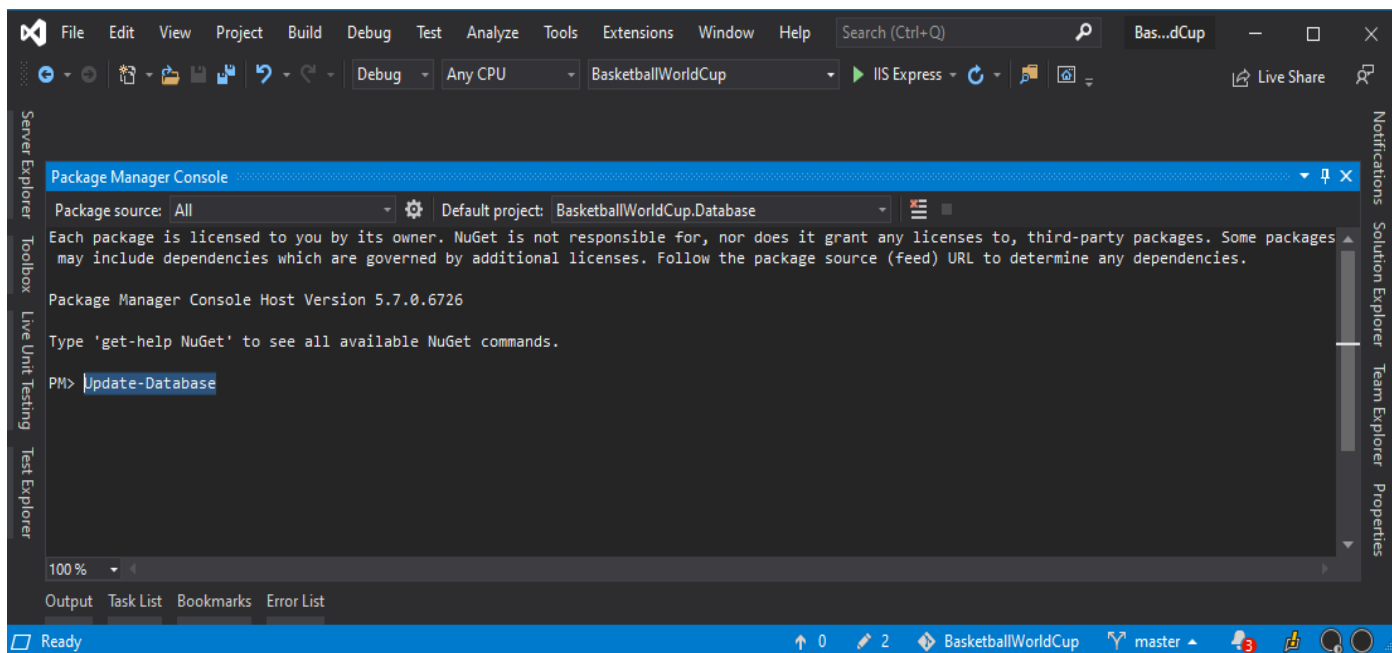
Po wczytaniu pliku należy rozwinąć zakładkę *Package Manager Console* i przełączyć opcję Domyślne projekty (*ang. Default projects*) na *BasketballWorldCup.Database*. Następnie wpisujemy komendę *Update-Database*, która pozwoli automatycznie utworzyć bazę danych w programie SSMS.



