Projekt Warriors

Autorzy:
Maciej Kowalczyk, Radosław Mikołajczyk, Arkadiusz Pasek

Symulacja dyskretna systemów złożonych

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Spis treści

1 Wstęp

1.1 Opis problemu

1.1.1 Cel projektu

Celem bieżącego projektu będzie przedstawienie symulacji starć historycznych formacji zbrojnych na podstawie zebranych informacji historycznych. Symulowanie bitew staje się coraz bardziej popularne, przeważająco w dziedzinach rozrywki, kinematografii oraz gier komputerowych. Zwracanie się w kierunku symulacji wojennych w poprzednio wymienionych dziedzinach jest spowodowane pragnieniem osiągnięcia większego realizmu prezentowanych zdarzeń.

1.1.2 Zastosowanie realne w świecie

Głównym problemem odtwarzania przebiegu bitew przeszłości jest wymóg wymuszenia dużej dokładności oraz ciągłości przebiegu starcia, dużo większej niż możemy osiągnąć z czytania źródeł historycznych. Możemy z nich wydobyć informacje na przykład na temat położenia pola walki, rozstawienia wojsk, liczebności, wyszkolenia oraz siły sił zbrojnych poszczególnych stron konfliktu. Często jest to wystarczająca ilość informacji pozwalająca wyprowadzić wnioski dotyczące głównych założeń przebiegu oraz potencjalnych wyników bitwy. Niestety, tworząc scenę walki w filmie lub realizując grę historyczno-taktyczną wymagamy znacznie większej ciągłości. Oferując widzowi lub graczowi bitwę nie wystarczą założenia bitwy, musi on otrzymać ruch sił konfliktu w kolejnych chwilach przebiegu, obserwując bitwę od chwili jej rozpoczęcia. Przykładowo, potrzebujemy wiedzieć czy zwycięska strona zawdzięcza lepszy wynik w walce impaktowi początkowej szarży czy wręcz przeciwnie, desperacko broniła się we wstępnych fazach przebiegu konfliktu stopniowo zdobywając przewagę. Jeśli zatem będziemy w stanie odpowiedzieć na takie pytania symulując przebieg bitwy, będziemy w stanie zaproponować realizm końcowemu odbiorcy.

1.1.3 Temat projektu

Tematem przewodnim tego projektu jest stworzenie silnika pozwalającego na symulację obrazującą przebieg bitwy. Tworzone symulacje chcielibyśmy oprzeć o trzy ważne bitwy w Europie:

- Bitwa pod Chocimiem - 1509
- Bitwa pod Chocimiem - 1621
- Bitwa pod Chocimiem - 1673

Uzasadnieniu wyboru tych bitew poświęcony jest następny rozdział. Najważniejszymi punktami otrzymanych symulacji będzie możliwość odtworzenia przebiegu bitwy, jej rozwoju w kolejnych chwilach oraz możliwość wyciągnięcia wniosków.

1.2 Potencjalne rozwiązanie

Po zapoznaniu się ze źródłami na stronach z bazami danych o poszczególnych bitwach zdecydowaliśmy, że nasze rozwiązania będą się opierać na danych pochodzących ze źródeł historycznych, które w sposób szczegółowy opisują bitwy. Na podstawie tych danych, po ich analizie będziemy starać się zaimplementować algorytmy, które będą oddawać przebieg oraz rezultat starć pod Chocimiem na przestrzeni dwóch wieków. Oprócz symulacji bitew wprowadzimy również dane uśrednione w celu pokazania jak w poszczególnych latach zmieniała się armia, taktyka oraz sekwencja potyczek.

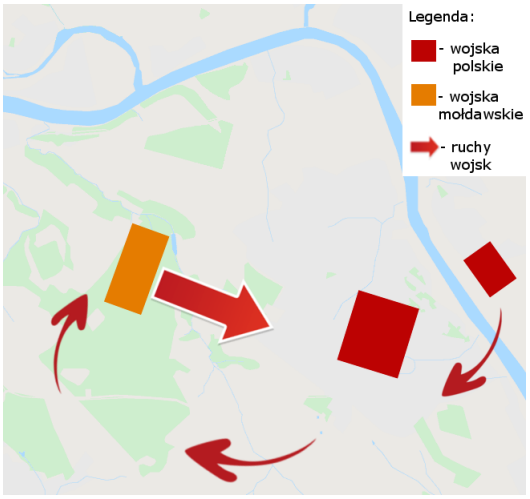
2 Historia

Starcia pod Chocimiem miały znaczący impakt na przebieg wydarzeń w szesnastowiecznej i siedemnastowiecznej Europie. Przyjrzyjmy się jak wyglądały poszczególne bitwy w rzeczywistości.

2.1 1509

Bitwa miała miejsce 4 października 1509 roku. Była starciem kończącym kilkuletnią wojnę polsko-mołdawską. Wojska polskie pod dowództwem hetmana Mikołaja Kamienickiego po nieudanym wielomiesięcznym oblężeniu stolicy mołdawskiej były zmuszone do odwrotu. Podczas przeprawy przez Dniestr w okolicach Chocimia wojska mołdawskie zdecydowały się wydać Polakom bitwę. Pomimo efektu zaskoczenia Polacy, dzięki manewrowi jazdy, która niepostrzeżenie odłączyła się od reszty sił i obeszła wzgórze, z którego przeprowadzony był atak od zachodniej strony i uderzyła na wrogie wojska od tyłu, pokonali połączone wojska mołdawskie i tureckie.
Skala bitwy:

- ok. 20 tys. żołnierzy polskich (w tym tzw. pospolite ruszenie),
- prawie cała ówczesna armia mołdawska oraz 2 tys. żołnierzy tureckich.



