

Raport z wykrywania treści AI

Wyniki oparte są na przewidywaniu czy tekst został utworzony przy użyciu narzędzi sztucznej inteligencji opartych na GPT tj. GPT-2, GPT-J, GPT-NEO, GPT-3 lub GPT-4, Przykładowe narzędzia to ChatGPT, Bing, Gemini, Jasper..

Średnie prawdopodobieństwo wykorzystania AI

68%

Algorytm określa prawdopodobieństwo wykorzystania AI dla poszczególnych fragmentów dokumentu. Ogólny wskaźnik przedstawia średnie prawdopodobieństwo wykorzystania AI dla całego dokumentu. Wynik 90% należy traktować jako 90% pewności, że treść została stworzona przy użyciu narzędzi opartych o AI.

Na tekście zaznaczono prawdopodobieństwo wykorzystania AI dla poszczególnych fragmentów dokumentu i pogrupowano je w 5 zakresów zgodnie z legendą.

- 0 - 20%
- 21 - 40%
- 41 - 60%
- 61 - 80%
- 81 - 100%

Zawartość treści AI w tekście

29%

Wskaźnik określa jaka część tekstu została zakwalifikowana jako treść wytworzona przy użyciu narzędzi opartych na AI. Za treści wytworzone przez AI system uznaje fragmenty, dla których prawdopodobieństwo wykorzystania AI przekracza wartość progową, domyślnie 0.8 (w skali od 0 do 1). Wynik 90% oznacza, że 90% tekstu zostało stworzone przez sztuczną inteligencję.

Szczegóły

W tabeli poniżej znajdują się fragmenty, uszeregowane od tych, które najprawdopodobniej zostały wygenerowane przez model językowy AI, do tych o najmniejszym prawdopodobieństwie.

LP	FRAGMENT	WYKRYWANIE TREŚCI AI	LICZBA SŁÓW
1	Tworzenie dynamicznych tekstów wymaga wielu kroków...	94%	261
2	Unity, choć bardzo elastyczne, wymaga od użytkowni...	91%	59
3	Narzędzie to zostało zaprojektowane z myślą o pros...	74%	17
4	Z kolei w Unreal Engine brakuje wbudowanego system...	71%	15
5	Wstęp Dynamiczne kampanie w grach komputerowych st...	69%	37
6	Brak zaawansowanych mechanizmów oraz elastyczności...	69%	53
7	Układ pisemnej pracy został zoptymalizowany dzięki...	68%	64
8	Ostatnim narzędziem jest Unreal Engine, który jest...	64%	46
9	To jest program stworzony z myślą o amatorskich tw...	64%	19
10	Niestety brak dedykowanego systemu do zarządzania ...	56%	57
11	Umożliwiają one zarządzanie interakcjami gracza, z...	55%	32
12	Pomimo elastyczności, Unity wymaga zaawansowanych ...	55%	93
13	To czyni go łatwym w obsłudze nawet dla osób bez d...	55%	36
14	Tam też decyzje podejmowane przez gracza wpływają ...	51%	323

Wstęp

Dynamiczne kampanie w grach komputerowych stały się istotnym elementem współczesnej branży gier wideo, umożliwiając graczom większą swobodę w podejmowaniu decyzji oraz wpływ na rozwój fabuły. W odróżnieniu od tradycyjnych, liniowych scenariuszy, dynamiczne kampanie oferują nieliniową narrację. Tam też decyzje podejmowane przez gracza wpływają na dalszy przebieg gry. Taka struktura fabularna zwiększa immersję, co sprawia, że gracz czuje się częścią świata przedstawionego w grze.

Jednym z kluczowych wyzwań przy tworzeniu dynamicznych kampanii jest efektywne zarządzanie interakcjami gracza oraz implementacja narzędzi wspomagających budowę nieliniowej narracji. To wymaga zastosowania odpowiednich funkcji, takich jak dynamiczne pisanie tekstów, elastyczne systemy wyborów i synchronizacja narracji z innymi elementami gry, takie jak animacje i dźwięki. Złożoność takich systemów sprawia, iż projektowanie i implementacja dynamicznych kampanii wymagają użycia zaawansowanych narzędzi deweloperskich i są bardzo czasochłonne.

