

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[wsei-csharp201 / cs-lab-Monitorowanie-pozycji-zawodnikow](#) Public[Watch](#) **1**[Fork](#) **2**[Star](#) **0**[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)[main](#)[1 branch](#)[0 tags](#)[Go to file](#)[Add file](#)[Code](#)[About](#)[Ćwiczenie na wykorzystanie event-ów w C#](#)[Readme](#)[0 stars](#)[1 watching](#)[2 forks](#)[Releases](#)[No releases published](#)[Packages](#)[No packages published](#)[README.md](#)

Monitorowanie pozycji zawodników

- Autor: *Krzysztof Molenda*
- Wersja: 2020-12-05

Problem

Tworzysz oprogramowanie do monitorowania przemieszczania się K zawodników (np. $K=5$) po wirtualnym boisku o rozmiarze $N \times M$ (np. 500×1000).

W klasie `Location` rejestrujesz aktualny czas oraz położenie zawodnika na boisku.

Zawodnika opisuje klasa `Player` - jego `id` oraz aktualne położenie. Zawodnik przesuwa się po boisku w różnych kierunkach (o rozsądne odległości - raz szybciej, innym razem wolniej). Zawodnik nie opuszcza boiska, czasami nie przemieszcza się. Fakt przesunięcia się zawodnika realizuje wywołanie metody `ChangeLocation` zaimplementowanej w klasie `Player`.

W klasie `Tracker` zapamiętujesz aktualne pozycje zawodników (dowolnej liczby). Fakt ten realizuje metoda `OnLocationChange` obsługująca zdarzenia zmiany miejsca zawodnika (zawodnicy zgłaszają zmiany swoich pozycji).

Zadanie

Zaimplementuj ten prosty system komunikowania się zawodników z monitorem śledzącym ich ruchy z wykorzystaniem `event`-ów (w projekcie typu *Class Library*).

Napisz prostą aplikację konsolową - symulator weryfikujący poprawność:

- utwórz K zawodników,
- zawodnicy rejestrują się w monitorze śledzącym ich ruchy,
- zmuś zawodników do losowego poruszania się lub nie poruszania się (zmiany pozycji rozsądnie, bez odległych skoków) - zmiany pozycji zawodników uruchom w oddzielnych wątkach.

Po zakończeniu symulacji określ, który z zawodników pokonał największą odległość.

Zadanie dodatkowe

Zrealizuj opakowanie wizualne - w formie aplikacji okienkowej. Aplikacja powinna wyświetlać ruchy zawodników na wirtualnej planszy. Po zakończeniu może wyświetlać ścieżkę danego zawodnika. Możesz również zrealizować inne własne pomysły.

Przykładowe wykonanie zadań znajdziesz w folderze `/example-bin` - dwa pliki `.exe`

odpowiednio dla aplikacji konsolowej i desktopowej, bazujące na kodzie wspólnej biblioteki (wymagane środowisko .Net5).

Co dalej ...

Jeśli wykonałeś powyższe proste zadanie, zrozumiałeś, jak działa komunikacja między obiektami z wykorzystaniem zdarzeń i ich obsługi.

W C#4 do .Net Framework dodano *Location API*: `System.Device.Location`.

UWAGA: Nie ma tego API w .Net Standard ani .Net Core.

W tej przestrzeni nazw zdefiniowano podobną infrastrukturę do tej z zadania.

- Zapoznaj się z typami zdefiniowanymi w `System.Device.Location`
- Rzuć okiem na projekty przykładowe, skompiluj, uruchom, zmodyfikuj:
 - <https://github.com/microsoft/Windows-universal-samples/tree/master/Samples/Geolocation>
 - <https://github.com/Microsoft/Windows-appsample-lunch-scheduler>
 - <https://github.com/microsoft/windows-appsample-trafficapp/>