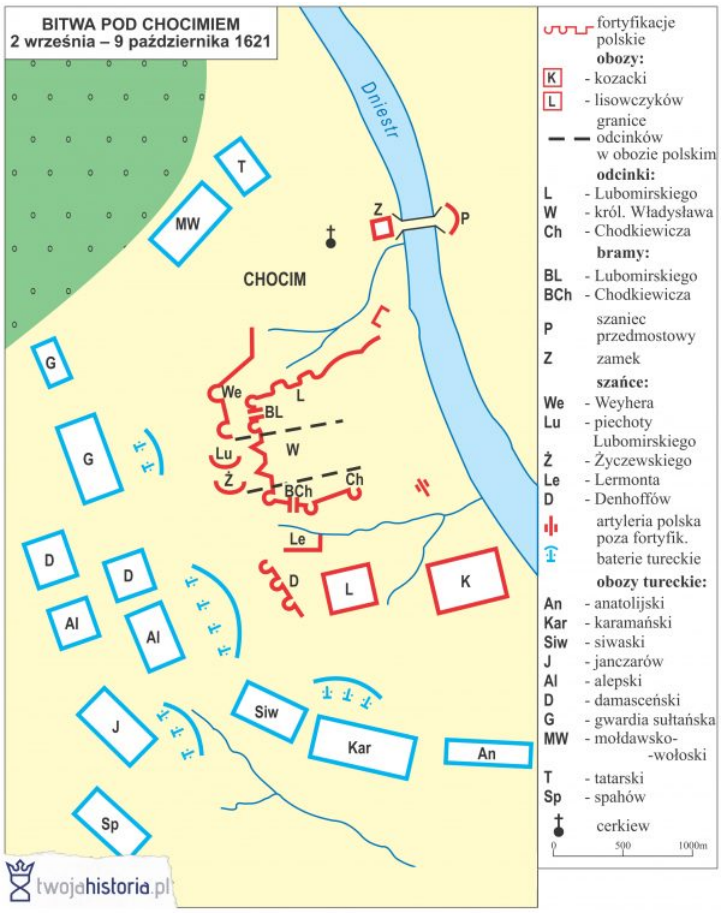
Rysunek 1: Wizualizacja ruchu jednostek w bitwie pod Chocimiem w 1509r.

2.2 1621

Starcie odbywało się od 2 września do 9 października 1621 roku. Bitwa była stoczona między wojskami Rzeczypospolitej i Turcji podczas wojny polsko - tureckiej. Na złe stosunki wzajemne miały wpływ w głównej mierze prywatne wyprawy magnatów na Mołdawię, ekspedycje kozaków na wybrzeża Morza Czarnego, najazdy tatarskie na Podole i Ukrainę oraz wojna trzydziestoletnia. Pierwsze ataki tureckie nie przyniosły żadnego sukcesu - wojskom polskim udało się odeprzeć szturmy 2 i 3 września jak również kolejne uderzenie pod osłoną ognia artyleryjskiego. Po tych klęskach Turcja otoczyła obóz polski odcinając im dostawy zapasów. Morale polskiego wojska spadało, zaczynało brakować żywności, konie zaczynały umierać, a żołnierze chorowali. Morale wojska tureckiego również zaczynały słabnąć ze względu nie przyswojonych do jesiennej pogody Europy Środkowo - Wschodniej. Po wielu atakach na obóz polski, które zostały odparte, sułtan obawiający się nowych sił polsko-litewskich zdecydował o przyspieszeniu rokowań pokojowych. Rozpoczęły się one 28 września a ich wynikiem był pokój zawarty 11 dni później.

Straty:

- polskie: 2 tysiące poległych żołnierzy, 3 tysiące zmarłych od głodu i chorób oraz 2.7 tysiąca dezerterów. Kozacy stracili 6.5 tysiąca ludzi.
- tureckie: ok. 40 tysięcy żołnierzy, w głównej mierze zmarłych podczas szturmów na obozy polskie i obozy kozaków.

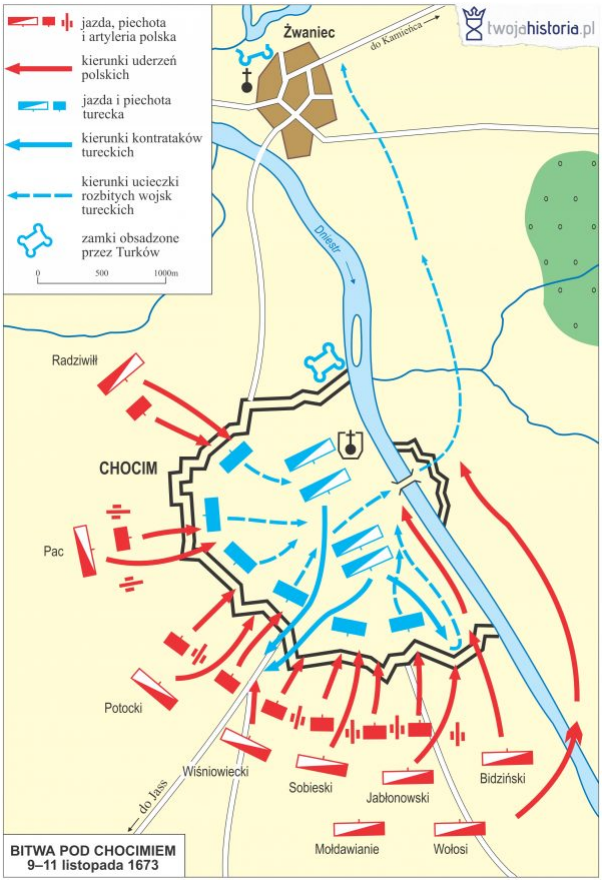


Rysunek 2: Wizualizacja rozmieszczenia jednostek podczas bitwy pod Chocimiem w 1621r.

