Strata

# Идея и суть проекта

# Class division

# Функция Animation()

## Флаги

## Вращение

## Перемещение

# Миникарта

# Проверка на коллизию

# Система урона

Для одержания побед в сражениях, вам нужно следить за тем, чтобы ваши отряды были направлены фронтом (жёлтый шарик) в сторону противников.

В игре действует ограничение на количество одновременно двигающихся отрядов. Вы не можете двигать отряды, если на поле в движении находится два отряда, включая противников. В то же время, недостаток умственного развития бота компенсируется его возможностью ходить под ограничением в три отряда. Хоть он переключается между отрядами довольно быстро, больше двух он вряд ли когда-нибудь запустит в движение. У вас есть возможность перехватить инициативу, если будете кликать достаточно быстро и займёте весь пул движущихся отрядов своими.

Не забывайте о возможности выбора нескольких отрядов. Однако, не всегда отряды выполняют команды в правильном порядке. В битве при Каннах и других ориентируйтесь не на порядок выбора отрядов, а на номер в их названии. Они будут двигаться всегда в одном порядке от большего к меньшему или наоборот – уже не помню.

A – атакующий;

D – защищающийся;

Определяется тем, кто вызвал событие коллизии.

da — разность между углом фронта отряда и истинным углом между отрядами.

Урон атакующего = A.attack \* |Cos(daA)| \* D.armor \* A.hp / 100;

Ответный урон = D.attack \* |Cos(daD)| \* A.armor \* D.hp / 100;

D.hp -= damA;

A.hp -= damD;

При атаке, на расчёт защищающийся влияет только своей бронёй.

ПРИМЕР: разработать характеристики катапульты.

1. В атаке: большой урон и выживаемость;
2. В защите: низкий ответный урон и вероятная гибель.

Из 1 очевидна высокая броня и высокий хп.

Из 2 очевидно низкое хп и высокая броня.

Взаимоисключающие параметры.

1. Урон атакующего ~ attack \* hp / 100;

Ответный урон ~ armor;

1. Урон атакующего ~ armor;

Ответный урон = attack \* hp / 100;

Создаётся ощущение, что они оба наносят друг другу «атакующий» урон

Возможно, стоит снимать хп у защитника сразу после атаки, и только тогда запускать ответный удар.

А как будет честнее и реалистичнее?

Похоже, что другого выбора, чтобы реализовать катапульту, нет. Ввожу изменения и начинаю испытания.

\*\*\*

Это решение значительно усложнило игру, так как бот ходит намного чаще игрока, а значит снимает наши хп быстрее. Считаю улучшение правильным.

damA = A.attack \* |Cos(daA)| \* D.armor \* A.hp / 100;

D.hp -= damA;

damD = D.attack \* |Cos(daD)| \* A.armor \* D.hp / 100;

A.hp -= damD;

Получилось сделать катапульты смертными при характеристиках

hp = 100;

speed = 3;

rot\_speed = Math.PI / (120);

attack = 100;

width = 80;

height = 80;

armor = 1;

Но они капитально проигрывают в битве против стены.

Попробуем увеличить атаку до 250.

Стало лучше, но стену не берут.

Сделаем больше хп и уменьшим броню.

Напоминаю, что чем значение брони меньше, тем выше выживаемость.

hp = 150;

armor = 1.5;

этого хватает с избытком. К тому же, есть проблема, что катапульта ломает с двух тычек пехоту противника в лоб.

Кроме того, возникла проблема, что при высоком уроне хп атакующего регенерируют.

Не вижу ничего отрицательного в формуле урона и не знаю, откуда могла возникнуть проблема.

Проблема возникает от того, что хп защитника становится отрицательным.

Исправлено через if().

Ну вот вроде и устаканилась ситуация с катапультой.

В новой системе урона выяснилось, что и простая пехота получает только 50 урона прежде чем снести стену, хотя раньше стена уничтожала полтора отряда.

Увеличим урон стены.

Пока что всё, чего удалось добиться – катапульта разменивает своё хп на урон намного быстрее пехоты.

… спустя несколько опытов:

if (clas == "gun")

{

hp = 180; //\* longg;

speed = 3;

rot\_speed = Math.PI / (120); //\* longg);

attack = 250;// + 70\* longg;

width = 80;// + 60\* longg;

height = 80;

image = Image.FromFile("gun.png");

armor = 1.5;

}

if (clas == "wall")

{

hp = 100; //\* longg;

speed = 0;

rot\_speed = Math.PI / (100); //\* longg);

attack = 30;// + 70\* longg;

width = 130;// + 60\* longg;

height = 50;

image = Image.FromFile("wall.png");

armor = 0.05;

}

Вроде баланс достигнут.

# Режим игры Песочница

## Разные виды отрядов

## Испытание отрядов

## Сохранения

## Создание кастомных сохранений

# Режим игры Сражение

## Реконструирование формы

# Действия бота