

GAME AI

# Pracownia Gier w OpenGL

Michał Drobot  
Drobot.org

# Plan wykładu : Podstawy

- AI - nauka a praktyka
- Znaczenie i zastosowanie w grach
- Podstawowe techniki AI
- Podstawy budowy systemów AI
- Praktyczne zastosowania w grach

# AI - nauka a praktyka

- Nauka dąży do maksymalnego odwzorowania poprawności opisu matematycznego rzeczywistości
  - Często mało wydajne rozwiązania
  - Na obecnym etapie nie istnieje system będący w stanie udawać człowieka w kompleksowym środowisku gry
- Produkcja AI do gier kładzie nacisk na
  - Tworzenie percepcyjnie poprawnego AI
  - Wydajność oraz skalowalność systemu

# AI - nauka a praktyka

- AI w grach stanowi połączenie
  - Rozwiązań naukowych
    - Przeskalowanych pod względem wydajności oraz skali do świata gier
  - Rozwiązań proceduralnych
  - Rozwiązań odgórnych (skrypty itp..)

# Znaczenie i zastosowanie

- W przeciągu ostatniej dekady doszło do rewolucji graficznej oraz gameplay'owej
- AI jedynie ewoluowało, często jednak pozostając na podobnym poziomie
  - Pierwsze AI do FPP miały 20 linii kodu
    - Obecnie zdarza się, że AI stoi na podobnym poziomie

# Znaczenie i zastosowanie

- Sytuacja wynikła z
  - Braku wsparcia oraz istnienia konkretnej platformy
    - Brak SDK jak w środowiskach graficznych
  - Stosunkowo duży nacisk lobby graficznego
  - Małe zainteresowanie graczy
  - Brak zasobów pamięciowych oraz wydajnościowych

# Znaczenie i zastosowanie

- Obecnie coraz większy nacisk na rozwój AI
  - Często ważny element gameplay'u
  - Wizualizacje efektownych zjawisk wymagają systemów AI
    - Zarządzanie tłumem
    - Bitwy
    - Symulacje miejskie
    - Zaawansowane systemy cząteczkowe

# Znaczenie i zastosowanie

- Do systemów AI zaliczamy systemy
  - Obsługi animacji
    - Blendowanie animacji
    - Animacja proceduralna
    - Animacja behawioralna
    - Systemy poznawcze świata gry
      - Dynamiczne zintegrowane ze światem fizycznym
      - Statyczne
  - Zarządzania rozgrywką
    - Adaptacja poziomu trudności
    - Dynamiczne tworzenie scenariuszy

# Znaczenie i zastosowanie

- Zaawansowane metody AI w grach
  - F.E.A.R
    - Zaawansowane AI oparte na FSM, Fuzzy Logic oraz pamięci udającej system uczący
  - FORZA 2
    - Sieci neuronowe dokonujące korekt skrętów podczas przejazdu samochodów przez tor
  - Spore
    - Proceduralne AI grup oraz animacji szkieletowej
  - Left 4 Dead
    - AI Director - System zarządzający poziomem trudności oraz tworzący scenariusze dla graczy

# Znaczenie i zastosowanie

- Zaczyna pojawiać się middleware
  - Xaitment
    - Kompletny system wraz z narzędziami do obsługi hierarchicznych FSM, Fuzzy Logic, Path Planning, Crowd control, Procedural Behaviour
  - Euphoria
    - System animacji proceduralno-behawioralnej oparty na biomechanice i reakcjach natychmiastowych żywych organizmów

# Znaczenie i zastosowanie

- Wystarczająca moc obliczeniowa
  - GPU
    - Obliczenia równoległe na dużych zbiorach danych
      - Flocking
      - Steering
      - Particles
  - Multithreading
    - Możliwość desygnacji pojedynczego wątku / CPU na stałe dla obliczeń AI

# Podstawowe techniki AI

- FSM
- Autonomous agents
- Steering behaviours
- Flocking behaviours
- Graph search
- Path planning
- Fuzzy logic