2.3 1673

Bitwa stoczona 9 - 11 listopada 1673 między wojskami Rzeczypospolitej i Turcji w ramach wojny polsko-tureckiej. Hetman wielki koronny Jan Sobieski na czele polskiej armii udał się pod Chocim, w którym stacjonowała armia turecka. Obóz turecki był ostrzeliwany przez armaty polskie. Rankiem 11 listopada polska natarła na szeregi tureckie. Obóz usytuowany na prawym brzegu Dniestru został zdobyty.
Skala bitwy:

- 52-tysiące wojsk polskich, w połowie piechoty, w połowie jazdy
- 30-tysięczna armia turecka



Rysunek 3: Ruch wojsk w trakcie bitwy pod Chocimiem w 1673r.

3 Potencjalne rozwiązania problemu i model implementacji

3.1 Opis i cel modelu symulacji

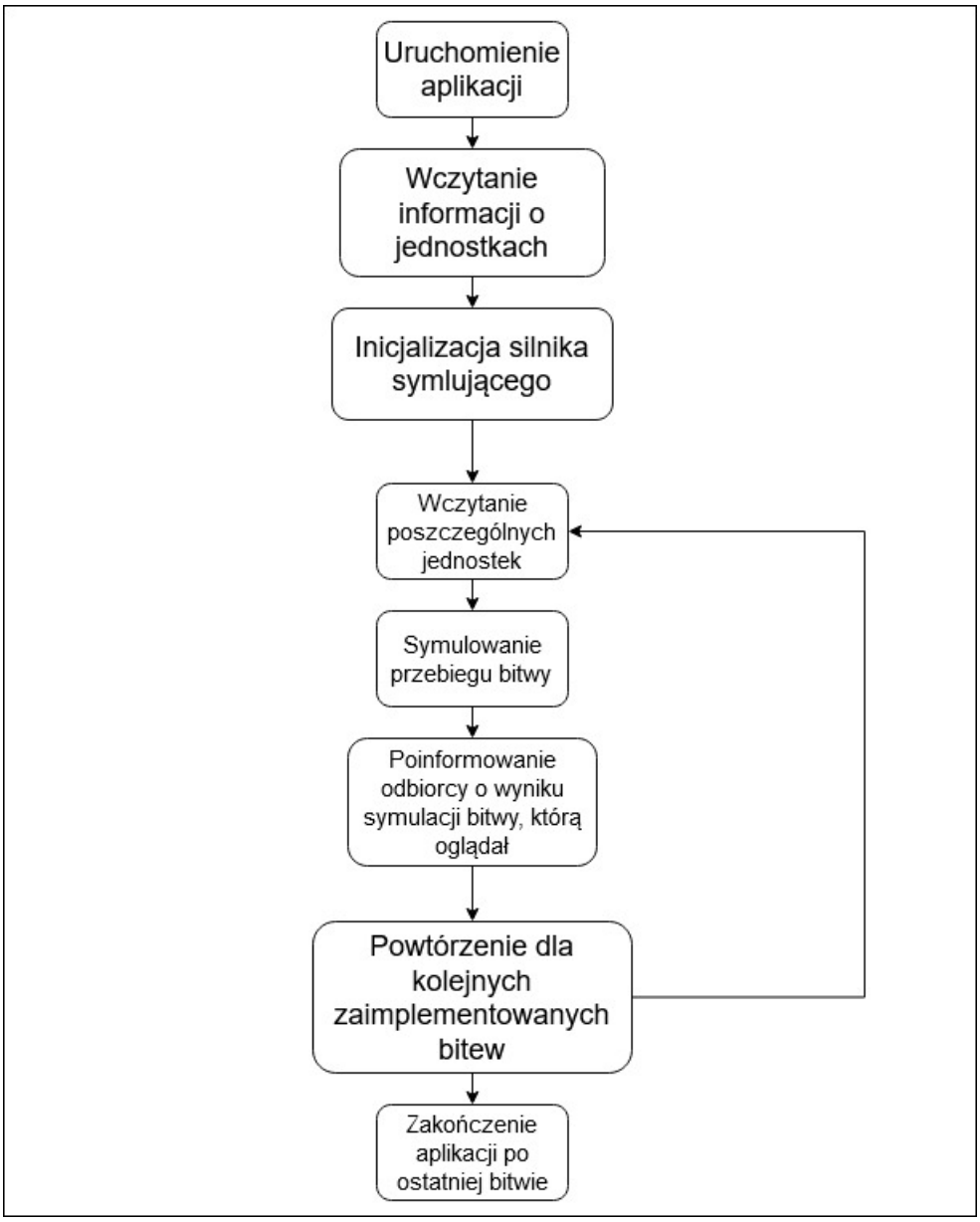
Naszym celem jest utworzenie silnika symulacji w wybranym języku programowania (python) oraz z wykorzystaniem dostępnych bibliotek możliwie ułatwiających osiągnięcie tego zadania (pygame).

Wspominany silnik musi być w stanie zwizualizować starcia wojsk podczas bitw, tzn. być w stanie rozpoznać różne rodzaje jednostek i oddziałów, ukazać je na mapie, implementować ich poruszanie się oraz starcie z oddziałem wroga.