Rysunek 63



Rysunek 64



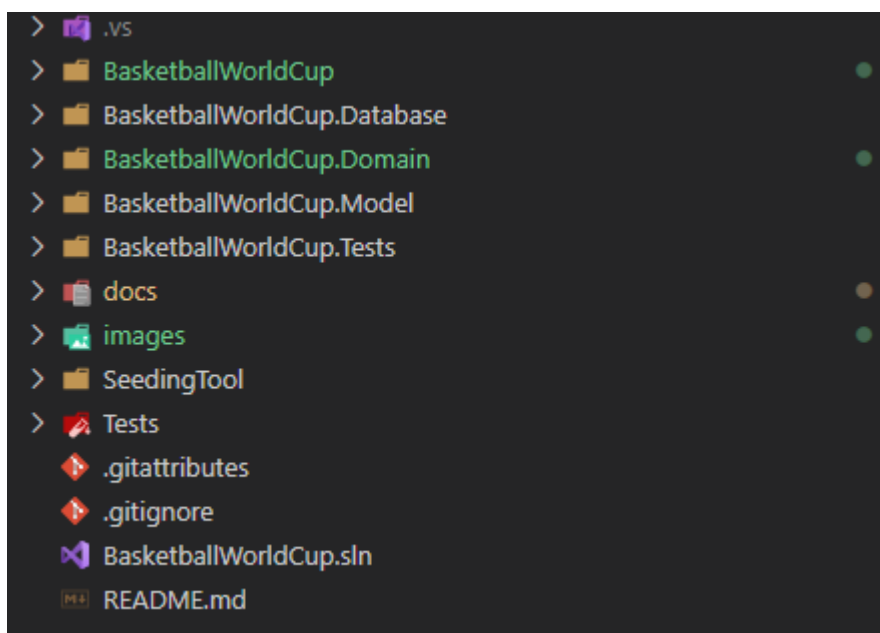
Rysunek 65

## 5 Bibliografia

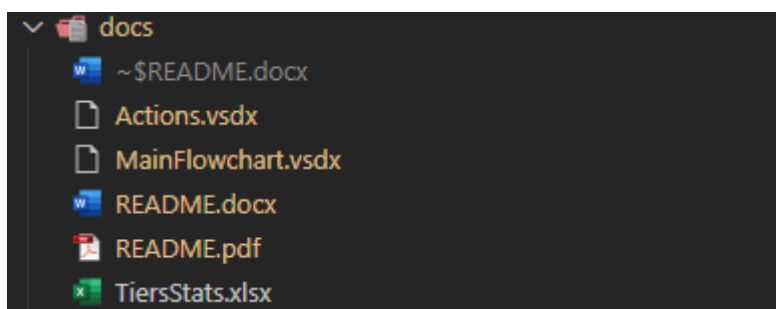
- [1] FIBA. COMPETITION SYSTEM. *FIBA Basketball*. [Online] 2019.  
<http://www.fiba.basketball/basketballworldcup/2019/competition-system>.
- [2] 2019 FIBA Basketball World Cup. *Wikipedia*. [Online] <https://wikipedia>.
- [3] Gof. M. *PRZEPISY GRY W KOSZYKÓWKĘ*. Szczecin, Poland : brak nazwiska, 2013.
- [4] SphinxKnight. JavaScript. [Online] MDN, 2019.  
<https://developer.mozilla.org/pl/docs/Web/JavaScript>.
- [5] Google. Angular. [Online] Google, 2010. <https://angular.io/>.
- [6] Microsoft. ASP.NET. [Online] Microsoft. <https://dotnet.microsoft.com/apps/aspnet>.
- [7] —. Download SQL Server Management Studio (SSMS). *SQL Docs*. [Online] Microsoft, 22 07 2020.  
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>.
- [8] —. Visual Studio 2019. *Visual Studio* . [Online] Microsoft.  
<https://visualstudio.microsoft.com/pl/vs/>.
- [9] Wikipedai. Architektura trójwarstwowa. *Wikipedia*. [Online]  
[https://pl.wikipedia.org/wiki/Architektura\\_tr%C3%B3jwarstwowa?oldformat=true](https://pl.wikipedia.org/wiki/Architektura_tr%C3%B3jwarstwowa?oldformat=true).
- [10] Wikipedia. Plain Old CLR Object. *Wikipedia*. [Online]  
[https://pl.wikipedia.org/wiki/Plain\\_Old\\_CLR\\_Object?oldformat=true](https://pl.wikipedia.org/wiki/Plain_Old_CLR_Object?oldformat=true).
- [11] —. Dogrywka - koszykówka. *Wikipedia*. [Online]  
[https://pl.wikipedia.org/wiki/Dogrywka\\_\(koszyk%C3%B3wka\)?oldformat=true](https://pl.wikipedia.org/wiki/Dogrywka_(koszyk%C3%B3wka)?oldformat=true).
- [12] —. A tour of the C# language. *Microsoft*. [Online]  
<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [13] J. Greene, A. Stellman, *Head First C#, 3rd Edition*, Helion, Gliwice, 2014

## 6 Zawartość płytki CD

Płyta CD zawiera w sobie projekt programistyczny, wraz z dokumentacją. W głównym folderze znajdują się pliki do repozytorium programu kontroli wersji *git* (pliki *.gitattributes*, *.gitignore*), plik solucji programu VS (*BasketballWorldCup.sln*). Główny folder zawiera foldery odnoszące się również do projektu programistycznego (foldery *BasketballWorldCup*, *BasketballWorldCup.Database*, *BasketballWorldCup.Domain*, *BasketballWorldCup.Model*, *BasketballWorldCup.Tests*). Folder *docs* posiada plik PDF dokumentacji projektu (plik *README.pdf*) wraz z jego wersją edytowalną (*README.docx*). Oprócz tego w tym folderze można znaleźć pliki odnoszące się diagramów i wykresów użytych w dokumentacji (pliki *Actions.vsd*, *MainFlowchart.vsd*, *TierStats.xlsx*). Natomiast folder *images* posiada obrazy użyte w dokumentacji.



Rysunek 66



Rysunek 67

### Oświadczenie

Oświadczam, że niniejszy projekt i sprawozdanie wykonałem osobiście i wyrażam zgodę na jego wykorzystanie dla celów dydaktycznych. Równocześnie, oświadczam, że niniejszy projekt można uruchomić na jednym z komputerów w Laboratoriach Wydziału Matematyczno-Przyrodniczego UKSW.

*Damian Uboński*  
Podpis

*Załącznik Nr 1 do Decyzji Nr 10/2020 z dnia 2 czerwca 2020 r. Prorektora ds. Studenckich i Kształcenia*

DAMIAN UBOŃSKI  
Imię i nazwisko studenta:  
104216  
Nr albumu  
INFORMATYKA  
Nazwa studiów (kierunek, poziom, forma, profil)  
PROJEKT INDYWIDUALNY  
Nazwa przedmiotu

### OŚWIADCZENIE

1. Świadomy(a) odpowiedzialności prawnej oświadczam, że przesłana praca pisemna pod nazwą

*SYMULACJA MISTRZOSTW ŚWIATA W PIŁKĘ KOSZYKOWĄ*

została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

2. Oświadczam, że jestem autorem/współautorem przesłanej pracy i przysługują mi w zakresie objętym tym oświadczeniem osobiste prawa autorskie do tej pracy

*Damian Uboński*  
podpis studenta