# Podstawowe techniki AI

## ■ FSM

- Maszyny stanów skończonych
- Hierarchiczne, uruchamiające pomniejszych automaty
- Rozbudowane systemy wejść, wykonań oraz wyjść ze stanu do stanu w razie zaistnienia konkretnych warunków

# Podstawowe techniki AI

- Autonomous agents
  - Jednostki posiadające warstwę percepcyjną, decyzyjną oraz lokomocyjną
    - Percepcyjna automatycznie „poznaje otoczenie” np. poprzez próbkowanie przestrzeni dookoła
    - Decyzyjna podejmuje konkretne działania jak np. podążanie za zapachem, do celu itp..
    - Lokomocyjna przemieszcza jednostkę w przestrzeni symulacji wg decyzji wyższych warstw

# Podstawowe techniki AI

- Steering behaviours
  - Systemy operujące na obsłudze dużej ilości Autonomicznych Agentów
  - Wydzielono konkretne rodzaje zachowań dla AA
    - Follow
    - Seek
    - Arrive at
    - Evade
    - Escape
    - Etc.

# Podstawowe techniki AI

- Flocking behaviours
  - Rozszerzają Steering Behaviours o dodatkową warstwę
    - Warstwa zachowań grupowych
      - Parametrycznie wyznaczana jest decyzja ogólna grupy
      - Zachowanie całej grupy jest brane pod uwagę w podejmowaniu decyzji przez jednostkę
      - Dodaje zachowania : zgrupowania, rozproszenia, spójności
    - Umożliwia zachowania stadne
      - Ryb
      - Ptaków
      - Szarańczy
      - Cząsteczek
      - Etc.

# Podstawowe techniki AI

- Graph search
  - Podstawowe techniki przeszukiwań grafów
  - Grafy rozmieszczane dynamicznie w środowisku gry bądź ręcznie przez designera w zależności od funkcji i potrzeb

# Podstawowe techniki AI

## ■ Path planning

- Planowanie przejść w środowisku (np. grafowym) przy uwzględnianiu dodatkowych czynników
  - Opłacalności konkretnej ścieżki
    - Czas
    - Koszt
    - Ryzyko
  - Działań innych jednostek w świecie gry
    - Ominięcie wrogiej armii w RTSie
  - Celów długoterminowych
    - Odcięcie drogi ucieczki przeciwnikowi

# Podstawowe techniki AI

## ■ Fuzzy logic

- Logika rozmyta używana w automatach decyzyjnych
- Wprowadza odrobinę randomizacji w podejmowanych decyzjach
- Decyzje nie opierają się na wynikach binarnych lecz na języku zrozumiałym dla „większości ludzi” ;]
  - Dalej
  - Bliżej
  - Niedaleko
  - Mocno
  - Słabo
  - Etc.

# Podstawy systemów AI

- System decyzyjny
  - Rodzaje rozwiązań
    - Decyzje strategiczne
      - Ocena celów które system chce osiągnąć
    - Decyzje taktyczne
      - Wybranie metody która zostanie użyta do osiągnięcia celu
  - Reaktywność agentów
    - Czas podejmowania decyzji uzależniony probabilistycznie / w zależności od potrzeb i gatunku
  - Realizm systemu
  - Gatunek

# Podstawy systemów AI

## ■ Nawigacja

- systemy gridowe
  - Siatki kwadratowe, hexagonalne wg potrzeb gry (strategie...)
- pola stanów
  - Tzw. Triggery – obszary o określonych parametrach, mogą wyzwać pewne akcje
- mapy grafowe
  - Systemy połączonych grafów
- siatki nawigacyjne
  - Navmeshe – wstawiane ręcznie bądź dynamicznie, wyznaczają obszary w przestrzeni, po których określone jednostki mogą się poruszać ze względu na swoją charakterystykę fizyczną (wielkość, zwinność itp.)

# Przykładowe zastosowania

## ■ Assassins Creed

- System behawioralnego zarządzania tłumem
  - Tłum < 50 NPC na ekranie (w pamięci), NPC wychodzące z pola widzenia są 'spwanowane' w nowych alejkach
  - Tłum reaguje na poczynania gracza
    - Wznosi okrzyki
    - Wspiera w walce, bądź utrudnia
    - Unika stratowania
    - Zachowuje się naturalnie : NPC chodzą, rozmawiają, handlują itp.

# Przykładowe zastosowania

## ■ Assassins Creed

### ■ System biomechanicznej animacji

- Połączenie silnika fizycznego z IK
  - Postać gracza podczas wspinaczki po ścianach ‘idealnie’ łapie się i przylega do krawędzi
- W momencie uderzenia, popchnięcia, upadku, utraty równowagi NPC zachowują się naturalnie, próbując zachować balans itp.. Przy użyciu IK i inteligentnie blendowanej animacji

# Przykładowe zastosowania

## ■ F.E.A.R 2

- Goal Oriented Action Planning System
  - Priorytetowy system decyzyjny
    - Wyznacza kilka podstawowych celów
      - Przetwanie, zabicie wroga, pomoc kompanowi
    - W zależności od sytuacji i opisujących ją czynników dokonuje wyboru
    - Przetacza się w nową maszynę decyzyjną dotyczącą celi pobocznych koniecznych do wykonania danej akcji
    - System dociera do akcji atomowych i wykonuje je dynamicznie korygując swoje decyzje

# Przykładowe zastosowania

## ■ F.E .A.R 2

### ■ Squad Based AI

- System odgórnego zarządzania oddziałem
- Wyznacza nadrzędne cele
  - Otoczenie przeciwnika
  - Wspomaganie oddziału
    - Ogień zaporowy
    - Wsparcie medyczne
  - Decyzje podejmowane są na podstawie określonych danych parametrycznych
- Podejmuje decyzje które są przekazywane niżej do pojedynczych NPC i ich systemów decyzyjnych

# Przykładowe zastosowania

## ■ Left 4 Dead

### ■ AI director

- System zarządzania globalnym AI przeciwników
- W zależności od postępów oraz umiejętności gracza
  - Dostosowuje liczbę, agresywność oraz typ przeciwników
  - Wybiera odpowiednie miejsca taktyczne do spawnowania NPC
  - Dynamicznie tworzy 'scenariusze' akcji z pewnych gotowych sytuacji wiązanych w łańcuchy zdarzeń
- Bazuje na wiedzy zdobytej podczas gry gracza
  - Analizuje
    - Czasy reakcji gracza, jego stan percepcji itp.
    - Poziom zaangażowania gracza w akcje
    - Poziom niepokoju (używanie apteczek itp..)
    - Poziom zaawansowania gracza

# Przykładowe zastosowania

## ■ XAitment

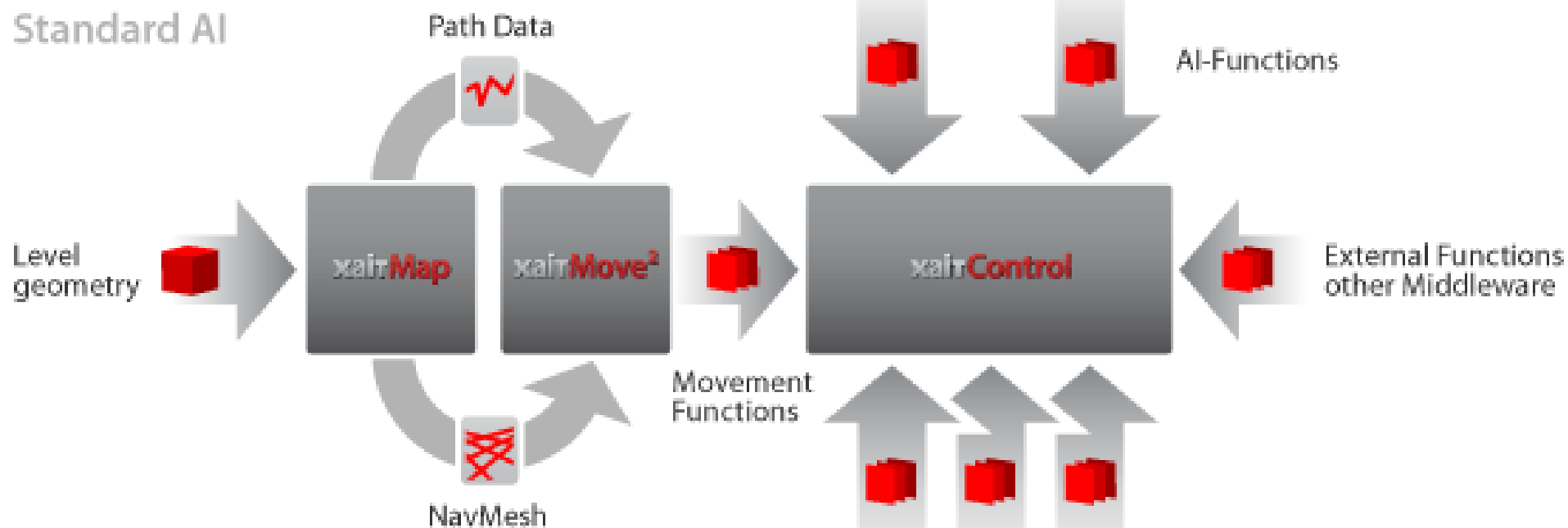
- Kompleksowy system zarządzania AI
- Dzieli się na moduły nisko i wysoko poziomowe
  - Nisko
    - Map - dynamicznie bądź statycznie buduje opis świata 3D
    - Move - porusza się po opisie świata 3D
    - Control - opiera się na FSM i odpowiada za akcje atomowe kontrolując animacje, dźwięk, lokomocje i inne warstwy
  - Wysoko
    - Think - podejmuje decyzje wysokopoziomowe i zleca ich wykonanie 'niżej'
    - Know - zapamiętuje wiedze o świecie przydatną przy podejmowaniu decyzji (np współczynniki niebezpieczeństw itp..)

# Przykładowe zastosowania

High-Level AI



Standard AI



Game

Animation Graphics / FX Functions  
Sound Functions  
Any other Functions

# Plan wykładu : AI w TW:TT

- Pathfinding w TW: The Temptation
- Zachowania AI
- System walki

# Pathfinding

- Przeszukiwanie grafów (Graph search)
  - Podczas standardowego poruszania się NPC-ów w świecie
    - Wykonywanie codziennych czynności
    - Czynności zaplanowanych (questy, cutscenes)
- RRT (rapidly-exploring random tree)
  - Dynamiczna struktura badająca otoczenie w 3D dookoła NPCa
  - Buduje się dynamicznie wraz z zapotrzebowaniem
    - Walka w nieznanym terenie
  - Ma ograniczenie ze względu na czas wykonywania

# Pathfinding

- Graph search
  - W przestrzeni przewidywalnej
    - Miasta
    - Pomieszczenia
    - Labirynty
    - Zawitych częściach lasu itd.

# Pathfinding



# Pathfinding

## ■ RRT

- W przestrzeni nieprzewidywalnej
  - Lasy
  - Pustynia
  - Miejsca nie wyposażonych przez Leveldesignera w grafy (mogą to być również zaułki w miastach)

# Pathfinding



# Pathfinding & AI

- Dalsze zastosowania grafów
  - Node'y zawierają dodatkowe opisy
    - Node'om możemy przypisać prace
      - Np. sprzątanie, siedzenie, jedzenie itd.
    - Node'y mogą być manipulowane z poziomu skryptów
      - Ustawianie na nich NPC'ów, wrogów itp..
      - blokowanie przejścia dla NPC'ów
    - Node'y wyznaczają typowe zachowania NPC'ów

# Pathfinding



# Zachowania AI

- Zachowania NPC (non-player character)
  - Predefiniowane
  - Generyczne

# Zachowania AI

- Predefiniowane
  - Jako prace na Node'ach
    - Głoszenie przemowy
    - S przątanie
    - Zebranie
    - Itp..

# Zachowania AI

## ■ Generyczne

- Sytuacyjne w odniesieniu do
  - Otoczenia
    - Rzucenie pieniążka bezdomnemu
    - Klaskanie lub taniec przy grajku
    - Kupowanie przy straganach
  - Gracza
    - Ucieczka gdy gracz atakuje
    - Pokazywanie palcem jeśli gracz się wspina
- W trakcie walki
  - Przy ucieczce
  - W odniesieniu do obiektów fizycznych

# Walka

## ■ Założenia

- Każdy typ przeciwnika posiada indywidualne zachowanie
- Walka indywidualna i grupowa powinna być dla gracza wyzwaniem
- Wrogowie muszą posiadać indywidualne reakcje na inne typy przeciwników znajdujące się w pobliżu
- Na złożenie AI mamy 3 miesiące

# System walki

- S pecyfikacje S ystemu
  - Brain przeciwnika zapisany jest jako graph
  - Każda akcja może być połączona za pomocą linków z dowolną ilością innych akcji.
  - Decyzje podejmowane są w grafie zachowań według:
    - Triggerów i priorytetów

# System walki

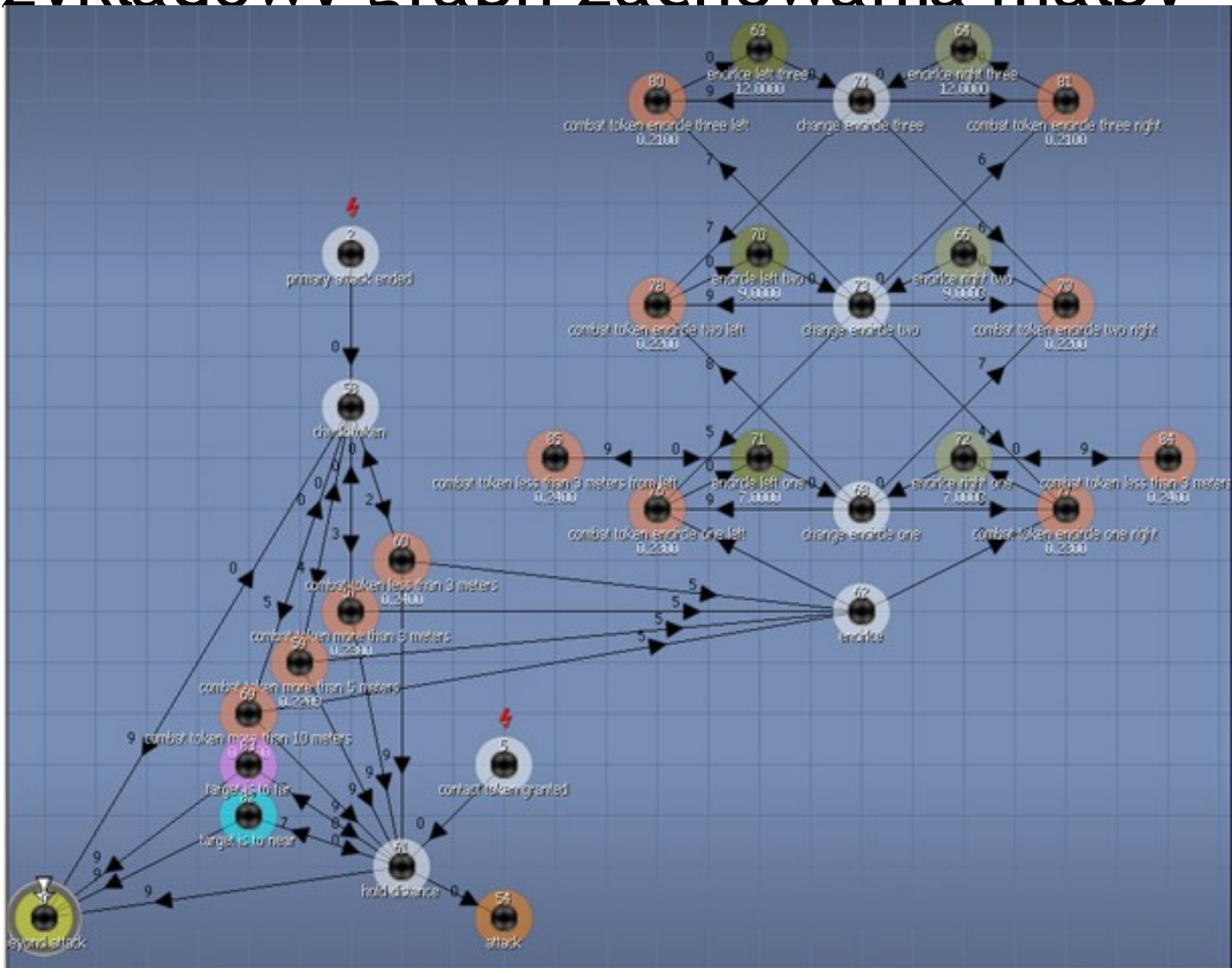
- Action (rep. Jako Node)
  - Attack
  - encircleTarget
  - decreasedDistance
- Triggery (rep. Jako Link)
  - dayPeriod
  - warriorHealthRate
  - warriorDistanceToSpawnPoint