Po przetestowaniu różnych rozwiązań na możliwe praktyczne pokazanie bitwy, naszym wyborem na spełnienie tych założeń, w celu zagwarantowania przejrzystości graficznej wizualizacji starcia oraz zapewnieniem, że aplikacja nie przytłoczy odbiorcy końcowego chaosem poruszających się po polu bitwy masy jednostek, jest ukazywanie biorących udział jednostek jako oddziałów, których parametry są obliczane na podstawie jednostek, z których taki oddział się składa.

Kolejnym ważnym elementem jest zaimplementowanie logiki oddziałów. Nie jesteśmy w stanie ręcznie zaprogramować każdej kolejnej akcji każdego oddziału, zatem musimy napisać uproszczony system sztucznej inteligencji, który będzie decydował o swojej następnej akcji i jeśli będziemy w stanie wykorzystać szczegółowe informacje o przebiegu bitwy ze źródeł historycznych wymuszać pewne zachowania oddziału (na przykład, czerpiąc informację ze źródeł, zamiast pozwolić oddziałowi ruszyć do bitwy z innym oddziałem, wymuszamy na nim przemieszczenie się w pewne miejsce skąd będzie miał większą skuteczność i zdobędzie większe prowadzenie dla swojej strony w bitwie).

3.2 Zaplanowany cykl aplikacji



Rysunek 4: Diagram przedstawiający działanie aplikacji.

3.3 Opis działania silnika

Tworzona przez nas aplikacja w trakcie jej działania musi wykonać kolejno poszczególne zadania:

- Otrzymać dane o typach jednostek i przełożyć je na praktyczną implementację w silniku (tzn. określić różne parametry, które są kluczowe w trakcie symulowania starć zbrojnych: wytrzymałości, obrony, prędkości poruszania, zdolności do atakowania, itd.)
- Otrzymać informację o rozmieszczeniu i rodzaju poszczególnych jednostek dla każdej kolejnej bitwy i utworzyć z nich oddziały, które będą poddane symulacji
- Wczytać mapę, która pozwoli lepiej zrozumieć odbiorcy przebieg bitwy
- Rozpocząć symulację dla kolejnych bitew
  - Wczytać oddziały i ich pozycję
  - Na podstawie logiki silnika lub zaimplementowanych specjalnych zadań lub tras poruszania się, nadać im działania
  - Ukazywać kolejne chwile przebiegu bitwy
  - Po dojściu do punktu końcowego, wypisać jak bitwa się skończyła, tzn. kto wygrał, jakie oddziały oraz ile żołnierzy w tych oddziałach przetrwało

4 Realizacja technologiczna

4.1 Odziały i jednostki

Aby oddać możliwie realistycznie przebieg bitwy, z jednoczesnym zachowaniem przejrzystości podczas oglądania symulacji, zdecydowaliśmy się zaimplementować wojska w postaci oddziałów, które składają się z jakiegoś typu jednostki(te jednostki mają swoje unikalne parametry: żywotność, wytrzymałość, prędkość, obronę, atak, pancerz), ilości takich jednostek oraz tego gdzie taki oddział powinien w początkowej fazie bitwy zostać ulokowany, co zostało zrealizowane poprzez nadanie mu współrzędnych.

```
if pattern == 1:
    return [
        # Jednostki polskie
        [
            Infantry(600, 300, 5000),
            SpearMan(630, 230, 5000),
            Hussar(900, 275, 5000)
        ],
        # Jednostki tatarskie
        [
            TatarInfantry(475, 170, 5000),
            TatarInfantry(450, 210, 4000),
            TatarCavalry(425, 250, 5000),
            SpearMan(400, 290, 4000)
        ]
    ]
```

Rysunek 5: Kod ilustrujący tablice, na podstawie której oddziały są tworzone na początku pierwszej bitwy

4.2 Atak i starcia oddziałów

Następnie z takiej tablicy tworzone są odziały jednostek i rozpoczynana jest walka. Walka odbywa się na poziomie oddziałów, tzn. oddział porusza się jako całość, jako całość wybiera, do którego przeciwnika powinien się udać i atakuje siłą obliczoną na podstawie tego jakie jednostki zawiera i ile ich posiada. Walka jednego oddziału z drugim następuje wtedy, gdy oddziały zdołały do siebie dotrzeć.

```
def is_inside(l_top, size, x, y):
    return (l_top[0] < x < l_top[0] + size and
            l_top[1] < y < l_top[1] + size)

def does_overlap(first, second):
    l1 = [first.x, first.y]
    l2 = [second.x, second.y]

    return (is_inside(l1, first.get_size(), l2[0], l2[1])
        or is_inside(l1, first.get_size(), l2[0] + second.get_size(), l2[1])
        or is_inside(l1, first.get_size(), l2[0], l2[1] + second.get_size())
        or is_inside(l1, first.get_size(), l2[0] + second.get_size(), l2[1] + second.get_size())

        or is_inside(l2, second.get_size(), l1[0], l1[1])
        or is_inside(l2, second.get_size(), l1[0] + first.get_size(), l1[1])
        or is_inside(l2, second.get_size(), l1[0], l1[1] + first.get_size())
        or is_inside(l2, second.get_size(), l1[0] + first.get_size(), l1[1] + first.get_size())
    )
```

Rysunek 6: Kod ilustrujący sprawdzenie czy oddziały dotarły do siebie i czy są w stanie zaatakować się nawzajem

