

Plemiona

DOKUMENTACJA PROJEKTU

Grupa w składzie:

Marcin Borkowski

(272974)

Jakub Swatowski

(272911)

Jakub Cieślicki

(272999)

Spis treści

Opis symulacji.....	3
System poruszania się	3
System walki	3
Dane wejściowe.....	5
Diagram klas	6
Diagram obiektów	7
Dokumentacja JavaDocs	7
Repozytorium GitHub	7

Opis symulacji

W symulacji występuje 5 plemion:

- żółte,
- białe,
- fioletowe,
- czarne,
- zielone.

Każde z nich posiada unikalnie rozwinięte umiejętności i cechy. Celem symulacji jest walka pomiędzy plemionami, rozmieszczonymi losowo na planszy wyświetlanej na interfejsie UI. Pojedynki pomiędzy jednostkami trwają tak długo, dopóki na planszy znajduje się co najmniej jedna jednostka w co najmniej dwóch plemionach, tym samym symulacja kończy się w momencie, gdy na planszy pozostaną jednostki z tylko jednego plemienia. Parametry jednostek każdego plemienia zostały ustawione i zróżnicowane w taki sposób (Tabela 1.), by wynik symulacji był jak najbardziej nieprzewidywalny na samym starcie. W żadnym plemieniu wartości atrybutów jednostki nie są identyczne (wyjątkiem jest jedynie atrybut odpowiadający za prędkość poruszania). Suma atrybutów każdego z plemion sprowadza się do tej samej wartości, a dzięki odpowiedniemu zróżnicowaniu, wszystkie mają podobne szanse na zwycięstwo.

Atrybuty	Green	Black	White	Yellow	Purple
Strength	1	5	3	4	2
Iq	2	1	4	5	3
Agility	4	3	1	2	5
Endurance	5	3	4	1	2
Suma atrybutów	12	12	12	12	12

Tabela 1. Przedstawienie rozkładu wartości poszczególnych atrybutów pomiędzy plemionami.

System poruszania się

System przemieszczania się jednostek po planszy jest oparty na liczbach pseudolosowych generowanych przez odpowiednią klasę. Najpierw losowana jest liczba dla zmiany pozycji w osi poziomej (x), a następnie w osi pionowej (y). Wylosowana liczba całkowita z zakresu od -1 do 1 zostaje przemnożona przez wartość mnożnika prędkości w danej osi. Na końcu tego procesu następuje stworzenie wektora (x, y), na podstawie wylosowanych liczb, sprawdzenie czy na danym polu nie znajduje się już inna jednostka (jeśli **tak**, następuje pojedynek), a następnie przemieszczenie członka plemienia w wylosowanym wcześniej kierunku na nową pozycję.

Uwaga: Może wystąpić sytuacja, w której jednostka nie wykona ruchu. Jeżeli liczba x i y na skutek losowania wyniesie 0, jednostka nie przemieści się. Dalej może ona jednak brać udział w walce i bronić swojej pozycji.

System walki

Walka następuje w sytuacji, gdy na tym samym polu chcą się znaleźć dwie lub więcej jednostek.

Uwaga: walka pomiędzy jednostkami należącymi do tego samego plemienia jest niemożliwa. Zatem wszystkie możliwe przypadki pojedynku to:

1. Dwie jednostki chcą się przenieść na to samo pole.
2. Więcej niż dwie jednostki chcą się przenieść na to samo pole.
3. Jednostka nie poruszyła się w trakcie tury i dalej znajduje się na polu, na które teraz chce wejść członek innego plemienia.
4. Jednostka nie poruszyła się w trakcie tury i dalej znajduje się na polu, jednak teraz chcą wejść na to pole członkowie co najmniej dwóch różnych plemion.

Sytuacja 1 (Dwie jednostki chcą się przenieść na to samo pole):

Najpierw następuje losowanie, który spośród wszystkich atrybutów będzie decydował o zwycięstwie w pojedynku, następnie atrybuty są porównywane i ostatecznie na nową pozycję przenosi się jednostka zwycięska, a pokonany członek plemienia jest usuwany z planszy.

