

Specyfikacja Dean's Paranoia

Wojciech Twarowski, Adam Ropelewski, Dawid Rychlik

15 listopada 2023

Spis treści

1	Czym jest Dean's Paranoia	2
1.1	Definicja	2
1.2	Fabula	2
1.3	Cel gry	2
2	Elementy gry	3
2.1	Niezbędne	3
2.2	Opcjonalne	4
3	Wybrane technologie	5
3.1	Silnik i technologie	5
3.2	Narzędzia poboczne	5
4	Strona wizualna	5
5	Konwencje	6
5.1	Nazewnictwo	6
5.2	Umieszczanie plików	6
5.3	Branche	6
5.4	Commity	6



Rysunek 1: propozycja logotypu

1 Czym jest Dean's Paranoia

1.1 Definicja

Dean's Paranoia to gra komputerowa multiplayer, dalej nazywana Grą. Jej fabuła jest powiązana z Wydziałem Zastosowań Informatyki i Matematyki w Szkole Głównej Gospodarstwa Wiejskiego, natomiast mechanika inspirowana grą The Matriarch. Można w nią grać na urządzeniach z systemem operacyjnym Windows, które są w tej samej sieci - poprzez połączenie peer-to-peer.

1.2 Fabuła

Na wydziale Zastosowań Informatyki i Matematyki mogą przetrwać tylko dwa rodzaje studentów - ci, którzy mieszkają w bibliotece oraz ci, którzy wykorzystają każdą okazję do złamania regulaminu, snując niesamowite intrygi w rozbudowanej siatce przestępczości studenckiej. Przez nieuważnych pierwszaków dziekan zaczyna jednak coś podejrzewać... Nie jest to bynajmniej zwątpienie wobec programu nauczania, lecz trop oszustwa na miarę historii: przebiegli studenci chcą włamać się do serwera z odpowiedziami do kolokwii! Konsekwencje dla poziomu zdawalności (oraz dochodu z warunków) są niewyobrażalne. Czy Pan Profesor w fioletowej todze ze złotym łańcuchem na kłacie złapie spiskujących, zanim będzie za późno?

1.3 Cel gry

Cel gracza zależy od wybranej postaci - może być dziekanem albo studentem.

- **Dziekan:** jego celem jest ustalenie, którzy studenci są graczami, a którzy botami. Odróżnienia można dokonać na podstawie zachowania postaci: boty wykonują przez każdą rundę (czas między spotkaniami porządkowymi na auli) te same schematyczne czynności, tymczasem gracze w celu wygrania rozgrywki muszą wykonywać również inne zadania. Studenta może wyeliminować na każdym spotkaniu porządkowym (jednego na spotkanie, a jeśli wykorzysta jednorazowy alarm przeciwpożarowy, to trzech) oraz jeżeli złapie go poza obszarami wykonywania schematycznych czynności przez boty.
- **Student:** jego celem jest znalezienie wszystkich cyfr kodu do terminali, które odblokowują dostęp do serwera z odpowiedziami. Są one ukryte w pokojach na trzecim piętrze, w miejscach, gdzie dziekan może złapać studentów poza obszarami wykonywania czynności. Po znalezieniu cyfr, student musi przekopać się przez przeszkody na czwartym piętrze i wpisać kod, cały czas będąc uznawanym za bota przez dziekana na spotkaniach porządkowych.

2 Elementy gry

2.1 Niezbędne

- **Postać dziekana:** może poruszać się po trzecim piętrze, łapać na nim studentów będących poza obszarem wykonywania zadań, a także zaznaczać w tablicie który student jest botem, a który graczem. Jeżeli alarm przeciwpożarowy nie został jeszcze wykorzystany podczas danej rozgrywki ani sabotowany, może go użyć - wówczas nastychmiast zbiera się spotkanie porządkowe, na którym musi wykreślić trzech studentów. W normalnych okolicznościach spotkanie porządkowe odbywa się co regularną ilość czasu i na każdym może wykreślić tylko jednego studenta.
- **Postać studenta:** może poruszać się po trzecim piętrze, udając że jest botem (wykonując ich zadania), zbierając cyfry do serwera poza obszarami. Ponadto może zebrać narzędzie do szybszego odkopywania gratów (jeden raz w ciągu rozgrywki; narzędzie zostaje na stałe przypisane do danego studenta), a także sabotować alarm przeciwpożarowy, jeśli nie został jeszcze wykorzystany ani dotąd sabotowany. Może również skorzystać z windy na trzecim piętrze, przechodząc w ten sposób na piętro czwarte, gdzie przekopuje się przez graty do terminali; może ustawić w każdym terminalu jedną cyfrę, które razem odblokowują dostęp do serwera. Może też podejść do serwera i sprawdzić, czy wprowadzony kod jest poprawny. Na czwartym piętrze również są windy umożliwiające powrót na piętro trzecie. Podczas spotkania porządkowego student nie ma możliwości wykonania czegokolwiek.
- **Sterowanie klawiaturą:** gracze mają możliwość poruszania się w czterech podstawowych kierunkach oraz po skosie za pomocą klawiszy W, A,