Ważną metodą było również odnajdywanie najbliższego oddziału (wprowadzenie systemu walki oddziałów zdecydowania pozwoliło ułatwić algorytm wyszukiwania najbliższego przeciwnika):

```
def find_closest(x, y, arr):
    currently_best = None
    currently_best_dist = 0
    for i in range(0,len(arr)):
        distance = get_distance(x,y,arr[i].x,arr[i].y)
        if(distance < currently_best_dist or currently_best_dist == 0):
            currently_best_dist = distance
            currently_best = arr[i]

    return [currently_best, currently_best_dist]
```

Rysunek 7: Kod zwracający najbliższy oddział przeciwnika i odległość od niego dla danego oddziału

4.3 Rysowanie oddziałów, a jego parametry

Aby osoba końcowa, oglądająca tę symulację mogła w czasie rzeczywistym obserwować zmiany zachodzące podczas starcia zbrojnego różnych oddziałów, postanowiliśmy dynamicznie zmniejszać wielkość oddziału (oddział w wyniku starcia traci swoje jednostki, więc od tego uzależniliśmy to jaką wielkość będzie posiadał po narysowaniu, uwzględniając również żywotność jednostek, które zawiera).

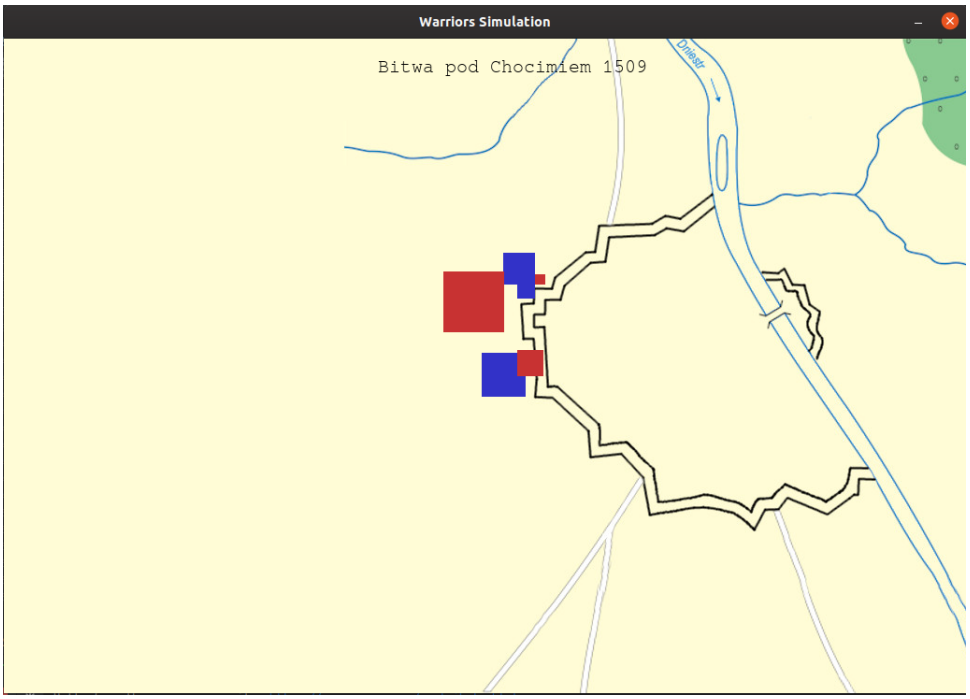
```
def get_size(self):
    return (self.number * self.hp) / 10000

def draw(self, screen, ally):
    color = (50, 50, 200) if not ally else (200, 50, 50)
    rect = pygame.draw.rect(screen, color, (self.x, self.y, self.get_size(), self.get_size()))
```

Rysunek 8: Kod ilustrujący obliczanie wielkości rysowania dla oddziału w zależności od jego parametrów

5 Wyniki symulacji

5.1 Bitwa 1509

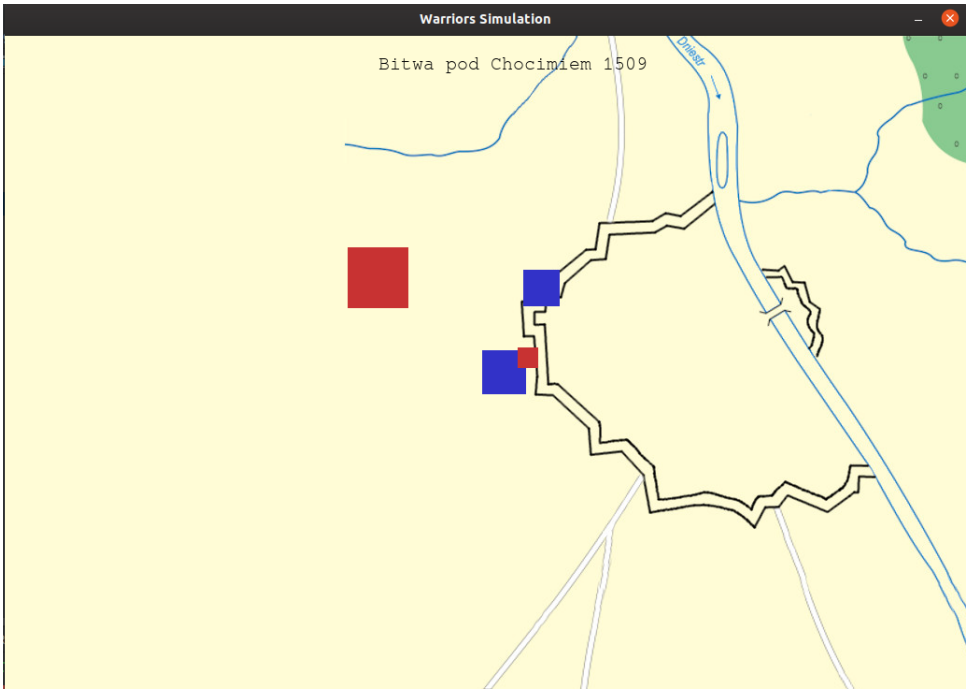
Po wprowadzeniu danych dla jednostek w pliku tekstowym, wprowadzone jednostki wojskowe, które zostały dobrane na podstawie źródeł historycznych rozpoczęły bitwę.