# System walki

- Algorytm podejmowania decyzji:
  - Sprawdzane są wszystkie warunki (triggery) i układane w przypadkowej kolejności w liście
  - Triggery odchodzące od danej akcji są sortowane wg priorytetów (jeśli jest spełnione > niż 1 warunek)
  - Jeśli nadal mamy więcej niż 1 link przejściowy to wybierany jest pierwszy z listy

# System walki

■ P T Z V K I a U O W V Z T a D H Z a C T O W a H H a H a U O V



# System walki



# System walki



# System walki

- Projektowanie walki grupowej
  - walka grupowa jest oparta na przydzielaniu „kawałków tortu” (slotów od 0.000-1.000).
- Wielkość kawałków które mogą być przydzielone ustala Projektant.

# System walki

## ■ Działanie:

- Każda postać ma wokół siebie ograniczoną przestrzeń do prowadzenia walki - tzw. ilość „slotów” wyrażoną w  $[0..1]$
- Każda postać atakująca potrzebuje określonej ilości „slotów” dookoła swojego celu do przeprowadzenia ataku (np. potężne ogry więcej niż gobliny)
- Jeśli atakujący „nie mieści się” dookoła celu, zaczyna aktywne oczekiwanie na zwolnienie się slotów. W tym czasie może prowadzić działania wspomagające jak np. leczenie

# System walki

- Przykładowe zachowanie Małpy:
  - podbiega i atakuje
  - jeśli zostanie zaatakowana ucieka i z dystansu atakuje bananem
  - jeśli jedna umrze wszystkie uciekają i czekają chwilę aż im się znowu podwyższą „morale” poczym wracają i dalej atakują

# System walki

- Przykładowe zachowanie Fat Bob:
  - Pluje na gracza trucizna
  - Jeśli gracz zaatakuje go mieczem (normalne uderzenie) to pęka - a z jego brzucha wyskakują 2 tasieńce które walczą z bohaterem.
  - Jeśli nie chcemy walczyć z jego wnętrznościami to atakujemy z wysoku unikając trafień w brzuch

# System walki

## ■ Uniqu

- Składanie złożonych zachowań w parę minut
- Szybkie zlokalizowanie i naprawienie Bugów
- Większość błędów można wyłapać w edytorze
- Na walkę wpływa wiele czynników jak pora dnia, ilość przeciwników, broń użyta przez gracza, morale itd.

# Podsumowanie

For more information contact me

[Disordermail@gmail.com](mailto:Disordermail@gmail.com)

[hello@drobot.org](mailto:hello@drobot.org)

Slides will be available at

[Drobot.org](http://Drobot.org)

# Questions

---

