Symulacja bitwy

Raport

Wykonali: Mikołaj Kardyś

Dominik Sulik

Zadaniem modelu jest zasymulowanie średniowiecznej bitwy za pomocą symulacji agentowej. W bitwie uczestniczą dwie drużyny i kończy się ona, kiedy wszyscy członkowie jednej z nich znikną z planszy (przez śmierć lub ucieczkę).

W symulacji, odległość jest zawsze liczona w sposób kartezjański (pierwiastek z sumy kwadratów różnić współrzędnych).

**Działanie symulacji:**

W każdej iteracji symulacji wykonujemy dla wszystkich agentów (w losowej kolejności) następujące operacje:

- Sprawdzamy po naszych punktach akcji, czy w tej turze możemy wykonać ruch (czy ich wartość jest mniejsza od 1); jeśli nie, odejmujemy 1 i czekamy;

- Jeśli możemy wykonać akcję, to zaczynamy od obliczenia naszego obecnego morale:

* Za wszystkie ciosy otrzymane od poprzedniej akcji otrzymujemy minus do morale równy:
* Za każdego widocznego sojusznika w odległości max. 3 otrzymujemy bonus do morale równy: *wartość\_jednostki / odległość.*
* Za każdego widocznego przeciwnika w odległości max. 3 otrzymujemy minus do morale równy: *wartość\_jednostki / odległość.*

- Następnie w zależności od własności jednostki oraz parametru morale wybieramy jedną z dostępnych akcji:

1. *RUCH* – akcja ta może zostać wykonana tylko w wyniku akcji *UCIECZKA* lub *ATAK*. Polega na poruszeniu się na podane pole, za odpowiedni koszt punktów akcji.\*
2. *ATAK* – zaczynamy od wyboru celu w zależności od następujących kryteriów:

* Jest blisko nas: moc kryterium: 1 / *odległość*
* Riposta – zaatakował nas od naszej ostatniej akcji; moc kryterium: 4
* Jego żywotność jest niższa od naszej; moc kryterium: 2
* Od jego ostatniej akcji był atakowany przez innych członków naszej drużyny (liczba ataków: ); moc kryterium:

Celem w zbiorze przeciwników będzie ten, dla którego wartość iloczynu tych kryteriów jest najwyższa.

- Jeśli żaden przeciwnik nie znajduje się w naszym zasięgu, to cel wybieramy ze zbioru wszystkich widocznych przeciwników i wykonujemy akcję *RUCH* na pole znajdujące się najbliżej niego;

- Jeśli w naszym zasięgu znajdują się przeciwnicy, to cel wybieramy z ich zbioru. Po zadanym ciosie,\* otrzymujemy liczbę punktów morale równą parametrowi *wartość\_jednostki* tej jednostki. Jeśli przeciwnik ginie od tego ciosu, otrzymujemy drugie tyle.

1. *UCIECZKA* – rozpatrujemy wszystkie pola w naszym bezpośrednim sąsiedztwie. Dla każdego z nich liczymy jego średnią odległość od wszystkich naszych widocznych przeciwników. Wybieramy to, które znajduje się najdalej i wykonujemy na to miejsce akcję *RUCH*. Jeśli znajdujemy się bezpośrednio na krawędzi planszy ustawiamy flagę *escapes* na prawda; w następnej iteracji symulacji zostaniemy dzięki temu usunięci z planszy.

Uwaga*:* Jeśli nie jest możliwe wykonanie tej akcji, a chcemy ją wykonać, to jeśli w naszym bezpośrednim sąsiedztwie znajdują się przeciwnicy, domyślnie będziemy próbowali wykonać akcję *ATAK*. W przeciwnym razie, wykonamy akcję *CZEKAM*.

1. *CZEKAM –* nie wykonujemy żadnej z powyższych akcji. Odnawiam 10% moich punktów zdrowia i pozostaję z tą samą ilością punktów akcji.

\* – koszt akcji *RUCH*, a także siła oraz koszt ciosu w akcji *ATAK* są zależne od typu jednostki oraz terenu. Koszt każdej z tych akcji ponosimy, dodając go do obecnej wartości naszych punktów akcji.

**Typy jednostek:**

Typy jednostek oparte są na rodzajach żołnierzy biorących udział w bitwach w okolicy bitwy pod Hastings (1066 r.).