S, D. Każdy kierunek poruszania się ma odpowiednią animację. Dodatkowo, dziekan za pomocą SPACJI może złapać studenta poza obszarem wykonywania zadań albo wykorzystać alarm przeciwpożarowy, a student za pomocą SPACJI może przekopywać się przez graty, podnieść narzędzie przyspieszające kopanie, skorzystać z windy, zmienić numer w terminalu, spróbować odblokować serwer albo sabotować alarm przeciwpożarowy. Ponadto dziekan za pomocą TAB może otworzyć tablet do notowania spostrzeżeń o studentach i botach. Zadania dla botów i studentów na trzecim piętrze będą wykonywane za pomocą przycisku E

- **Piętro trzecie, zadania, alarm, narzędzie do kopania, windy:** jest inspirowane piętrem Wydziału Zastosowań Informatyki i Matematyki. Zawiera obszary do wykonywania minimum trzech zadań, takie jak automat z jedzeniem czy sala lekcyjna, a także alarm przeciwpożarowy, narzędzie do szybszego kopania oraz miejsca z cyframi do terminali. Znajdują się na nim windy na piętro czwarte.
- **Piętro czwarte, graty, terminale, serwer, windy:** na to piętro może wejść jedynie gracz, który jest studentem. Jest ono pokryte gratami, przez które należy się przekopać w celu ustawienia kodów w terminalach. Znajduje się na nim również serwer - można przy nim podjąć próbę jego odblokowania, co zajmuje określoną ilość czasu w celu utrudnienia ataków brute-force. Znajdują się na nim windy na piętro trzecie.
- **Aula:** odbywają się na niej regularnie spotkania porządkowe. Jeżeli dziekan użyje alarmu przeciwpożarowego, spotkanie porządkowe jest zwoływane natychmiast. Studenci jedynie obserwują spotkania na niej, a dziekan wybiera studentów do skreślenia.
- **Menu gry:** do wpisania adresu serwera lokalnego, do którego gracz chce się połączyć, ustawienia roli w grze oraz paramterów rozgrywki typu czas rundy, a także rozpoczęcia rozgrywki.

2.2 Opcjonalne

- **Efekty dźwiękowe**
- **Czat dla studentów:** można na nim się komunikować, (choć główna komunikacja odbędzie się na żywo skoro jest to gra po sieci lokalnej) a także zapisywać na nim zdobyte cyfry.
- **Informacja o położeniu poza obszarem wykonywania zadań:** będzie sygnalizowana studentowi poza obszarem oraz dziekanowi nad postacią studenta, który znajduje się poza obszarem.
- **Ustawienia gry:** do wybrania takich opcji jak odstęp między spotkaniami porządkowymi czy czas kopania jednego pola gratów.
- **Animacja chodzenia po skosie**

- **Strzałka pokazująca studentom przybliżone położenie dziekana**
- **Minimapa**
- **Interakcja dla skreślonych studentów:** mogą poruszać się po mapie w celu informowania pozostałych graczy o dokładnym położeniu dziekana, a także bardzo wolno przekopywać graty oraz wykonywać telefony spowalniające dziekana. Skreśleni studenci nie widzą cyfr potrzebnych do odblokowania terminali.
- **Prymus:** student-gracz, którego celem jest doprowadzenie do wykreślenia wszystkich pozostałych studentów-graczy przed nim, zanim uda im się odblokować serwer. Może sabotować rozgrywkę poprzez blokowanie wind oraz podawanie błędnych cyfr na czacie - nikt nie wie kto jest prymusem poza nim samym. Podczas każdego spotkania porządkowego studenci mogą zagłosować kto jest prymusem i również go skreślić z listy.
- **Minigry:** podczas wykonywania zadań na trzecim piętrze.

3 Wybrane technologie

3.1 Silnik i technologie

- Silnik: Godot 4.1
- Język programowania: GD Script
- Połączenie Peer to Peer (P2P)
- Framework do testów: GUnit4

3.2 Narzędzia poboczne

- Nifty - Project Management
- Discord - Komunikator
- Git & GitHub - Kontrola Wersji
- Visual Paradigm - Diagramy UML
- ProjectLibre - Wykres Gantta

4 Strona wizualna

Gra 2D z widokiem z góry (top down) - rzut izometryczny

5 Konwencje

5.1 Nazewnictwo

- Nazwy skryptów, scen, testów - snake_case
- Nazwy Node - PascalCase

5.2 Umieszczanie plików

- Skrypty - folder scripts
- Sceny - folder scenes
- Grafika - folder assets

5.3 Branche

Będziemy korzystać z tzw. feature branchów, tzn że każda klasa główna będzie mieć swój własny branch nazwany tak samo jak klasa

5.4 Commity

Każdy commit powinien być zatytułowany w języku angielskim, powinien dotyczyć pojedynczej funkcjonalności oraz być na odpowiednim branchu.