Odpowiadając na te wyzwania i w ramach tej pracy inżynierskiej zostały stworzone funkcje wspomagające zrobienia dynamicznych kampanii w grach komputerowych. Te funkcje zostały zaprojektowane z myślą o ułatwieniu pracy twórcom gier poprzez uproszczenie procesu implementacji narracji jak też zwiększeniu elastyczności w dostosowywaniu mechanik gry do potrzeb konkretnego projektu. Praca ta skupia się na opracowaniu rozwiązań, które nie tylko poprawiają efektywność pracy, ale również podnoszą jakość narracji. Oferuje to graczom bogatsze jak też bardziej angażujące doświadczenia.

Praca ta opiera się głównie na funkcjach dynamicznej narracji, które pozwalają na elastyczne wyświetlanie tekstu w grze. Niestety głównym problemem w takich rozwiązaniach jest zapewnienie tej elastyczności. Tradycyjne systemy wyświetlania tekstu w grach, czy to w formie dialogów, narracji, czy też opcji wyboru. Często one tworzone są w sposób statyczny, a generuje to dodatkowe obciążenie projektu o dodatkową zawartość. Proponowane w tej pracy funkcje rozwiązują ten problem dzięki przemyślanej strukturze i elastycznym zmiennym oraz prostym w użyciu narzędziom. Pozwalają one programistom i projektantom skupić się na bardziej złożonych aspektach projektu.

Praca składa się z pięciu rozdziałów przedstawiających kolejne etapy realizacji projektu. Rozdział pierwszy i drugi prezentuje analizę możliwych silników gier jak też istniejących rozwiązań wspierających dynamiczne kampanie. Rozdział trzeci i czwarty koncentruje się na implementacji oraz projektowaniu funkcji. Przy czym czwarty rozdział szczegółowo opisuje główne funkcje poczynając od ich podstawowych składowych aż po pełne makra. Piąty, który jest ostatnim rozdziałem przedstawia proces testowania i analizę wyników testów. Układ pisemnej pracy został zoptymalizowany dzięki konsultacjom z narzędziem wspierającym redakcję treści. Pomocne było to w zachowaniu spójności strukturalnej. Proces pisania oraz merytoryczna zawartość pracy zostały ostatecznie ukształtowane bezpośrednio przez autora, który odpowiada za opracowanie koncepcji i jej realizację.

Rozdział 1 Analiza istniejących rozwiązań:

1. Charakterystyka istniejących narzędzi do tworzenia dynamicznych kampanii w grach komputerowych

Tworzenie dynamicznych kampanii w grach komputerowych wymaga odpowiednich narzędzi. Umożliwiają one zarządzanie interakcjami gracza, zmianami w rozgrywce oraz narracją. Na rynku dostępnych jest kilka takich narzędzi, które w różnym stopniu pozwalają na realizację tych zadań. Przykładem popularnych narzędzi jest RPG Maker. To jest program stworzony z myślą o amatorskich twórcach gier RPG. Umożliwia on korzystanie z gotowych szablonów i zasobów. To czyni go łatwym w obsłudze nawet dla osób bez doświadczenia w programowaniu. Ten program posiada wbudowany system wydarzeń co pozwala na tworzenie prostych narracji i interakcji. Jednak widoczne są jego ograniczenia w bardziej złożonych projektach. Brak zaawansowanych mechanizmów oraz elastyczności sprawia, że RPG Maker nadaje się głównie do tworzenia gier w jednym określonym gatunku.

Innym rozwiązaniem jest Unity, to wszechstronny silnik gier umożliwiający tworzenie dynamicznych kampanii poprzez skrypty w języku C#. Dzięki tak bogatej dokumentacji oraz szerokim możliwościom technologicznym Unity stało się jednym z najpopularniejszych narzędzi w branży. Pomimo elastyczności, Unity wymaga zaawansowanych umiejętności programistycznych. Tworzenie dynamicznych kampanii wymaga pisania wielu linii kodu jak też budowania systemów narracyjnych praktycznie od podstaw, a to znacząco wydłuża czas pracy.

Należy też wspomnieć o Twine. To narzędzie specjalizuje się w interaktywnych opowieściach. Jest proste w obsłudze i intuicyjne. To czyni je idealnym dla projektów narracyjnych, w których tekst stanowi główny element rozgrywki. Twine umożliwia łatwe tworzenie interakcji oraz wyborów gracza. Niestety jednak ogranicza się do narracji tekstowej oraz nie oferuje wsparcia dla bardziej zaawansowanych funkcji, jak efekty dźwiękowe, animacje czy też złożona mechanika gry.