1. Piechota

Lekkie odziały walczące w ręcz. Nie byli to profesjonalni żołnierze, ich arsenał i użyteczność na polu bitwy były więc mocno ograniczone.

Dodatkowe zachowanie: *brak*

Interakcja z innymi jednostkami:

Ze względu na niski stopień wytrenowania i słabe uzbrojenie, jednostki tego typu będą słabe przeciwko wszystkim pozostałym jednostkom walczącym wręcz.

Interakcja z terenem:

Lekkie uzbrojenie ułatwia poruszanie się. Jednostki tego typu otrzymują bonus do akcji na zwykłym terenie oraz średnie kary w lesie i w rzece.

1. Ciężka piechota

Jednostki wzorowane na oddziałach anglosaskich huskarli, ubranych w ciężkie kolczugi i uzbrojonych w włócznie, krótką broń (zwykle topór, rzadziej miecz) i dużą tarczę, a czasem także długie duńskie topory.

Dodatkowe zachowanie:

FORMACJA – jednostki tego typu w naszej symulacji będą poruszały się inaczej niż zwyczajna piechota. Agenci tego typu będą w pierwszym etapie starcia kłaść większy nacisk na trzymanie szyku, symulując zachowanie formacji zwanej Murem Tarcz.

Interakcja z innymi jednostkami:

Dobre wyszkolenie, uzbrojenie oraz dyscyplina daje im wysoki bonus do walki z lekką piechotą i kawalerią. Mniejsze tempo poruszania się, zarówno przez obciążenie jak i poruszanie się w formacji czyni ich słabym przeciwko łucznikom.

Interakcja z terenem:

Ciężar uzbrojenia sprawia, że na terenie otwartym i w lesie jednostki tego typu otrzymują nieco większe kary niż zwykła piechota. Wpływa to jednak najgorzej na akcje w rzece, dając do nich bardzo wysoką karę.

1. Kawaleria

Na przełomie pierwszego i drugiego tysiąclecia kawaleria powoli zaczynała się kształtować w znanych nam ze schyłku epoki rycerzy w zbrojach płytowych. W tym okresie jednak wciąż jeszcze ciężka kawaleria nie była stałym widokiem na polu bitwy. Zarówno w armii bizantyjskiej, jak i normańskiej znajdowały się lekkie konne jednostki, pełniące głownie funkcje zwiadowcze.

Dodatkowe zachowanie:

LUŹNY SZYK – jednostki tego typu będą próbowały trzymać niewielki odstęp od innych sojuszników, co ułatwia im zachowanie zdolności do manewrowania

KONTOLOWANY ODWRÓT – z uwagi na dominującą prędkość tego typu oddziałów na polu bitwy, nie musiały się one zbyt obawiać pościgiem ze strony innego typu jednostek. Będą więc stopniowo odzyskiwać morale, jeśli przy próbie ucieczki znajdą się odpowiednio daleko od wroga.

Interakcja z innymi jednostkami:

Uzbrojenie i wyszkolenie dają tym oddziałom przewagę nad zwykłą piechotą, a szybkość poruszania się i luźna formacja – przeciwko łucznikom. Mimo to, w zwarciu nie będą tak skuteczni jak ciężka piechota, otrzymują więc karę przy starciu z nią.

Interakcja z terenem:

Poruszanie się na grzbiecie konia zapewnia tym jednostkom najwyższą ze wszystkich skuteczność poruszania się po otwartym terenie, jak i pozwala najłatwiej pokonywać rzekę. Stanowi ono jednak przeszkodę przy poruszaniu się po lesie, dając wysoką karę w tym obszarze.

1. Łucznicy

Łucznicy byli popularnym typem jednostki na polu bitwy od starożytności. Podczas inwazji Anglii normańscy łucznicy stanowili ważną część armii Wilhelma Zdobywcy. Będzie to jedyna w naszej symulacji jednostka walcząca na dystans, co pozwoli jej na dosięgnięcie przeciwników zanim ci będą mogli zadać jej szkodę.

Dodatkowe zachowanie:

TRZYMAJ POZYCJĘ – jednostki tego typu nie będą szarżować tak jak pozostałe, zachowując energię, aby móc zaatakować wroga kiedy tylko wejdzie w ich zasięg.