Rysunek 9: Symulacja bitwy pod Chocimiem 1509 r.

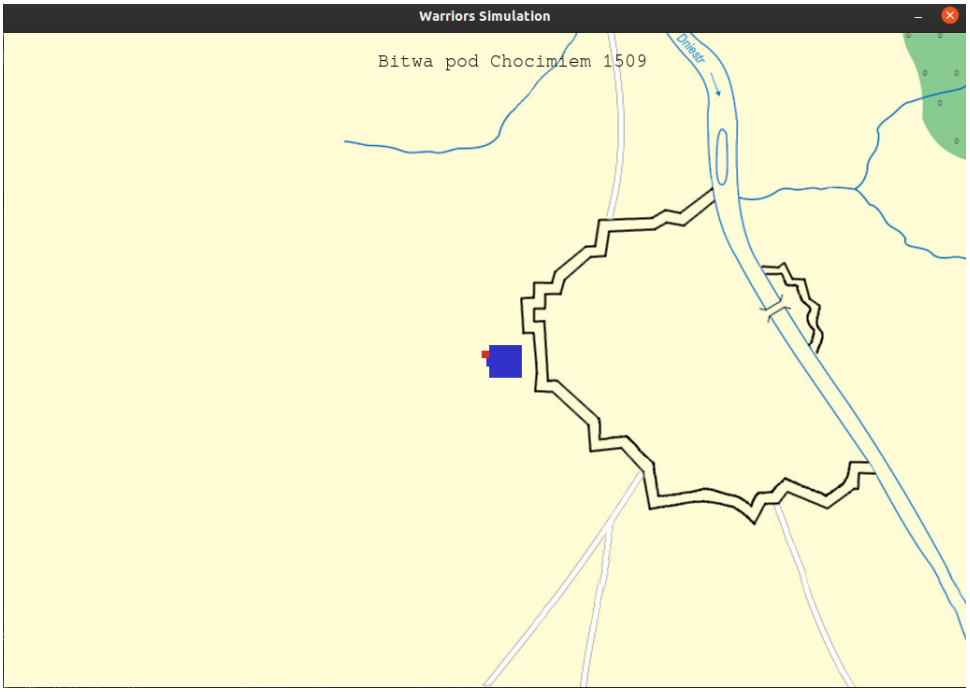
Bitwa od początku przebiegała zgodnie z wydarzeniami historycznymi, tzn. oddziały piechoty polskiej podjęły walkę z przeważającymi siłami wojsk tureckich, natomiast element zaskoczenia, którym były wojska polskiej husarii przesądził o wyniku starcia. Polska jazda zastosowała manewr odcinający odwrót jednostkom tureckim atakując z tyłu.

Podczas eksperymentów zarówno z ilością armii, jak i jej parametrami zauważyliśmy, że w zależności od tych parametrów losy bitwy mogły potoczyć się w zupełnie inny sposób. Podczas zmniejszenia parametru prędkości poruszania się husarii okazało się, że polska jazda nie dociera na czas na pole bitwy, co poskutkowało przegrupowaniem sił tureckich oraz odparciu ataku husarii.



Rysunek 10: Symulacja bitwy pod Chocimiem 1509 r - Zmieniona prędkość poruszania się Husarii.

Zniknął już tutaj element zaskoczenia i bitwę ostatecznie wygrały wojska tureckie.



Rysunek 11: Symulacja bitwy pod Chocimiem 1509 r. - Zwycięstwo tureckie

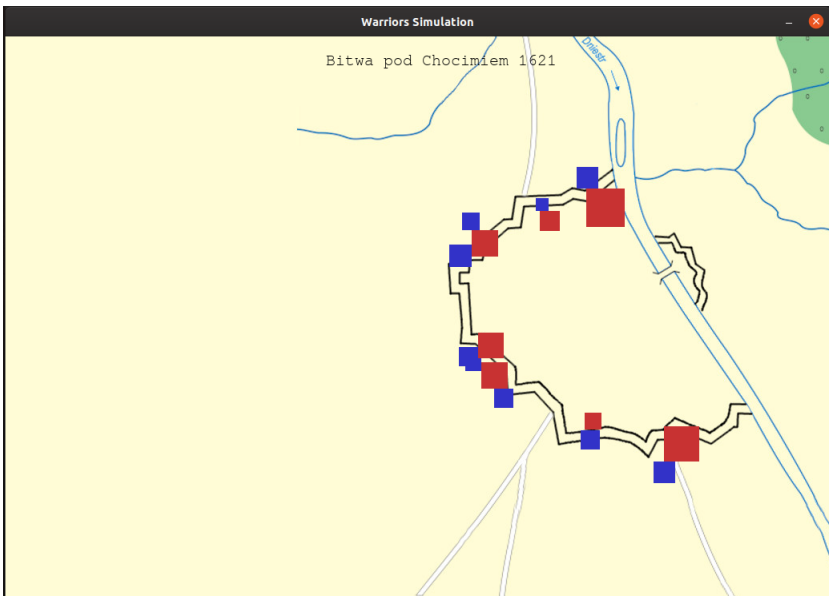
```
ENEMY WINS
Left: piechota_tatarska, Units left = 4375
Left: piechota_tatarska, Units left = 934
```

Rysunek 12: Symulacja bitwy pod Chocimiem 1509 r. - Zwycięstwo tureckie

Powyższa symulacja pokazała, że losy tej bitwy nie były z góry przesądzone. Ogromnym czynnikiem, dzięki któremu wojsko polskie wygrało było m.in. dotarcie husarii na pole bitwy w odpowiednim czasie, dzięki czemu oddziały polskiej piechoty otrzymały pomoc przed rozbiciem przez armię wroga.

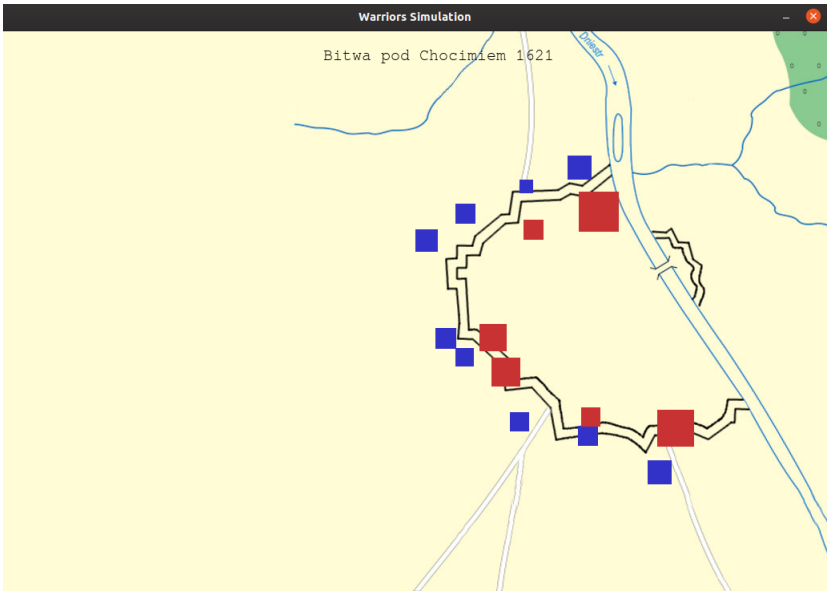
5.2 Bitwa 1621

Bitwa pod Chocimiem w 1621 polegała na oblężeniu przeprowadzonym przez oddziały tureckie. W rzeczywistości bitwa kończy się zwycięstwem Polaków pomimo przewagi liczebnej przeciwników. W naszej symulacji wynik bitwy jest podobny.



Rysunek 13: Niebieski kolor - oddziały tureckie, kolor czerwony - polskie

Jednak, gdy chociaż w jednym miejscu obronnego 'pierścienia' zabrakło by oddziału polskiego to bitwa kończy się zwycięstwem wojsk tureckich. Linia oraz szyk obronny polskich jednostek zostają złamane umożliwiając wtargnięcie oddziałom tureckim za polską linię obrony. Armia polska musi walczyć niejako na dwóch frontach, odpierając Turków chcących przekroczyć defensywę oraz z oddziałami, którym się to udało dzięki złamaniu 'pierścienia' polskiej obrony.

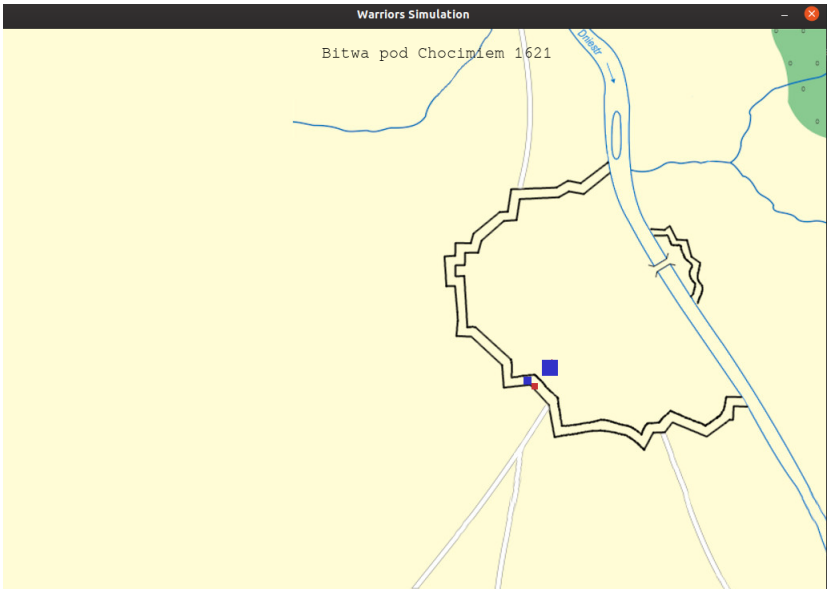


Rysunek 14: Niebieski kolor - oddziały tureckie, kolor czerwony - polskie

```
if pattern == 2:
    return [
        # Jednostki polskie
        [
            Cossack(800, 435, 3600),
            SpearMan(720, 455, 3500),
            LightCavalry(640, 385, 3600),
            SpearMan(610, 350, 4900),
            # Infantry(635, 250, 3200),
            Hussar(705, 230, 3900),
            Cossack(620, 290, 2800),
        ],
```

Rysunek 15: Usunięcie jednego oddziału z polskiej armii

Decydujące w tej bitwie było utrzymanie swoich pozycji przez polskie oddziały najdłużej jak się da, aby nie pozwolić przeciwnikom wedrzeć się za linię obrony i dać czas reszcie oddziałów na uporanie się z wrogami i zgrupowanie w celu pomocy innym. Niewielka zmiana w tej strategii powodowała, że wynik symulacji był zupełnie inny od oczekiwanego.



Rysunek 16: Zwycięstwo wojsk tureckich

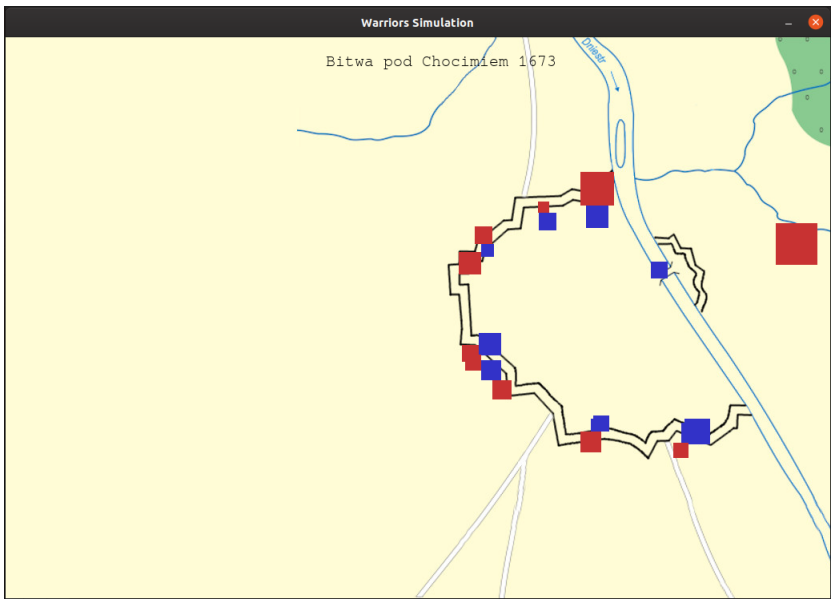
```
ENEMY WINS
Left: piechota_tatarska, Units left = 2292
Left: piechota_tatarska, Units left = 387
```

Rysunek 17: Zwycięstwo wojsk tureckich

Zatem można stwierdzić, że w tej bitwie najważniejszym elementem, dzięki któremu Polacy zwyciężyli tę bitwę była strategia obronna pierścienia. Liczba żołnierzy tureckich, którzy pozostali jest niewielka co pokazuje, że pozycja Polaków w tej bitwie była silna, a przegraną bitwy był skutek usunięcia jednostki polskiej z mapy bitwy.

5.3 Bitwa 1673

Bitwa pod Chocimiem z 1673 roku przedstawia natomiast odwrotną sytuację. To Polacy oblegają przeciwnika i mają zarazem miażdżącą przewagę liczebną. W rzeczywistości bitwa skończyła się zwycięstwem Polaków, którzy rozgromili nawet oddziały chcące wycofać się z pola bitwy.



Rysunek 18: Niebieski kolor - oddziały tureckie, kolor czerwony - polskie

Dzięki symulacji widzimy, że w tej bitwie o zwycięstwie polskich oddziałów zaważyła przede wszystkim przewaga liczebna. Oddziały po kolei eliminowały kolejnych wrogów, a polski oddział kozaków odciął drogę ucieczki wycofującym się jednostkom piechoty wojska tureckiego. Ponownie zauważamy, że wystarcza jedna wyrwa w obronnym 'pierścieniu', aby całkowicie zaprzepaścić szanse na zwycięstwo bitwy i tylko przyspieszyć nadejście porażki. Z kolei dzięki manewrowi odcięcia drogi ucieczki, Turcy nie byli w stanie poprosić o pomoc, czy też przegrupować się i stanąć do walki jeszcze raz, co również mogło mieć wpływ na ogólny przebieg bitwy.

6 Podsumowanie projektu

6.1 Napotkane problemy

Zaczynając projekt natchnęliśmy się na różne wyzwania i problemy do rozwiązania. Podstawowym zadaniem, które stało na naszej drodze było dobre skonstruowanie modelu, na którym mieliśmy opierać nasze bitwy i jakie narzędzia do tego celu wykorzystać. Wykorzystaliśmy bibliotekę PyGame, która ułatwiła nam to zadanie, jak również pomogła we współpracy całego zespołu, gdyż każdy z nas jest w niej bardzo dobrze obeznany, co wpłynęło na dobrą komunikację i rozdzielenie zadań.

6.2 Wnioski

Symulacje starć bitewnych, które udało nam się zrealizować, w dobrym stopniu oddały faktyczne wyniki bitew historycznych, na których się opieraliśmy, co jest dla nas bardzo istotną wiadomością jeśli chodzi o walidację wyników. Mogliśmy z poszczególnych bitew wyciągnąć różne wnioski, np. co zadecydowało o losach bitwy, jak również jak dana bitwa mogłaby się skończyć w przypadku choćby lekkiej zmiany ustawienia wojsk, czy też samej ilości oddziałów. Całą wiedzę, którą zdobyliśmy postaraliśmy się opisać jak najbardziej rzetelnie.

6.3 Źródła

- Marcin Bielski "Kronika Polska"T.3 Księga 6 s.952-954
- https://twojahistoria.pl/encyklopedia/leksykon-bitew/bitwa-pod-chocimiem-9-11-listopada-1673/
- https://twojahistoria.pl/encyklopedia/leksykon-bitew/bitwa-pod-chocimiem-2-wrzesnia-9-pazdziernika-1621/
- https://twojahistoria.pl/encyklopedia/leksykon-bitew/bitwa-nad-dniestrem-4-pazdziernika-1509/
- http://www.borderoftheculturas.archeo.uw.edu.pl/strona,chocim
- https://histmag.org