Sytuacja 2 (Więcej niż dwie jednostki chcą się przenieść na to samo pole):

Po ustaleniu przez symulację, gdzie znajdzie się która jednostka, następuje sprawdzenie ilu członków plemienia znajdzie się na tej samej pozycji. Jeżeli jest ich więcej niż 2, dochodzi do losowego wyboru dwóch z nich. Następnie cały proces przebiega identycznie jak w sytuacji 1. Po wyłonieniu zwycięzcy dochodzi do kolejnego pojedynku. Proces powtarzany jest do momentu wyłonienia jednego zwycięzcy.

Sytuacja 3 (Jednostka nie poruszyła się w trakcie tury i dalej znajduje się na polu, na które teraz chce wejść członek innego plemienia):

Proces przebiega identycznie jak w sytuacji 1.

Sytuacja 4 (Jednostka nie poruszyła się w trakcie tury i dalej znajduje się na polu, jednak teraz chcą wejść na to pole członkowie co najmniej dwóch różnych plemion):

Proces przebiega identycznie jak w sytuacji 2.

Jednostki, które nie poruszają się w trakcie swojej tury nie otrzymują z tego powodu żadnych kar, ani bonusów.

Dane wejściowe

Przed rozpoczęciem symulacji, należy podać dane wejściowe do których zaliczamy:

- ilość osadników z każdego plemienia na mapie,
- wielkość planszy.

Każda z tych liczb musi spełniać pewne wymagania, aby symulacja mogła poprawnie działać. Warunki są następujące:

- suma członków wszystkich plemion, nie może być większa niż liczba pól dostępnych na planszy (plansza jest kwadratowa, więc łączna ilość osobników nie może być większa niż kwadrat długości boku planszy);
- liczba członków plemienia nie może być liczbą ujemną,
- długość boku planszy nie może być liczbą ujemną.

Diagram klas

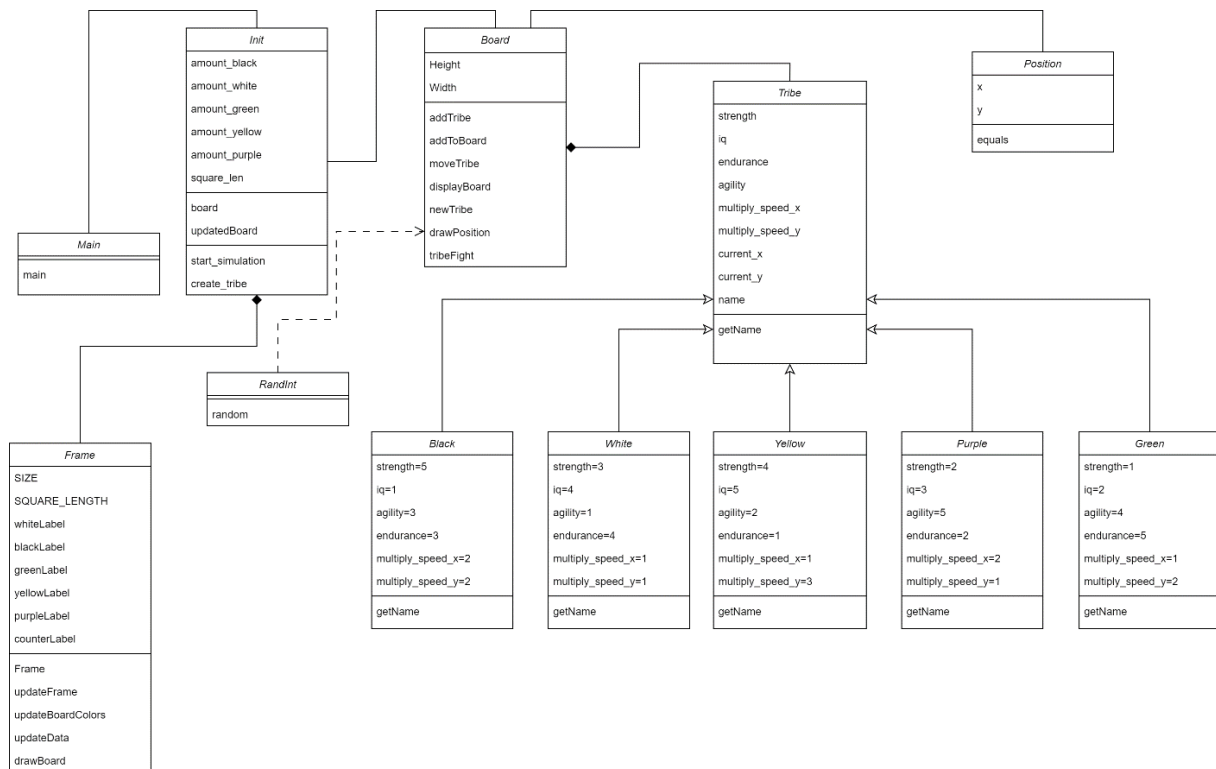
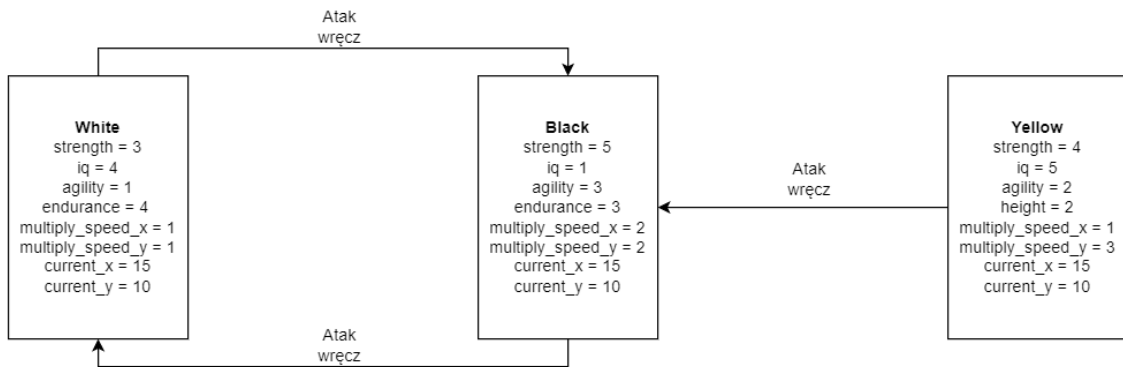


Diagram przedstawia klasy używane podczas działania symulacji. Strzałkami zaznaczono odpowiednie relacje pomiędzy strzałkami. W wewnętrznych prostokątach zostały zawarte atrybuty i metody opisane w klasach.

Diagram obiektów



Na diagramie przedstawiono przykładowe starcie pomiędzy jednostkami z różnych plemion, które chcą być na tej samej pozycji. W wyniku losowania z trzech plemion „White”, „Black” i „Yellow” dwa pierwsze zostały wytypowane do pojedynku między sobą. Jeżeli jako decydujący o zwycięstwie atrybut zostanie wylosowane „iq” lub „endurance” zwycięży jednostka należąca do plemienia białych. Jeżeli jednak będzie to „strength” lub „agility” wygrają czarni. Zwycięzca pojedynku zmierzy się z członkiem plemienia żółtych. W tym pojedynku ponownie zostanie wylosowany atrybut decydujący. Można zauważyć, że przy takim zestawieniu nie można łatwo przewidzieć ostatecznego zwycięzcy. Jeżeli pierwszy pojedynek zostanie wygrany przez „Białego”, w kolejnym większe szanse na zwycięstwo będzie miał „Żółty”, ponieważ wygrywa on z „Białym” w trzech z 4 atrybutów. Jeśli jednak zwycięży „Czarny”, to „Żółty” nie będzie już faworytem drugiego pojedynku („trzy do jednego” w atrybutach na korzyść „Czarnego”).

Dokumentacja JavaDocs

Fragment dokumentacji wygenerowany przez komponent JavaDocs można znaleźć w folderze „Docs” w pliku index.html.

Repozytorium GitHub

Link do repozytorium z całym projektem:

https://github.com/MBCstudio/Race_race_master_v2