Ostatnim narzędziem jest Unreal Engine, który jest jednym z najpotężniejszych silników gier, a dzięki systemowi Blueprintów pozwala na tworzenie zaawansowanych mechanik bez potrzeby pisania kodu w językach programowania. Omawiany Unreal Engine oferuje szerokie możliwości integracji dynamicznych tekstów z innymi elementami gry, jak animacje czy też dźwięki. Niestety brak dedykowanego systemu do zarządzania tekstami sprawia, iż proces ten wymaga dużej liczby manualnych kroków, a to może być czasochłonne oraz mniej intuicyjne.

2. Słabe strony analizowanych rozwiązań

Każde z omówionych narzędzi ma swoje ograniczenia, które mogą wpłynąć na efektywność tworzenia dynamicznych kampanii.

W przypadku RPG Maker główną wadą jest brak wsparcia dla bardziej zaawansowanych mechanik. Narzędzie to zostało zaprojektowane z myślą o prostych grach RPG i trudno je zastosować w innych gatunkach.

Unity, choć bardzo elastyczne, wymaga od użytkowników umiejętności programistycznych, co może stanowić barierę dla mniej doświadczonych twórców. Tworzenie dynamicznych kampanii w Unity często wymaga pisania długich i skomplikowanych skryptów, co zwiększa czas produkcji. Twine, mimo intuicyjności, jest ograniczone do narracji tekstowej i nie oferuje narzędzi pozwalających na łatwą integrację z bardziej zaawansowanymi mechanikami gier, takimi jak animacje czy dźwięki.

Z kolei w Unreal Engine brakuje wbudowanego systemu zarządzania sekwencjami tekstów oraz mechanizmów wyborów gracza. Tworzenie dynamicznych tekstów wymaga wielu kroków, takich jak ręczne budowanie widgetów i integrowanie ich z logiką gry, co może być czasochłonne.

3. Propozycje unikania słabych stron w projekcie

Unreal Engine 5 jest zaawansowanym narzędziem do tworzenia gier, oferującym nowoczesne technologie, takie jak Lumen i Nanite, które znacząco podnoszą jakość wizualną oraz wydajność. Mimo tych zalet, w kontekście tworzenia dynamicznych kampanii nadal istnieją pewne ograniczenia, które mogą wpłynąć na proces deweloperski.

Jednym z głównych wyzwań jest brak wbudowanego, dedykowanego systemu do zarządzania dynamicznymi kampaniami. Twórcy muszą samodzielnie implementować mechanizmy narracyjne, co wymaga tworzenia kompleksowych struktur w Blueprintach lub pisania kodu w języku C++. Choć system Blueprintów znacząco ułatwia budowanie logiki gry, bardziej zaawansowane mechanizmy, takie jak rozbudowane drzewa dialogowe czy dynamiczne zmiany fabularne, mogą wymagać dodatkowej pracy i optymalizacji.

W Unreal Engine 5 proces tworzenia dynamicznych tekstów i wyborów gracza wymaga wielu kroków. Twórcy muszą ręcznie tworzyć widżety, zarządzać wyświetlaniem tekstów oraz integrować je z innymi elementami gry, takimi jak animacje czy efekty dźwiękowe. Brak narzędzi do automatyzacji tych procesów wydłuża czas pracy i może być trudny dla mniej doświadczonych programistów.

Dodatkowo, zarządzanie dużymi ilościami danych, takimi jak sekwencje tekstów czy interakcje gracza, może być wyzwaniem w przypadku projektów o większej skali. Twórcy muszą zaprojektować własne systemy przechowywania danych i zarządzania nimi, co wymaga dodatkowego wysiłku. Mimo zaawansowanych technologii, takich jak Nanite i Lumen, Unreal Engine 5 nadal wymaga dużej dbałości o optymalizację w przypadku gier o dużych otwartych światach lub złożonych kampaniach dynamicznych. Zarządzanie zasobami oraz utrzymanie płynności działania gry pozostają kluczowym wyzwaniem w kontekście projektów, które wykorzystują dynamiczne zmiany fabularne.