KROK W TYŁ – łucznicy będą za wszelką cenę unikali walki wręcz, wykonując jednorazowo akcję *UCIECZKA,* jeśli przeciwnik podejdzie zbyt blisko, dając szansę na to, aby dosięgnęły go ataki reszty sojuszników. Gdy w zasięgu nie będzie już więcej przeciwników, jednostka wróci na swoje miejsce.

Interakcja z innymi jednostkami:

Jednostki tego typu otrzymują bonus do walki z ciężką piechotą ze względu na jej niskie tempo poruszania się, ale z tego samego powodu otrzymują też karę do walki z kawalerią.

Interakcja z terenem:

Lekkie uzbrojenie daje tym jednostkom takie same możliwości poruszania się, jakie ma lekka piechota. Jednak z uwagi na sposób ataku, jednostki tego typu jako jedyne otrzymują dodatkowy minus do jego siły przy walce w lesie.

**Generowanie terenu**

Do wygenerowania terenu użyliśmy następujących algorytmów:

1. Algorytm marching squares
2. Perlin noise
3. Algorytm Bresenhama
4. Algorytm A\*
5. Wysokość terenu nad poziomem morza

Każda komórka na mapie posiada parametr *altitude*, który określa wysokość tego punkt nad poziomem morza. Do wygenerowania tej cechy wykorzystaliśmy szum Perlina. Dla każdej komórki o współrzędnych *x* i *y* wartość

*altitude* *= noise(x\*terrainFrequency, y\*terrainFrequency)*,

gdzie *terrainFrequency* jest liczbą z zakresu *(0;1)*.

Wysokość terenu wpływa na szybkość poruszania się wszystkich jednostek. Tempo ruchu spada gdy jednostka idzie pod górę oraz rośnie gdy jednostka schodzi w dół.

1. Las

Dla każdego z punktów węzłowych komórek na mapie wygenerowaliśmy szum Perlina jw. (z inną wartością częstotliwości - *forestFrequency*). Następnie wszystkie wygenerowane wartości sprawdziliśmy warunkiem *noise >= sparseForestThresh*, tym punktom dla których warunek okazał się prawdziwy przypisaliśmy wartość *1*, pozostałym *0*. Otrzymaną siatkę wartości binarnych potraktowaliśmy algorytmem marching squares. W wyniku tej operacji otrzymaliśmy struktury przypominające las, który tym samym algorytmem podzieliliśmy na las gęsty (*denseForestThresh*) i las rzadki.

1. Rzeka

Dla każdego z punktów węzłowych komórek na mapie wygenerowaliśmy szum Perlina, który następnie poddaliśmy binaryzacji progowej jw. (odpowiednio z *obstaclesFrequency* i *obstaclesThresh*). Następnie wylosowaliśmy dwa punkty na granicach mapy. Ostatnim krokiem było zastosowanie algorytmu A\*, który znalazł najkrótszą (o ile istnieje) ścieżkę pomiędzy punktami na brzegach mapy. Tak powstała ścieżka to rzeka.

1. Widoczność

Do poruszania się jednostek niezbędna jest funkcja odpowiedzialna za określanie czy z danego punktu na mapie widać inny punkt.

Do zaimplementowania tej metody użyliśmy algorytmu Bresenhama (służącego do wyznaczania komórek pomiędzy dwoma pozycjami na siatce).

Dla tak wyznaczonych punktów obliczyliśmy następującą wartość *visibility* (zainicjalizowaną na zero):

- jeśli punkt znajduje się w gęstym lesie wówczas odejmij od *visibility* 2

- jeśli punkt znajduje się w rzadkim lesie wówczas odejmij od *visibility* 1

- w przeciwnym wypadku nie odejmuj od sumy

Następnie w zależności od położenia jednostki która obserwuje daną lokację oraz otrzymanej sumy określiliśmy widoczność punktu:

* jeśli jednostka znajduje się w lesie rzadki, a obserwowana przez nią pozycja na łące, wówczas próg (*fromForestVisibility*) odpowiedzialny za dostrzeżenie pola jest średni,
* jeśli jednostka znajduje się na łące, a obserwowana przez nią pozycja w lesie, wówczas próg (*toForestVisibility*) odpowiedzialny za dostrzeżenie pola jest wysoki,
* w pozostałych przypadkach próg (*throughForestVisibility*) widoczności jest niski

**Symulacja**

Przeprowadziliśmy symulację bitwy dla wylosowanego terenu oraz z następującym rozkładem jednostek: