

STRESZCZENIE

ABSTRACT

SPIS TREŚCI

1.	Wstęp i cel pracy	8
2.	Wprowadzenie do dziedziny	10
2.1.	Rola technologii w edukacji (Konrad Czarnecki)	10
2.1.1.	Nowoczesne metody nauczania matematyki	10
2.1.2.	Gamifinacka w edukacji	11
2.2.	Edukacyjne zastosowania escape roomów (Konrad Czarnecki)	11
2.2.1.	Historia i definicja escape roomu	12
2.2.2.	Escape roomy jako metoda aktywizacji uczniów	12
2.3.	Wirtualna rzeczywistość (Konrad Czarnecki)	13
2.3.1.	Wirtualna rzeczywistość i jej zastosowanie w nauce	13
2.3.2.	Wpływ interaktywności na zaangażowanie gracza	13
3.	Analiza istniejących rozwiązań	15
3.1.	Wirtualne escape roomy	15
3.2.	Przegląd gier edukacyjnych z elementami VR	15
3.2.1.	Blockerzz	15
3.2.2.	Mage Math	15
3.2.3.	EdScape XR	15
3.3.	Wnioski z analizy dostępnych rozwiązań	15
4.	Projekt systemu	16
4.1.	Wymagania funkcjonalne i niefunkcjonalne (Konrad Czarnecki)	16
4.1.1.	Wymagania funkcjonalne	16
4.1.2.	Wymagania niefunkcjonalne	17
4.2.	Scenariusz gry (Konrad Czarnecki)	17
4.3.	Zagadki (Konrad Czarnecki)	18
4.3.1.	Wyrażenia algebraiczne	18
4.3.2.	Planimetria	18
4.3.3.	Równania i nierówności	19
4.3.4.	Funkcje	19
4.3.5.	Geometria analityczna na płaszczyźnie kartezjańskiej	19
4.3.6.	Kombinatoryka	19
4.3.7.	Liczby rzeczywiste	19
4.3.8.	Trygonometria	20
4.3.9.	Ciągi	20
4.3.10.	Rachunek prawdopodobieństwa i statystyka	20
4.3.11.	Optymalizacja i rachunek różniczkowy	20
4.3.12.	Układy równań	21
4.3.13.	Stereometria	21
4.4.	Model przypadków użycia	21
4.5.	Projekt architektury systemu	21
4.6.	Projekt interfejsu użytkownika i środowiska gry	21

5.	Technologie i narzędzia	22
5.1.	Silnik gry (Konrad Czarnecki)	22
5.2.	Laboratorium Zanurzonej Wizualizacji Przestrzennej (Konrad Czarnecki)	22
5.3.	Środowisko 3D (Konrad Czarnecki)	23
5.4.	System kontroli wersji (Konrad Czarnecki)	23
6.	Analiza dydaktyczna i potencjał edukacyjny	24
6.1.	Motywacja uczniów do nauki matematyki	24
6.2.	Możliwość personalizacji i różnicowania poziomu trudności	24
6.3.	Wpływ imersji na naukę	24
7.	Opis teoretyczny zagadek	25
7.1.	Zadanie 1 – Wzory skróconego mnożenia (Autor)	25
7.1.1.	Cel zadania	25
7.1.2.	Zasady działania	25
7.1.3.	Zakończenie zadania	25
7.2.	Zadanie 2 – Planimetria (Jan Walczak)	25
7.2.1.	Cel zadania	25
7.2.2.	Zasady działania	25
7.2.3.	Zakończenie zadania	26
7.3.	Zadanie 3 – Nierówności (Konrad Czarnecki)	26
7.3.1.	Cel zadania	26
7.3.2.	Zasady działania	26
7.3.3.	Zakończenie zadania	26
7.4.	Zadanie 4 – Funkcje (Konrad Czarnecki)	27
7.4.1.	Cel zadania	27
7.4.2.	Zasady działania	27
7.4.3.	Zakończenie zadania	27
7.5.	Zadanie 5 – Geometria analityczna (Konrad Czarnecki)	27
7.5.1.	Cel zadania	27
7.5.2.	Zasady działania	27
7.5.3.	Zakończenie zadania	28
7.6.	Zadanie 6 – Kombinatoryka (Autor)	28
7.6.1.	Cel zadania	28
7.6.2.	Zasady działania	28
7.6.3.	Zakończenie zadania	28
7.7.	Zadanie 7 – Liczby rzeczywiste i działania na zbiorach liczbowych (Jan Walczak)	28
7.7.1.	Cel zadania	29
7.7.2.	Zasady działania	29
7.7.3.	Zakończenie zadania	29
7.8.	Zadanie 8 – Znaki funkcji trygonometrycznych (Konrad Czarnecki)	30
7.8.1.	Cel zadania	30
7.8.2.	Zasady działania	30
7.8.3.	Zakończenie zadania	30

8.10 Zadanie 10 – Prawdopodobieństwo (Autor)	46
8.11 Zadanie 11 – Optymalizacja i rachunek różniczkowy (Autor)	46
8.12 Zadanie 12 – Układy równań (Autor)	46
8.13 Zadanie 13 – Stereometria (Jan Walczak)	46
8.13.1. Problemy i różnice w realizacji zadania w praktyce	46
8.13.2. Implementacja zadania	46
8.13.3. Przebieg zadania	47
9. Podsumowanie	49
9.1. Ocena realizacji celów	49
9.2. Propozycje rozwoju projektu	49
9.3. Wnioski końcowe	49
Wykaz literatury	50
Spis rysunków	50
Spis tabel	51

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

Escape room – interaktywna gra, w której gracze rozwiązuje zagadki, by „uciec” z zamkniętego pokoju lub przejść do kolejnego etapu.

VR – wirtualna rzeczywistość, technologia pozwalająca na zanurzenie się w cyfrowym świecie za pomocą gogli i kontrolerów.

Laboratorium Zanurzonej Wizualizacji Przestrzennej – specjalne środowisko lub przestrzeń, gdzie łączy się elementy rzeczywistości wirtualnej i rzeczywistości rzeczywistej, często wykorzystywane do edukacji i interaktywnych doświadczeń.

1. WSTĘP I CEL PRACY

Współczesny rozwój technologii informatycznych i multimedialnych umożliwia tworzenie nowoczesnych, interaktywnych narzędzi wspomagających proces nauczania, które pozwalają uczniom nie tylko na bierne przyswajanie wiedzy, ale również na aktywne uczestnictwo w zajęciach oraz samodzielne rozwiązywanie problemów w angażującym środowisku. W szczególności dynamicznie rozwijające się technologie wirtualnej i rozszerzonej rzeczywistości znajdują coraz szersze zastosowanie nie tylko w przemyśle i rozrywce, lecz również w edukacji na różnych poziomach nauczania. Zastosowanie immersyjnych środowisk w nauczaniu matematyki może znacząco wpływać na wzrost zaangażowania uczniów oraz efektywność przyswajania materiału dydaktycznego.

Matematyka, jako jedna z podstawowych dziedzin nauki, często bywa postrzegana przez uczniów szkół średnich jako przedmiot trudny i mało atrakcyjny. Jednym ze sposobów przełamywania tego stereotypu jest wprowadzenie do procesu dydaktycznego elementów gier edukacyjnych, które poprzez zabawę i rywalizację angażują uczniów do aktywnego rozwiązywania problemów. W ostatnich latach coraz większym zainteresowaniem cieszą się tzw. escape roomy — gry logiczne, w których uczestnicy muszą w określonym czasie rozwiązać serię zagadek i łamigłówek, aby „wydostać się” z wirtualnego lub rzeczywistego pomieszczenia. Włączenie tego typu rozwiązań do nauczania matematyki stwarza szansę na połączenie nauki z zabawą oraz zastosowanie wiedzy teoretycznej w praktycznych sytuacjach problemowych.

Przedmiotem niniejszej pracy jest opracowanie i wykonanie aplikacji stanowiącej wirtualny pokój zagadek matematycznych (ang. escape room), przeznaczonej do działania w systemie jaskiń rzeczywistości wirtualnej dostępnych w Laboratorium Zanurzonej Wizualizacji Przestrzennej (LZWP). Laboratorium to wyposażone jest w nowoczesne rozwiązania umożliwiające projekcję obrazów w technologii VR na ścianach specjalnie przystosowanych pomieszczeń, dając użytkownikowi wrażenie całkowitego zanurzenia w trójwymiarowym środowisku. Aplikacja powinna umożliwiać uczestnikom interakcję z otoczeniem za pomocą kontrolerów ruchu wykorzystywanych w LZWP.

Celem głównym projektu jest stworzenie interaktywnej aplikacji edukacyjnej zawierającej 13 zagadek matematycznych, odpowiadających poszczególnym działom matematyki nauczany w szkołach średnich. Działy te obejmują: liczby rzeczywiste, wyrażenia algebraiczne, równania i nierówności, układy równań, funkcje, ciągi, trygonometrię, planimetrię, geometrię analityczną na płaszczyźnie kartezjańskiej, stereometrię, kombinatorykę, rachunek prawdopodobieństwa i statystykę oraz optymalizację i rachunek różniczkowy.

W ramach realizacji projektu przewidziano następujące etapy prac:

1. Zapoznanie się z funkcjonowaniem systemu jaskiń rzeczywistości wirtualnej dostępnych w LZWP oraz metodami programowania aplikacji dla tego typu środowisk.
2. Opracowanie koncepcji i projektów zagadek matematycznych reprezentujących wymienione działy matematyki, z uwzględnieniem ich formy, poziomu trudności oraz możliwych metod interakcji użytkownika z aplikacją.
3. Konsultacja opracowanych projektów zagadek z dydaktykami matematyki w celu dostosowania ich treści oraz sposobu prezentacji do wymagań programowych i poziomu uczniów szkół średnich.

4. Implementacja aplikacji integrującej wszystkie zaprojektowane i zatwierdzone zagadki w ramach jednego spójnego środowiska wirtualnego escape roomu.

5. Przeprowadzenie testów funkcjonalnych, wydajnościowych oraz badań pilotażowych z udziałem grupy uczniów i dydaktyków w celu weryfikacji poprawności działania aplikacji oraz jej efektywności dydaktycznej.

Ostatecznym rezultatem pracy będzie kompletna, działająca aplikacja edukacyjna przeznaczona do uruchomienia w jaskiniach rzeczywistości wirtualnej, umożliwiająca użytkownikom rozwijanie kompetencji matematycznych w angażującej i nowoczesnej formie. Wnioski płynące z przeprowadzonych badań pilotażowych pozwolą na ocenę przydatności tego typu rozwiązań w praktyce dydaktycznej oraz wskazanie możliwości ich dalszego rozwoju i wykorzystania w nauczaniu przedmiotów ścisłych.

2. WPROWADZENIE DO DZIEDZINY

2.1. Rola technologii w edukacji (Konrad Czarnecki)

Rozwój technologii cyfrowych wywarł ogromny wpływ na niemal wszystkie obszary współczesnego życia, w tym również na edukację. Tradycyjne metody nauczania, oparte w dużej mierze na wykładzie i pracy z podręcznikiem, są coraz częściej wzbogacane lub nawet zastępowane przez nowoczesne narzędzia dydaktyczne, które wspomagają i uatrakcyjnają proces kształcenia. W szczególności technologie informacyjno-komunikacyjne oraz środowiska immersywne, takie jak rzeczywistość wirtualna, rzeczywistość rozszerzona czy rzeczywistość mieszana, stają się istotnym elementem nowoczesnej edukacji.

Współczesne pokolenia uczniów i studentów od najmłodszych lat funkcjonują w środowisku przesyconym technologią i są przyzwyczajeni do interaktywnego oraz multimedialnego przekazu treści. Tradycyjny model nauczania nie zawsze jest w stanie zaspokoić potrzeby poznawcze młodych ludzi, co może prowadzić do spadku motywacji oraz trudności w przyswajaniu wiedzy. Odpowiedzią na to wyzwanie jest integracja narzędzi technologicznych z procesem dydaktycznym, co może przyczynić się do zwiększenia efektywności nauczania, ułatwienia zrozumienia trudnych zagadnień oraz podniesienia ogólnego poziomu zaangażowania uczniów.

Zastosowanie nowoczesnych technologii w edukacji nie tylko wpływa na sposób przekazywania wiedzy, ale również umożliwia tworzenie zupełnie nowych, interaktywnych form kształcenia. Szczególnie istotne znaczenie mają tutaj rozwiązania wykorzystujące elementy gamifikacji oraz immersyjnych doświadczeń edukacyjnych, które pozwalają uczniom wchodzić w bezpośrednią interakcję z materiałem dydaktycznym. W kontekście nauczania matematyki, będącej często postrzeganą jako przedmiot trudny i abstrakcyjny, technologie te mogą odegrać kluczową rolę w budowaniu pozytywnego nastawienia do nauki oraz lepszego zrozumienia prezentowanych treści.

2.1.1. Nowoczesne metody nauczania matematyki

Matematyka, jako dziedzina wymagająca myślenia analitycznego, logicznego rozumowania oraz umiejętności abstrakcyjnego operowania symbolami, od zawsze stawała przed uczniami szczególnie wyzwania. Z tego powodu nauczanie matematyki wymaga nieustannego poszukiwania skutecznych metod dydaktycznych, które nie tylko umożliwiają efektywne przekazanie wiedzy, ale również wzbudzą w uczniach zainteresowanie i motywację do nauki.

Współczesne podejścia do nauczania matematyki coraz częściej odchodzą od modelu opartego wyłącznie na wykładzie i ćwiczeniach przy tablicy, na rzecz metod aktywizujących, w których uczniowie samodzielnie odkrywają zależności matematyczne, rozwiązują problemy oraz współpracują w grupie. Szczególne miejsce wśród nowoczesnych metod zajmują rozwiązania oparte na technologiach komputerowych — aplikacje i platformy edukacyjne, programy do wizualizacji danych matematycznych, a także symulacje i środowiska interaktywne.

Zastosowanie narzędzi interaktywnych pozwala uczniom na dynamiczne eksplorowanie pojęć matematycznych, eksperymentowanie z danymi i obserwowanie skutków wprowadzanych zmian w czasie rzeczywistym. Programy takie jak GeoGebra, Desmos, czy MATLAB wspierają nauczanie

geometrii, analizy matematycznej i algebry w sposób znacznie bardziej przystępny i angażujący niż tradycyjne metody.

Również technologie immersywne, takie jak VR, zyskują coraz większe znaczenie w edukacji matematycznej. Umożliwiają one prezentację skomplikowanych struktur geometrycznych w przestrzeni trójwymiarowej, co szczególnie sprzyja nauczaniu stereometrii czy geometrii analitycznej. Dzięki temu uczniowie mogą lepiej zrozumieć zależności przestrzenne oraz intuicyjnie postrzegać abstrakcyjne pojęcia matematyczne.

2.1.2. Gamifinacka w edukacji

Gamifikacja (*ang. gamification*) to zastosowanie mechanizmów znanych z gier komputerowych i planszowych w kontekście niezwiązanym bezpośrednio z grami, takim jak edukacja, zarządzanie czy marketing. W praktyce oznacza to wprowadzanie elementów takich jak punkty, poziomy trudności, nagrody, rankingi czy wyzwania do tradycyjnych zadań edukacyjnych, co ma na celu zwiększenie motywacji, zaangażowania i satysfakcji uczestników procesu nauczania.

W edukacji gamifikacja znajduje szerokie zastosowanie, zwłaszcza w pracy z uczniami szkół podstawowych i średnich. Wprowadzenie grywalizacji do lekcji pozwala uczniom uczestniczyć w nauce w sposób bardziej aktywny i emocjonalnie zaangażowany. Zamiast biernego słuchania wykładu czy rozwiązywania zadań z podręcznika, uczniowie mogą wcielać się w bohaterów gier, zdobywać osiągnięcia i rywalizować z rówieśnikami w przyjazny sposób.

W kontekście nauczania matematyki, gamifikacja może znaczco ułatwić przyswajanie skomplikowanych treści. Zagadki logiczne, łamigłówki, quizy punktowane czy interaktywne escape roomy są przykładami narzędzi, które pozwalają uczniom na wykorzystanie wiedzy matematycznej w kontekście gry. Dzięki temu uczniowie nie tylko uczą się rozwiązywać konkretne typy zadań, ale także rozwijają umiejętności analitycznego myślenia, pracy zespołowej oraz podejmowania decyzji.

W szczególności w środowiskach immersyjnych, takich jak wirtualna rzeczywistość, gamifikacja osiąga nowy wymiar. Połączenie mechanizmów gry z wciągającym, interaktywnym środowiskiem pozwala użytkownikom na budowanie trwałych, pozytywnych skojarzeń z procesem nauki. Tego typu rozwiązania nie tylko zwiększają atrakcyjność zajęć, ale również mogą pozytywnie wpływać na wyniki edukacyjne i długofalowe postawy wobec nauki matematyki.

2.2. Edukacyjne zastosowania escape roomów (Konrad Czarnecki)

W ostatnich latach obserwuje się dynamiczny rozwój innowacyjnych metod dydaktycznych, które mają na celu zwiększenie zaangażowania uczniów oraz poprawę efektywności procesu nauczania. Jedną z takich metod są escape roomy, które pierwotnie funkcjonowały jako forma rozrywki, a z czasem zyskały również uznanie w środowisku edukacyjnym. Dzięki swojej interaktywnej i zespołowej formie, pokoje zagadek umożliwiają uczniom zdobywanie wiedzy oraz rozwijanie kompetencji miękkich w sposób atrakcyjny i emocjonujący.

Escape roomy w edukacji mogą przybierać różnorodne formy — od tradycyjnych wersji stacjonarnych, przez mobilne zestawy dydaktyczne, aż po aplikacje komputerowe i środowiska wirtualnej rzeczywistości. Ich celem jest nie tylko przekazywanie wiedzy merytorycznej, lecz także kształcenie umiejętności pracy zespołowej, logicznego myślenia, zarządzania czasem oraz podejmowania decyzji pod presją. Dlatego coraz częściej wykorzystywane są one w szkołach, na uczelniach wyższych oraz podczas szkoleń i warsztatów.

W kontekście nauczania matematyki escape roomy stanowią wyjątkowo wartościowe narzędzie. Zagadki oparte na treściach matematycznych wymagają od uczestników nie tylko opanowania materiału, ale również zastosowania wiedzy w praktyce, co sprzyja utrwalaniu wiadomości oraz rozwijaniu umiejętności rozwiązywania problemów. Połączenie elementów gry z edukacją umożliwia uczniom naukę w przyjaznej atmosferze i sprzyja budowaniu pozytywnego nastawienia do przedmiotów ścisłych.

2.2.1. Historia i definicja escape roomu

Escape room, znany również jako pokój zagadek lub gra typu „ucieczka z pokoju”, to interaktywna forma rozrywki polegająca na rozwiązywaniu serii łamigłówek i zadań logicznych w określonym czasie, aby wydostać się z zamkniętego pomieszczenia lub osiągnąć inny wyznaczony cel fabularny.

Pierwszy fizyczny escape room powstał w 2007 roku w Kioto w Japonii, a jego twórcą był Takao Kato, który postanowił przenieść ideę wirtualnej gry do świata rzeczywistego. Projekt szybko zyskał popularność, a w kolejnych latach podobne pokoje zaczęły powstawać w innych krajach azjatyckich, a następnie w Europie i Ameryce Północnej. Do Polski pierwsze escape roomy dotarły w 2014 roku i od tego czasu cieszą się dużym zainteresowaniem zarówno wśród młodzieży, jak i dorosłych.

Z czasem, obok komercyjnych pokoi zagadek, zaczęły pojawiać się również wersje edukacyjne, dostosowane do potrzeb szkół i uczelni. Edukacyjne escape roomy różnią się od wersji rozrywkowych tym, że ich głównym celem nie jest zabawa, lecz przekazanie wiedzy oraz rozwijanie określonych kompetencji. Zagadki w tego typu pokojach są projektowane w taki sposób, aby uczestnicy mogli przyswajać treści z wybranych dziedzin nauki podczas rozwiązywania interaktywnych zadań. Coraz częściej wykorzystywane są one także w środowiskach cyfrowych oraz w wirtualnej rzeczywistości, co dodatkowo zwiększa ich dostępność i atrakcyjność.

2.2.2. Escape roomy jako metoda aktywizacji uczniów

Jednym z największych wyzwań współczesnej edukacji jest utrzymanie wysokiego poziomu zaangażowania uczniów oraz motywowanie ich do aktywnego uczestnictwa w zajęciach. W tym kontekście escape roomy stanowią skutecną metodę dydaktyczną, która pozwala na połączenie nauki z emocjonującą formą zabawy. Dzięki swojej interaktywnej strukturze i zespołowemu charakterowi, pokoje zagadek mobilizują uczestników do współpracy, komunikacji oraz kreatywnego rozwiązywania problemów.

Z punktu widzenia dydaktyki escape roomy wspierają rozwój kompetencji kluczowych, takich jak logiczne myślenie, umiejętność analizowania i syntetyzowania informacji, podejmowanie decyzji pod presją czasu oraz zarządzanie zadaniami w zespole. Dodatkowo umożliwiają uczniom zastosowanie zdobytej wiedzy teoretycznej w praktyce, co znacząco wpływa na trwałość zapamiętywania materiału i lepsze zrozumienie omawianych treści.

W szczególności istotne znaczenie mają escape roomy realizowane w środowisku rzeczywistości wirtualnej. Dzięki immersijnemu charakterowi takich aplikacji uczestnicy mogą zanurzyć się w wirtualnym świecie, w którym zagadki matematyczne są częścią spójnej fabuły. Tego typu doświadczenie nie tylko zwiększa atrakcyjność nauki, ale również wzmacnia motywację wewnętrzną uczniów, którzy postrzegają naukę jako przygodę i wyzwanie, a nie obowiązek.

2.3. Wirtualna rzeczywistość (Konrad Czarnecki)

Współczesna edukacja coraz śmielej sięga po nowoczesne technologie, które pozwalają nie tylko wzbogacić proces nauczania, ale również znaczowo zwiększyć zaangażowanie uczniów. Jednym z najbardziej dynamicznie rozwijających się obszarów w tym zakresie są technologie immersywne, do których zalicza się wirtualną rzeczywistość, rzeczywistość rozszerzoną oraz rzeczywistość mieszaną. Ich wspólną cechą jest zdolność do tworzenia interaktywnych środowisk, w których użytkownik ma wrażenie pełnego zanurzenia w generowanym komputerowo świecie.

W kontekście edukacyjnym szczególnie istotne jest połączenie immersji z interaktywnością, czyli możliwością bezpośredniego wpływu na elementy wirtualnego środowiska. Interaktywne aplikacje VR sprzyjają aktywnej nauce poprzez angażowanie użytkownika w proces rozwiązywania problemów, podejmowania decyzji i wykonywania zadań w czasie rzeczywistym. Taka forma edukacji nie tylko zwiększa efektywność przyswajania wiedzy, ale również pozytywnie wpływa na motywację i postawy uczniów wobec nauki.

2.3.1. Wirtualna rzeczywistość i jej zastosowanie w nauce

Wirtualna rzeczywistość to technologia umożliwiająca tworzenie komputerowo generowanych środowisk trójwymiarowych, z którymi użytkownik może wchodzić w interakcję w czasie rzeczywistym. Dzięki zastosowaniu gogli VR oraz kontrolerów ruchu możliwe jest odwzorowanie ruchów użytkownika w przestrzeni wirtualnej, co pozwala na pełne zanurzenie w symulowanym środowisku. Technologie VR znajdują zastosowanie nie tylko w rozrywce, ale również w przemyśle, medycynie, wojsku oraz, coraz częściej, w edukacji.

W środowisku edukacyjnym wirtualna rzeczywistość oferuje szerokie możliwości w zakresie tworzenia interaktywnych laboratoriów, symulacji zjawisk fizycznych, rekonstrukcji historycznych czy wirtualnych wycieczek. Uczniowie mogą dzięki niej eksplorować trudno dostępne miejsca, takie jak wnętrze ludzkiego organizmu, przestrzeń kosmiczną, odległe zakątki świata lub historyczne budowle, co znacząco wzbogaca tradycyjny proces dydaktyczny.

W przypadku dydaktyki matematyki VR pozwala na wizualizację skomplikowanych zagadnień geometrycznych, przestrzennych oraz analitycznych w atrakcyjnej i przystępnej formie. Uczniowie mogą w wirtualnym środowisku manipulować bryłami, obserwować zmiany funkcji w czasie rzeczywistym czy uczestniczyć w interaktywnych grach logicznych i escape roomach matematycznych. Dzięki temu abstrakcyjne treści stają się bardziej zrozumiałe i łatwiejsze do przyswojenia, co przekłada się na lepsze wyniki w nauce oraz pozytywnie wpływa na rozwój umiejętności analitycznego myślenia.

2.3.2. Wpływ interaktywności na zaangażowanie gracza

Interaktywność stanowi jeden z kluczowych elementów nowoczesnych metod dydaktycznych, w tym również aplikacji wykorzystujących technologię wirtualnej rzeczywistości. Oznacza ona możliwość aktywnego uczestniczenia ucznia w procesie dydaktycznym poprzez bezpośrednie oddziaływanie na środowisko edukacyjne, podejmowanie decyzji oraz realizację zadań w czasie rzeczywistym. W przeciwieństwie do pasywnego odbioru treści w tradycyjnych formach nauczania, interaktywne środowiska angażują uczniów na wielu płaszczyznach — poznawczej, emocjonalnej i motorycznej.

Z perspektywy pedagogicznej interaktywność sprzyja rozwijaniu kompetencji poznawczych,

takich jak krytyczne myślenie, umiejętność analizy danych, wyciągania wniosków czy rozwiązywanie problemów. Uczniowie uczestniczący w interaktywnych lekcjach częściej angażują się w zadania, są bardziej zmotywowani do podejmowania wyzwań i wykazują większą samodzielność w poszukiwaniu rozwiązań.

W przypadku zastosowań wirtualnej rzeczywistości, interaktywność przybiera różnorodne formy — od prostych gestów i ruchów wykonywanych za pomocą kontrolerów, przez manipulowanie obiektyami w przestrzeni wirtualnej, aż po rozwiązywanie zagadek i wykonywanie eksperymentów. Szczególnie efektywne okazują się aplikacje edukacyjne, które łączą elementy gry z nauką, umożliwiając użytkownikom rywalizację, zdobywanie punktów, odblokowywanie kolejnych poziomów czy rozwiązywanie zagadek fabularnych.

Środowiska edukacyjne o wysokim stopniu interaktywności znacząco zwiększą motywację wewnętrzną oraz podnoszą poziom satysfakcji z nauki. Uczniowie mają również większą łatwość w przyswajaniu wiedzy oraz częściej uczestniczą w zajęciach, co pozytywnie wpływa na ogólne efekty dydaktyczne.

W kontekście projektowanej aplikacji typu escape room dla jaskini rzeczywistości wirtualnej, wysoki poziom interaktywności będzie kluczowym elementem wpływającym na atrakcyjność i skuteczność dydaktyczną opracowanego rozwiązania. Możliwość bezpośredniego wpływu na otoczenie, rozwiązywania zagadek matematycznych w przestrzeni wirtualnej oraz współpracy z innymi uczestnikami w czasie rzeczywistym stworzy warunki sprzyjające aktywnej, angażującej naucie i rozwijaniu kompetencji.

3. ANALIZA ISTNIEJĄCYCH ROZWIĄZAŃ

3.1. *Wirtualne escape roomy*

3.2. *Przegląd gier edukacyjnych z elementami VR*

3.2.1. *Blockerzz*

3.2.2. *Mage Math*

3.2.3. *EdScape XR*

3.3. *Wnioski z analizy dostępnych rozwiązań*

4. PROJEKT SYSTEMU

4.1. Wymagania funkcjonalne i niefunkcjonalne (Konrad Czarnecki)

Przed przystąpieniem do implementacji aplikacji edukacyjnej w formie escape roomu w środowisku rzeczywistości wirtualnej konieczne było precyzyjne określenie wymagań funkcjonalnych i niefunkcjonalnych, jakie powinna spełniać projektowana aplikacja. Zdefiniowanie wymagań pozwala na kontrolowanie procesu realizacji projektu, zapewnia jego zgodność z oczekiwaniami użytkowników końcowych oraz umożliwia przeprowadzenie testów walidacyjnych w końcowej fazie prac.

4.1.1. Wymagania funkcjonalne

Wymagania funkcjonalne określają, jakie działania użytkownik może wykonać w aplikacji oraz jakie funkcje powinna ona realizować. W przypadku projektowanej aplikacji escape room VR wymagania te obejmują:

Wyświetlanie wirtualnego środowiska escape roomu – Aplikacja powinna umożliwiać użytkownikowi poruszanie się po wirtualnym pokoju zagadek, obejmującym zestaw pomieszczeń związanych z różnymi działami matematyki.

Realizacja 13 zagadek matematycznych – Aplikacja musi zawierać 13 interaktywnych zagadek, po jednej dla każdego z wybranych działów matematyki: liczby rzeczywiste, wyrażenia algebraiczne, równania i nierówności, układy równań, funkcje, ciągi, trygonometria, planimetria, geometria analityczna, stereometria, kombinatoryka, rachunek prawdopodobieństwa i statystyka oraz optymalizacja i rachunek różniczkowy.

Interaktywność zagadek – Każda zagadka powinna wymagać od użytkownika aktywnej interakcji z otoczeniem wirtualnym, np. manipulowania obiekttami, wpisywania odpowiedzi czy przedstawiania elementów.

System weryfikacji poprawności odpowiedzi – Aplikacja powinna sprawdzać poprawność rozwiązań podawanych przez użytkownika oraz wyświetlać komunikaty informujące o sukcesie lub błędzie.

Podpowiedzi dla użytkownika – Każda zagadka powinna posiadać system podpowiedzi (tekstowych lub dźwiękowych), które użytkownik może wywołać w razie trudności z rozwiązaniem zadania.

Śledzenie postępu użytkownika – System powinien zapamiętywać, które zagadki zostały już rozwiązane, i odblokowywać dostęp do kolejnych pomieszczeń w ustalonej kolejności.

Zakończenie rozgrywki – Po rozwiązaniu wszystkich 13 zagadek aplikacja powinna wyświetlić ekran podsumowujący z informacją o ukończeniu escape roomu.

Obsługa urządzeń VR i interfejsów w Laboratorium – Aplikacja musi być kompatybilna z systemami śledzenia ruchu i projekcji wykorzystywanymi w Laboratorium Zanurzonej Wizualizacji Przestrzennej, w tym z systemami typu CAVE oraz kontrolerami ruchu.

4.1.2. Wymagania niefunkcjonalne

Wymagania niefunkcjonalne opisują cechy jakościowe systemu oraz warunki, jakie powinien spełniać, aby zapewnić poprawne działanie i pozytywne doświadczenia użytkowników. W kontekście projektowanej aplikacji escape room VR wymagania niefunkcjonalne obejmują:

Kompatybilność z infrastrukturą LZWP – Aplikacja musi działać poprawnie w warunkach technicznych Laboratorium Zanurzonej Wizualizacji Przestrzennej, współpracując z systemem projektijnym CAVE oraz wykorzystywanymi kontrolerami ruchu.

Wydajność – Aplikacja powinna działać płynnie, z minimalnym opóźnieniem renderowania obrazu i reakcji na ruchy użytkownika, zapewniając co najmniej 60 klatek na sekundę w środowisku VR.

Jakość grafiki i dźwięku – Wirtualne środowisko escape roomu powinno charakteryzować się realistyczną i spójną oprawą wizualną oraz odpowiednio dobranymi efektami dźwiękowymi wspomagającymi immersję.

Łatwość rozbudowy i modyfikacji – Struktura projektu powinna umożliwiać przyszłą rozbudowę aplikacji o nowe zagadki, pomieszczenia lub funkcjonalności bez konieczności przebudowy istniejącego kodu i modeli.

Przenośność kodu i zasobów – Wszystkie pliki źródłowe, modele oraz zasoby dźwiękowe powinny być przechowywane w repozytorium GitHub w sposób umożliwiający łatwe przenoszenie projektu pomiędzy stanowiskami roboczymi oraz serwerami laboratorium.

Dokumentacja – Projekt musi być opatrzony szczegółową dokumentacją techniczną i użytkową, opisującą strukturę aplikacji, sposób instalacji, obsługi oraz instrukcje dotyczące przyszłej rozbudowy.

4.2. Scenariusz gry (Konrad Czarnecki)

Po uruchomieniu aplikacji użytkownik znajduje się w wirtualnym holu startowym, który pełni funkcję ekranu powitalnego. W tym miejscu użytkownik może zapoznać się z zasadami działania aplikacji oraz sterowaniem w środowisku VR.

W momencie rozpoczęcia gry uruchamiany jest licznik czasu, który będzie rejestrował całkowity czas potrzebny na ukończenie escape roomu. Po wybraniu opcji rozpoczęcia gry użytkownik przenosi się do pierwszego z 13 wirtualnych pomieszczeń. Każde pomieszczenie jest utrzymane w odmiennym stylu wizualnym i zawiera jedną interaktywną zagadkę matematyczną z przypisanego działu.

W każdym pokoju gracz ma możliwość poruszania się po przestrzeni VR, wchodzenia w interakcje z elementami zagadki, wywoływania podpowiedzi (jeśli jest dostępna), przechodzenia do kolejnego pokoju po poprawnym rozwiązaniu zagadki.

Działanie każdej zagadki może wymagać różnych form interakcji — takich jak przeciąganie obiektów, wpisywanie wyników na wirtualnej klawiaturze, wskazywanie elementów przestrzeni, czy manipulowanie wirtualnymi narzędziami (np. suwaki, przełączniki, dźwignie). Po rozwiązaniu ostatniej, trzynastej zagadki użytkownik zostaje przeniesiony do wirtualnego pokoju podsumowań. W tym miejscu aplikacja prezentuje takie informacje jak łączny czas ukończenia gry i liczbę wykorzystanych podpowiedzi.

W przypadku przerwania rozgrywki przez użytkownika lub awarii systemu, aplikacja powinna zapewniać możliwość powrotu do holu głównego lub całkowitego zamknięcia gry bez ryzyka utraty

integralności danych systemu. Rozwiążanie to gwarantuje bezpieczeństwo użytkownika oraz stabilność aplikacji w środowisku CAVE.

Tak skonstruowany scenariusz rozgrywki umożliwia nie tylko wprowadzenie użytkownika w świat wirtualnej rzeczywistości, ale także zapewnia dynamiczny, uporządkowany i logiczny przebieg interakcji. Dzięki systematycznemu rozwiązywaniu zagadek i przemieszczaniu się między pokojami, aplikacja utrzymuje wysoki poziom zaangażowania i motywacji użytkowników do ukończenia całej gry.

4.3. Zagadki (Konrad Czarnecki)

W ramach realizacji projektu opracowano trzynaście interaktywnych zagadek matematycznych, z których każda reprezentuje inny dział matematyki nauczanej na poziomie szkoły średniej. Zagadki zostały zaprojektowane tak, aby nie tylko sprawdzać wiedzę użytkowników, ale również aktywizować ich poprzez wykorzystanie mechanik charakterystycznych dla gier logicznych i escape roomów. Dzięki zastosowaniu technologii wirtualnej rzeczywistości, gracze mogą wchodzić w bezpośrednią interakcję z otoczeniem i rozwiązywać zadania w formie angażujących, przestrzennych łamigłówek. Każde z zadań posiada unikalny scenariusz i zasady działania, dostosowane do specyfiki danego działu matematyki oraz do możliwości sprzętowych Laboratorium Zanurzonej Wizualizacji Przestrzennej. Poniżej przedstawiono szczegółowy opis wszystkich trzynastu zagadek, wraz z określeniem ich celu, przebiegu oraz warunków zakończenia.

4.3.1. Wyrażenia algebraiczne

Pierwsze z zadań wirtualnego escape roomu opiera się na klasycznej mechanice gry logicznej „piętnastki”, zaadaptowanej do środowiska matematycznego. Po aktywacji centralnego stołu w wirtualnym pomieszczeniu, na jego powierzchni pojawia się plansza o wymiarach 7×7 , zawierająca kafelki z symbolami matematycznymi i wyrażeniami algebraicznymi. Na kafelkach umieszczono między innymi litery zmiennych (a, b), symbole potęg ($^2, ^3$), operatory działań matematycznych ($+, -, =$), nawiasy oraz złożone wyrażenia typu $2ab$, ab^2 czy a^3 . Pod planszą wyświetlana jest lista nazw siedmiu wzorów skróconego mnożenia, takich jak: „Kwadrat sumy”, „Kwadrat różnicy”, „Różnica kwadratów” czy „Suma sześcianów”. W treści zadania nie podano jednak pełnych wzorów, co zmusza graczy do samodzielnego przypomnienia ich z wiedzy szkolnej. Celem użytkownika jest ułożenie siedmiu pełnych wyrażeń odpowiadających tym wzorom, poprzez przesuwanie kafelków na planszy i formowanie poprawnych układów w pionie lub poziomie, zgodnie z zasadami klasycznej gry „piętnastki”. System automatycznie rozpoznaje poprawnie ułożone wzory i oznacza je na liście oraz na planszy. Zadanie można rozwiązywać w dowolnej kolejności, a jego ukończenie następuje po poprawnym ułożeniu wszystkich siedmiu wyrażeń.

4.3.2. Planimetria

Po ukończeniu pierwszego etapu, powierzchnia interaktywnego stołu zmienia swój wygląd, a na ekranie pojawia się plansza zawierająca dwanaście par figur geometrycznych. Wśród nich znajdują się takie figury jak kwadraty, romby, prostokąty, równoległoboki, trapezy oraz różne typy trójkątów. Zadanie ma na celu sprawdzenie znajomości zależności między figurami oraz umiejętności rozpoznawania ich cech charakterystycznych. Na planszy, pomiędzy każdą parą figur, znajduje się miejsce przeznaczone na wybór relacji. Gracze, analizując właściwości obu figur w każdej parze, muszą zdecydować, czy jedna z nich jest szczególnym przypadkiem drugiej, czy

też nie. Do dyspozycji mają dwa symbole: strzałkę w prawo, oznaczającą, że pierwsza figura jest szczególnym przypadkiem drugiej, oraz przekreślona strzałka, oznaczającą brak takiej zależności. Po dokonaniu wszystkich wyborów system natychmiast weryfikuje poprawność ustawionych relacji. W przypadku błędnych odpowiedzi, gracze otrzymują stosowny komunikat i możliwość poprawienia wyłącznie tych relacji, które zostały oznaczone niepoprawnie. Zadanie kończy się w momencie, gdy wszystkie relacje zostaną poprawnie oznaczone.

4.3.3. *Równania i nierówności*

W trzecim zadaniu gracze trafiają na scenę stylizowaną na film przygodowy, gdzie postać odkrywcy musi przejść przez wiszący most zbudowany z kamiennych płyt. Na każdej z płyt widnieje liczba rzeczywista, a na ścianie pojawia się układ dwóch nierówności. Zadaniem uczestników jest rozwiązanie układu i wybranie płyt z wartościami należącymi do przedziału spełniającego oba warunki. Gracze prowadzą postać, klikając po kolej właściwe płytki, a błędny wybór powoduje zapadnięcie się płyt i konieczność powrotu na początek. Po poprawnym przejściu przez most gracze odblokowują dostęp do kolejnej zagadki.

4.3.4. *Funkcje*

W czwartym zadaniu gracze pracują z dużą interaktywną tablicą, na której wyświetlany jest układ współrzędnych. Ich celem jest dostosowanie parametrów funkcji liniowej w taki sposób, aby jej wykres przechodził przez trzy wyznaczone punkty. Dzięki suwakom lub polom edycyjnym uczestnicy mogą dynamicznie zmieniać współczynnik kierunkowy oraz wyraz wolny funkcji, obserwując na bieżąco, jak zmienia się jej wykres. Zadanie polega na precyjnym dobraniu wartości parametrów, by wykres przeciął wszystkie wskazane punkty.

4.3.5. *Geometria analityczna na płaszczyźnie kartezjańskiej*

W piątym zadaniu gracze ponownie pracują z interaktywną tablicą, na której tym razem pojawiają się dwa wykresy funkcji liniowych. Celem wyzwania jest odnalezienie współrzędnych punktu, w którym obie funkcje przecinają się na płaszczyźnie kartezjańskiej. Uczestnicy mogą odczytać ten punkt bezpośrednio z wykresu lub rozwiązać układ równań opisujących obie proste. Po ustaleniu wartości współrzędnych gracze wpisują je w specjalnie przygotowane pole. System weryfikuje poprawność odpowiedzi i odblokowuje przejście do kolejnego etapu gry.

4.3.6. *Kombinatoryka*

W szóstym zadaniu gracze natrafiają na metalowy sejf wyposażony w podświetlany panel numeryczny. Na wyświetlaczu obok pojawia się zagadka z zakresu kombinatoryki dotycząca liczby możliwych ustawień trzycyfrowego kodu PIN, w którym żadna z cyfr się nie powtarza. Uczestnicy muszą przypomnieć sobie zasady permutacji i wyliczyć, ile jest możliwych kombinacji, wybierając trzy różne cyfry spośród dziesięciu dostępnych. Po obliczeniu właściwej liczby gracze wpisują wynik na klawiaturze sejfu.

4.3.7. *Liczby rzeczywiste*

W siódmym zadaniu gracze pracują z dużą drewnianą skrzynią oznaczoną liczbami i symbolami zbiorów liczbowych. Pierwszym etapem jest przyporządkowanie wszystkich liczb nadrukowanych na ściankach skrzyni do odpowiednich zbiorów: liczb naturalnych, całkowitych, wymiernych

i niewymiernych. Na podstawie tego podziału gracze ustalają czterocyfrowy kod otwierający mechaniczny zamek. Po poprawnym wprowadzeniu kombinacji, wewnątrz skrzyni odnajdują płaską planszę ilustrującą przecięcia zbiorów w formie diagramu Venna. Obok leżą kafelki z symbolami działań na zbiorach, które należy odpowiednio rozmieścić na planszy, uzupełniając brakujące miejsca. Po wykonaniu tego zadania, gracze odkrywają kolejny mechanizm, gdzie muszą ułożyć zależności między zbiorami liczbowymi za pomocą ruchomych symboli i strzałek, odzwierciedlając relacje zawierania się zbiorów. Poprawne wykonanie wszystkich trzech etapów aktywuje mechanizm w skrzyni i otwiera dostęp do kolejnego etapu gry.

4.3.8. Trygonometria

W ósmym zadaniu gracze stają przed ścianą, na której umieszczone są cztery duże okręgi, symbolizujące funkcje trygonometryczne: sinus, cosinus, tangens i cotangens. Każdy okrąg podzielony jest na cztery ćwiartki układu współrzędnych oznaczone numerami I–IV. Zadaniem uczestników jest uzupełnienie pustych pól przy każdej ćwiartce odpowiednim znakiem „+” lub „–”, wskazującym, czy dana funkcja przyjmuje w tej ćwiartce wartości dodatnie czy ujemne. Poprawne przypisanie znaków wymaga znajomości własności funkcji trygonometrycznych w poszczególnych ćwiartkach.

4.3.9. Ciągi

W dziewiątym zadaniu gracze przenoszą się na interaktywną platformę stylizowaną na arenę gry rytmicznej, gdzie głównym wyzwaniem jest rozpoznawanie kolejnych wyrazów ciągów liczbowych. Na ekranie wyświetlana jest formuła ciągu arytmetycznego lub geometrycznego, a z góry spadają kostki z różnymi wartościami liczbowymi. Zadaniem uczestników jest przecięcie mieczem świetlnym tylko tych kostek, które zawierają poprawne elementy podanego ciągu. Jeśli kostka nie pasuje do ciągu, gracze powinni ją pozostawić nienaruszoną. Za każdą poprawną akcję przyznawany jest punkt, natomiast błędne cięcie skutkuje odebraniem punktu. Tempo gry rośnie wraz z postępem, zwiększając wymagania dotyczące refleksu i precyzji. Gra kończy się sukcesem po zdobyciu 50 punktów i składa się z dwóch etapów: rozpoznawania ciągu arytmetycznego, a następnie geometrycznego.

4.3.10. Rachunek prawdopodobieństwa i statystyka

W tym zadaniu gracze stają przed wyzwaniem probabilistycznym, reprezentowanym przez interaktywny stół z postacią Morfeusza oraz dwoma pojemnikami i zestawem 100 tabletek – po 50 czerwonych i niebieskich. Celem jest rozdzielenie tabletek między pojemniki w taki sposób, aby maksymalizować szansę wybrania czerwonej tabletki przez Morfeusza. Gracze mogą dowolnie przesuwać tabletki, testując różne układy i obserwując wyświetlany na bieżąco procent prawdopodobieństwa sukcesu. Optymalnym rozwiązaniem jest umieszczenie jednej czerwonej tabletki w pierwszym pojemniku, a pozostałych 49 czerwonych oraz 50 niebieskich w drugim, co daje maksymalną szansę około 74,75%. Zadanie wymaga od uczestników zrozumienia i wykorzystania zasad prawdopodobieństwa do podejmowania decyzji.

4.3.11. Optymalizacja i rachunek różniczkowy

Gracze stają przed wyzwaniem trafienia w cele na symulowanym polu bitwy. Sterują kątem nachylenia armaty, regulując go na pokrętle, aby zmodyfikować trajektorię lotu pocisku wyście-

tlaną na ekranie. Ich zadaniem jest precyzyjne ustawienie kąta, tak aby pocisk trafił w pojawiający się w losowej pozycji cel. Po każdym strzale pojawia się nowy cel, a gracz musi trafić łącznie 15 razy. W trakcie rozgrywki uczestnicy poznają związek między parametrami funkcji kwadratowej opisującej tor lotu, a jej maksimum, ucząc się podstaw optymalizacji i rachunku różniczkowego.

4.3.12. *Układy równań*

Gracze muszą umieścić odpowiednią liczbę jednostek energii w trzech kolorowych pojemnikach: czerwonym, zielonym i niebieskim, tak aby spełnić podane warunki dotyczące ich wydajności i sumy energii. Poprawne ustawienie 12 jednostek energii, z uwzględnieniem zależności między pojemnikami, spowoduje zapalenie się dużej żarówki nad stołem, sygnalizując sukces. System na bieżąco weryfikuje poprawność i informuje o błędach, dając możliwość kolejnych prób.

4.3.13. *Stereometria*

Gracze w półmroku obracają na obrotowym podeście pojawiającą się bryłę przestrzenną, taką jak graniastosłup, ostrosłup, walec czy stożek. Muszą rozpoznać jej nazwę oraz wskazać właściwości, takie jak liczba ścian, wierzchołków, krawędzi, rodzaj podstawy czy kąty między elementami.

4.4. *Model przypadków użycia*

4.5. *Projekt architektury systemu*

4.6. *Projekt interfejsu użytkownika i środowiska gry*

5. TECHNOLOGIE I NARZĘDZIA

Realizacja projektu aplikacji edukacyjnej typu escape room w środowisku rzeczywistości wirtualnej wymagała doboru odpowiednich narzędzi oraz technologii, które umożliwiły stworzenie interaktywnej, atrakcyjnej wizualnie i funkcjonalnej aplikacji kompatybilnej z systemami dostępnymi w Laboratorium Zanurzonej Wizualizacji Przestrzennej.

5.1. Silnik gry (Konrad Czarnecki)

Do stworzenia aplikacji wirtualnego escape roomu zdecydowano się na wykorzystanie silnika Unreal Engine 5, który jest jednym z najpopularniejszych środowisk do tworzenia gier komputerowych oraz aplikacji wirtualnej rzeczywistości. Unreal Engine, rozwijany przez firmę Epic Games, umożliwia tworzenie zaawansowanych wizualnie, interaktywnych projektów 3D oraz VR dzięki nowoczesnemu systemowi renderowania, rozbudowanemu edytoriowi oraz wsparciu dla technologii immersyjnych.

Silnik ten oferuje użytkownikom możliwość programowania logiki aplikacji zarówno w języku C++, jak i wykorzystaniem wizualnego systemu skryptowego Blueprint, co znacząco przyspiesza proces tworzenia prototypów i ułatwia implementację interakcji w środowisku VR. Unreal Engine zapewnia również bogaty zestaw narzędzi do tworzenia animacji, efektów specjalnych, obsługi dźwięku oraz integracji z zewnętrznymi bibliotekami.

Podczas analizy możliwych rozwiązań rozważano również wykorzystanie silnika Unity, który podobnie jak Unreal Engine jest szeroko stosowany w branży gier i aplikacji VR. Unity charakteryzuje się dużą elastycznością, wsparciem dla wielu platform oraz dostępnością licznych wtyczek i rozszerzeń. W porównaniu z Unreal Engine, Unity posiada mniej rozbudowany natywny system graficzny oraz wymaga większego nakładu pracy przy tworzeniu zaawansowanych efektów wizualnych.

Ostatecznym argumentem przemawiającym za wyborem Unreal Engine była kwestia kompatybilności — Laboratorium Zanurzonej Wizualizacji Przestrzennej, w którym aplikacja miała zostać wdrożona, nie wspiera nowszych wersji Unity, natomiast Unreal Engine zapewniał pełną zgodność z istniejącą infrastrukturą sprzętową i programową laboratorium.

5.2. Laboratorium Zanurzonej Wizualizacji Przestrzennej (Konrad Czarnecki)

Laboratorium Zanurzonej Wizualizacji Przestrzennej (LZWP) to specjalistyczne środowisko badawczo-edukacyjne wyposażone w systemy rzeczywistości wirtualnej, umożliwiające tworzenie i testowanie aplikacji immersyjnych w warunkach kontrolowanych. W skład laboratorium wchodzą między innymi systemy typu CAVE, czyli pomieszczenia projekcyjne z ekranami sciennymi i podłogowymi, które otaczają użytkownika obrazem 3D wyświetlonym z kilku projektorów.

LZWP wyposażone jest również w systemy śledzenia pozycji i ruchu użytkownika oraz kontrolery umożliwiające interakcję z wirtualnym środowiskiem. Dzięki temu laboratorium stanowi doskonałe zaplecze do testowania aplikacji edukacyjnych VR oraz prowadzenia badań nad efektywnością i ergonomią rozwiązań immersyjnych.

5.3. Środowisko 3D (Konrad Czarnecki)

W procesie tworzenia aplikacji niezbędne było opracowanie modeli 3D reprezentujących obiekty, elementy wystroju oraz interaktywne przedmioty pojawiające się w wirtualnym escape roomie. Do tego celu wykorzystano Blender — darmowe oprogramowanie do modelowania trójwymiarowego, animacji, teksturowania oraz renderowania.

Blender oferuje szeroki zakres narzędzi umożliwiających tworzenie szczegółowych modeli 3D, generowanie animacji oraz przygotowywanie materiałów i tekstur kompatybilnych z silnikami do tworzenia gier. Dzięki wsparciu dla formatów eksportowych takich jak FBX i GLTF, modele stworzone w Blenderze mogły zostać bezproblemowo zimportowane do Unreal Engine i wykorzystane w aplikacji VR.

5.4. System kontroli wersji (Konrad Czarnecki)

Dla zapewnienia bezpieczeństwa danych oraz efektywnego zarządzania projektem zastosowano system kontroli wersji Git wraz z usługą hostingową GitHub. GitHub umożliwia przechowywanie kodu źródłowego, modeli 3D, plików dźwiękowych oraz dokumentacji projektowej w repozytorium zdalnym z możliwością współdzielenia zasobów pomiędzy członkami zespołu.

Dzięki systemowi kontroli wersji możliwe było śledzenie historii zmian, zarządzanie gałęziami projektowymi oraz szybkie przywracanie poprzednich wersji w przypadku wystąpienia błędów. W projekcie GitHub pełnił również rolę platformy do przechowywania backupów.

6. ANALIZA DYDAKTYCZNA I POTENCJAŁ EDUKACYJNY

6.1. *Motywacja uczniów do nauki matematyki*

6.2. *Możliwość personalizacji i różnicowania poziomu trudności*

6.3. *Wpływ imersji na naukę*

7. OPIS TEORETYCZNY ZAGADEK

W poniższym rozdziale znajduje się teoretyczny opis zagadek, który stanowi punkt wejściowy do implementacji projektu inżynierskiego. Stanowi on podstawę teoretyczną i nakreśla charakter pracy. Opisy poszczególnych zagadek zostały zredagowane na podstawie ogólnodostępnej podstawy programowej dla szkół średnich z przedmiotu matematyka. Jest to kluczowy aspekt całego projektu inżynierskiego, gdyż aplikacja końcowa ma być skierowana do uczniów szkół średnich, w związku z czym wymagany zakres wiedzy nie może wykraczać poza podstawę programową.

7.1. **Zadanie 1 – Wzory skróconego mnożenia (Autor)**

7.1.1. *Cel zadania*

7.1.2. *Zasady działania*

7.1.3. *Zakończenie zadania*

7.2. **Zadanie 2 – Planimetria (Jan Walczak)**

Po ukończeniu pierwszego zadania pojawia się tabela z trzema kolumnami. Dwie skrajne kolumny zawierają nazwy figur geometrycznych. Środkowa kolumna jest pusta i będzie uzupełniana przez gracza odpowiednimi symbolami.

7.2.1. *Cel zadania*

Celem gracza jest poprawne ułożenie relacji między figurami w kolejnych rzędach. Przykładowo, relacją taką jest to, że „każdy kwadrat jest rombem” lub „każdy kwadrat jest prostokątem”.

7.2.2. *Zasady działania*

Gracze przechodzą przez kolejne rzędy sekwencyjnie. Dopóki nie ułożą pierwszej relacji poprawnie, to nie mogą ułożyć kolejnej. Po poprawnym ułożeniu rzędu (wstawieniu odpowiedniego symbolu) podświetla się on na zielono, sygnalizując graczowi, że może przejść do kolejnego punktu. Symbole, które układają gracze są następujące:

- > to <
- > to nie <

Kwadrat	> to <	prostokąt
Prostokąt	> to nie <	Kwadrat
romb	[]	kwadrat
...

Tabela 7.1: Przykładowa tabela zawierająca zależności między figurami geometrycznymi.

Gracze kierują się swoją wiedzą matematyczną oraz następującymi własnościami figur:

- liczba boków,

- długości boków,
- kąty (proste lub nie),
- cechy charakterystyczne danej figury.

7.2.3. Zakończenie zadania

Po poprawnym uzupełnieniu wszystkich relacji zadanie uznaje się za zakończone. Gracz może przejść do kolejnego zadania.

7.3. Zadanie 3 – Nierówności (*Konrad Czarnecki*)

Po wejściu do kolejnego pomieszczenia gracz widzi przed sobą most zawieszony nad przepaścią. Po drugiej stronie znajduje się zamknięte przejście, do którego gracz musi się dostać. Na bocznych ścianach wypisany jest układ nierówności: $2x - 5 < 9$; $x + 1 \geq 4$. Na płytach mostu umieszczone są różne liczby z zakresu liczb całkowitych (np.: 1, 2, 3, 4, 5, 6, 8, 7, 10).

7.3.1. Cel zadania

Gracz musi rozwiązać układ nierówności i wyznaczyć przedział, który spełnia oba warunki. Następnie powinien przeprowadzić postać przez most, poruszając się wyłącznie po płytach z wartościami należącymi do tego przedziału. Rozwiązywanie układu

1. Rozwiązywanie pierwszej nierówności:

- $2x - 5 < 9$
- $2x < 14$
- $x < 7$

2. Rozwiązywanie drugiej nierówności:

- $x + 1 \geq 4$
- $x \geq 3$

3. Wspólny przedział:

- $x \in [3; 7)$

Gracz musi wybrać wyłącznie płytki z wartościami większymi lub równymi 3 i mniejszymi od 7, np. 3, 4, 5, 6, 8.

7.3.2. Zasady działania

- Gracz poruszając postacią, przechodzi przez kolejne płytki.
- Poprawna płytka – postać przechodzi dalej.
- Błędna płytka – płytka zapada się lub podświetla na czerwono, a postać wraca na początek mostu. Gracz może próbować dowolną liczbę razy, aż do skutecznego przejścia na drugą stronę.

7.3.3. Zakończenie zadania

Zadanie zostaje uznane za zakończone, gdy graczowi uda się przejść na drugą stronę mostu.

7.4. Zadanie 4 – Funkcje (Konrad Czarnecki)

Zadania 4 i 5 są realizowane w jednym pomieszczeniu. Gracz widzi dwie duże tablice, umieszczone na ścianach. Na obu naniesiona jest siatka układu współrzędnych. W układzie współrzędnych pojawiają się trzy punkty, np.:

- Punkt A (1, 2)
- Punkt B (3, 4)
- Punkt C (5, 6)

oraz wzór funkcji z losowo wybranymi współczynnikami. Poniżej widoczne są elementy służące do sterowania wykresem funkcji. Dają one możliwość zmiany współczynników wylosowanego wzoru.

7.4.1. Cel zadania

Celem gracza jest dobranie odpowiednich współczynników funkcji tak, aby przeszła ona przez wszystkie wyświetcone punkty.

7.4.2. Zasady działania

Podczas wybierania współczynników funkcji przez gracza, jej wykres na tablicy zmienia się na bieżąco zgodnie z ustawionym wzorem. Wykres funkcji powinien być losowany z wcześniej zdefiniowanej puli par typu funkcja – punkty, tak aby po każdorazowym uruchomieniu zadania gracz czuł, że ma przed sobą nowe wyzwanie.

7.4.3. Zakończenie zadania

Rozwiążanie jest sprawdzane na bieżąco i gdy gracz dobierze współczynniki funkcji poprawnie, tj. przetnie ona wszystkie wybrane punkty, zadanie zostaje uznane za rozwiązane. W takim przypadku tablica się blokuje i podświetla na zielono.

7.5. Zadanie 5 – Geometria analityczna (Konrad Czarnecki)

W układzie współrzędnych pojawiają się dwa wykresy funkcji. Są one losowane z pewnej określonej puli, podobnie jak w zadaniu 4.

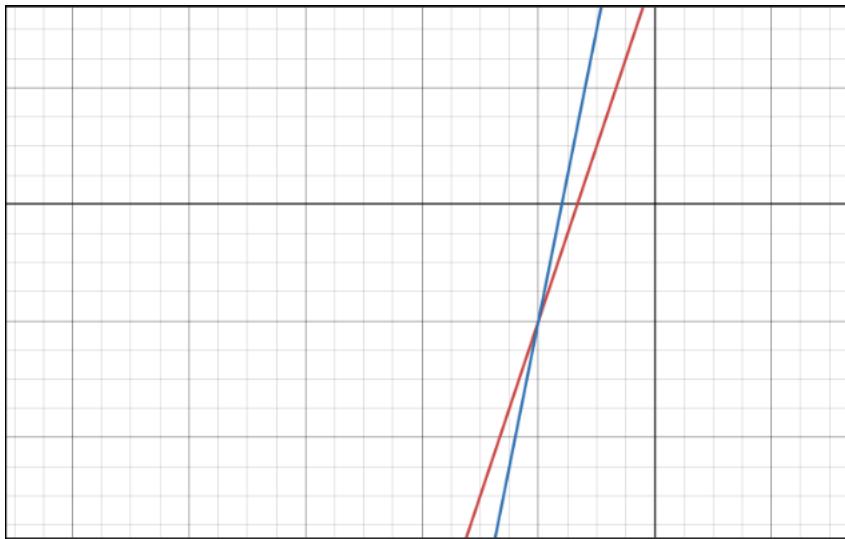
7.5.1. Cel zadania

Zadaniem gracza jest znalezienie punktu przecięcia dwóch funkcji prostych wyświetlanych w układzie współrzędnych. Określenie puli wyklucza możliwość prostych równoległych, które nie mają ze sobą żadnych punktów wspólnych lub w całości się pokrywają. W tych przypadkach zadanie byłoby niemożliwe do rozwiązania. Gracz będzie wpisywał swoją odpowiedź (współrzędne punktu przecięcia) na klawiaturze znajdującej się pod tablicą.

7.5.2. Zasady działania

Skala osi współrzędnych jest dobrana tak, by gracz nie mógł odczytać z niej rozwiązania; musi rozwiązać układ równań dysponując dwoma wzorami funkcji. Rysunek 7.1 obrazuje dwa wykresy:

- $y = 3x + 4$ (czerwony)
- $y = 5x + 8$ (niebieski)



Rysunek 7.1: Przykładowy wygląd tablicy z przecinającymi się wykresami

Gracz dysponuje tymi wzorami i na ich podstawie musi rozwiązać układ równań. Wykresy powinny być dobrane tak, aby obie współrzędne punktu przecięcia były liczbami całkowitymi.

7.5.3. Zakończenie zadania

Rozwiążanie jest sprawdzane na bieżąco i gdy gracz dobierze współrzędne punktu przecięcia funkcji poprawnie, zadanie zostaje uznane za rozwiązane. W takim przypadku tablica się blokuje i podświetla na zielono. Jeśli gracz wykonał wcześniej zadanie 4 (znajdujące się w tym samym pokoju), przechodzi do następnego zadania.

7.6. Zadanie 6 – Kombinatoryka (Autor)

a

7.6.1. Cel zadania

a

7.6.2. Zasady działania

a

7.6.3. Zakończenie zadania

7.7. Zadanie 7 – Liczby rzeczywiste i działania na zbiorach liczbowych

(Jan Walczak)

W kolejnym zadaniu gracze znajdują skrzynię, na której powierzchni – z każdej strony – narysowane są liczby: -3; 7; 12; 2; 5; 0; 10; 3,75; $\sqrt{2}$.

7.7.1. Cel zadania

Skrzynia wyposażona jest w mechaniczny zamek z pięcioma polami na cyfry, obok których znajdują się symbole zbiorów:

- N – liczby naturalne,
- Z – liczby całkowite,
- R – liczby rzeczywiste,
- W – liczby wymierne,
- NW – liczby niewymierne,

Gracz musi uważnie obejrzeć skrzynię i policzyć ile liczb, zapisanych na skrzyni, należy do jakiego zbioru. Po wpisaniu odpowiednich liczb, obok symboli zbiorów, skrzynia otwiera się, a gracz otrzymuje kilka symboli:

- zawieranie się zbiorów – \subset ,
- zbiór pusty – \emptyset ,
- iloczyn zbiorów – \cap .

Tym samym gracz przechodzi do drugiego etapu zadania. W drugim etapie w pokoju ukazuje się plansza z symbolami zbiorów, takimi jak na skrzyni. Celem gracza jest ułożenie poprawnej relacji między nimi tj. zbiór liczb naturalnych zawiera się w zbiorze liczb całkowitych, zbiór liczb całkowitych zawiera się w zbiorze liczb rzeczywistych itd.

7.7.2. Zasady działania

W pierwszym etapie zadania gracz powinien policzyć ile liczb ze skrzyni należy do danego zbioru i wpisać odpowiednią odpowiedź na kłódce. Przykładowo:

- N (naturalne): 7; 12; 2; 5; 10 – 5 liczb,
- C (całkowite): -3; 7; 12; 2; 5; 0; 10 – 7 liczb,
- R (rzeczywiste): wszystkie liczby – 9 liczb,
- W (wymierne): -3; 7; 12; 2; 5; 0; 10; 3,75 – 8 liczb,
- NW (niewymierne): $\sqrt{2}$ – 1 liczba.

Kombinacja do ustawienia na kłódce: 5, 7, 9, 8, 1.

W drugim etapie gracz powinien ustawić posiadane symbole między literami symbolizującymi kolejne zbiory i odpowiednio je obrócić, tak aby powstała między nimi poprawna relacja tj.

- $N \subset Z \subset R$,
- $NW \cap W = \emptyset$,
- $NW \cap R$.

7.7.3. Zakończenie zadania

Zadanie zostaje uznane za zakończone, gdy gracz poprawnie przejdzie przez oba etapy. Gracz nie może przejść do etapu drugiego, bez zakończenia etapu pierwszego – jest to wymuszone przez wymaganie otwarcia przez niego skrzyni.

7.8. Zadanie 8 – Znaki funkcji trygonometrycznych (Konrad Czarnecki)

Po ukończeniu zadania ze zbiorami gracz kieruje się do ściany z czterema dużymi okręgami jednostkowymi, oznaczonymi nazwami funkcji:

- $\sin(x)$
- $\cos(x)$
- $\tg(x)$
- $\ctg(x)$

Każdy z okręgów podzielony jest na cztery ćwiartki, oznaczone jako I, II, III, IV. Przy każdej ćwiartce znajduje się puste miejsce, które gracz musi wypełnić odpowiednim znakiem:

- „+” (plus) — funkcja przyjmuje wartości dodatnie
- „–” (minus) — funkcja przyjmuje wartości ujemne

Gracz zmienia znak naciskając na niego. Domyślnie wszystkie znaki są ustawione na puste, dopiero po pierwszym naciśnięciu zmieniają się na znak „+”, po kolejnym na „–”, następnie znów na „+”, itd.

7.8.1. Cel zadania

Celem gracza jest poprawne ustawienie znaków funkcji trygonometrycznych w każdej ćwiartce układu współrzędnych.

7.8.2. Zasady działania

Gracz przeciąga kostki „+” i „–” i umieszcza je w odpowiednich miejscach na planszy. Może dowolnie poprawiać swój wybór, dopóki nie zatwierdzi odpowiedzi.

7.8.3. Zakończenie zadania

Po poprawnym ułożeniu wszystkich znaków cianana rozświetla się na zielono.

7.9. Zadanie 9 – Ciągi liczbowe (Jan Walczak)

Poniżej opisane zadanie będzie podobne do gry wyprodukowanej przez Nintendo na platformę Pegasus.

7.9.1. Cel zadania

Zadaniem gracza będzie odpowiednio szybko obliczyć kolejne wyrazy ciągu arytmetycznego z podanego wzoru. Będzie on wyposażony w laserowy pistolet, którym będzie musiał strzelać w odpowiednio oznaczone kaczki.

7.9.2. Zasady działania

Na ekranie pojawia się formuła ciągu arytmetycznego lub geometrycznego, np.: $a_n = 2n + 1$ (ciąg arytmetyczny) wybranego z predefiniowanej puli. Gracz będzie musiał wybrać i strzelić do kaczki oznaczonej odpowiednią wartością kolejnych wyrazów ciągu. Gra przyspiesza (kaczki lecą coraz szybciej) razem z postępem w zadaniu. Aby ułatwić graczowi zadanie, na górze ekranu

będzie podany nie tylko wzór ciągu, ale też aktualny numer wyrazu tego ciągu. Przykładowo, dla wzoru $a_n = 2n + 1$:

- $n = 1$ – gracz musi trafić w kaczkę z liczbą 3,
- $n = 2$ – gracz musi trafić w kaczkę z liczbą 5

7.9.3. Zakończenie zadania

Gracz będzie zdobywał punkty za każdy poprawnie wybrany wyraz ciągu. Gra zakończy się po upływie określonego czasu lub jeśli gracz pomyli się trzy razy



Rysunek 7.2: gra Duck Hunt

7.10. Zadanie 10 – Prawdopodobieństwo (Autor)

a

7.10.1. Cel zadania

a

7.10.2. Zasady działania

a

7.10.3. Zakończenie zadania

7.11. Zadanie 11 – Optymalizacja i rachunek różniczkowy (Autor)

a

7.11.1. Cel zadania

a

7.11.2. Zasady działania

a

7.11.3. Zakończenie zadania

7.12. Zadanie 12 – Układy równań (Autor)

a

7.12.1. Cel zadania

a

7.12.2. Zasady działania

a

7.12.3. Zakończenie zadania

7.13. Zadanie 13 – Stereometria (Jan Walczak)

Bryły przestrzenne mogą sprawić uczniom trudność, ponieważ wymagają przejścia od rysunku dwuwymiarowego tj. takiego na kartce papieru, do wyobrażenia sobie ich w przestrzeni trójwymiarowej.

7.13.1. Cel zadania

Celem zadania jest zapoznać uczniów z podstawowymi wartościami brył geometrycznych poprzez ich samodzielne odkrywanie.

7.13.2. Zasady działania

Gracz wchodzi do ciemnego pokoju i jest wyposażony w źródło światła np. pochodnię lub latarkę. Na środku pokoju znajduje się jedna z brył, wylosowanych z danej puli: prostopadłościan, ostrosłup, graniastosłup, kula lub sześcian. Na ścianie pokoju znajduje się panel z pytaniemami np.:

- Ile ścian ma dana bryła?
- Ile krawędzi ma dana bryła?
- Co jest podstawą danej bryły?

Gracz obchodząc bryłę dookoła i rozświetlając ją latarką powinien móc udzielić odpowiedzi na te pytania.

7.13.3. Zakończenie zadania

Gdy gracz udziela poprawnych odpowiedzi na kolejne pytania zostają one podświetlone na zielono, sygnalizując graczyowi, że wykonał zadanie poprawnie. Po uzupełnieniu wszystkich odpowiedzi pokój rozświetla się, a gracz może przyjrzeć się bryle w pełnym świetle.

8. IMPLEMENTACJA ZAGADEK

8.1. *Zadanie 1 – wzory skróconego mnożenia (Autor)*

a

8.2. *Zadanie 2 – Planimetria (Jan Walczak)*

8.2.1. *Problemy i różnice w realizacji zadania w praktyce*

Podczas implementacji zadania, zgodnie z teorią opisaną w podrozdziale 7.2 napotkałem się z kilkoma problemami. Przede wszystkim zauważałem, że odpowiedzi są zero-jedynkowe. Przykładowo: uczeń ma do uzupełniania relację typu „każdy kwadrat ... prostokątem“. Jeśli odpowie niepoprawnie tj. zaznaczy odpowiedź „nie jest“, a zostanie od razu zapytany ponownie o tę samą relację, to od razu wyklucza jedną z odpowiedzi. Tym samym zadanie zatracza swoją wartość edukacyjną – uczeń może rozwiązać całe zadanie stosując jedynie metodę eliminacji.

Rozwiązań tego problemu, które zostało zaimplementowane, to zdefiniowanie puli takich relacji i po udzieleniu przez ucznia odpowiedzi, każdorazowe losowanie relacji innej niż ta poprzednia. W ten sposób uniemożliwia się uczniowi stosowania zasady eliminacji i wymusza na nim prawidłowe podejście.

Kolejnym problemem była prezentacja zadania. Zwykła tabela, którą uczeń miałby uzupełniać mogłaby wydać mu się mało ciekawa. Tym samym postanowiłem wizualnie usprawnić zagadkę. Na środku zostały umieszczone dwa przyciski:

- każdy
- nie każdy

Uczeń zostaje poinstruowany, że po obu ścianach pokoju zostaną wyświetlane różne figury geometryczne. Po lewej stronie, patrząc od przycisków – figury oznaczone kolorem czerwonym, po prawej stronie – figury oznaczone kolorem zielonym. Liczba figur jest stała, każdorazowo typ wyświetlanych figur jest wybierany z określonej puli i wyświetlany w pseudolosowej konfiguracji – losowane jest ich położenie, obrót oraz rozmiar. Zadaniem ucznia jest uzupełnianie kolejnych relacji poprzez wybieranie odpowiednich przycisków. Relacja wyświetlana na ścianie pokoju przedstawia się jako: „każdy typ figury, narysowany kolorem czerwonym, jest równocześnie typem figury oznaczonym kolorem zielonym“. Dodatkowo, tekst jest odpowiednio pokolorowany, tak aby uczeń nie miał wątpliwości, że chodzi o typ figury, wyświetlane tymże kolorem na ścianach.

8.2.2. *Implementacja struktury danych przechowującej relację*

Na początku pracy należało zdefiniować strukturę, przechowującą relację, czyli innymi słowy, pytanie na które uczeń będzie odpowiadał. Relacja taka została zdefiniowana jako aktor. Zawiera pola:

- Every – Wartość logiczna
- FigureA – ciąg tekstowy, reprezentujący pierwszą figurę w relacji
- FigureB – ciąg tekstowy, reprezentujący drugą figurę w relacji

Wartość zmiennej `Every` odpowiada na pytanie, czy każda figura typu pierwszego (`FigureA`) jest równocześnie figurą typu drugiego (`FigureB`). Struktura zawierała również funkcję słownikową, czyli taką, która tłumaczy wartości tekstowe na liczbowe.

8.2.3. Implementacja kontrolerów

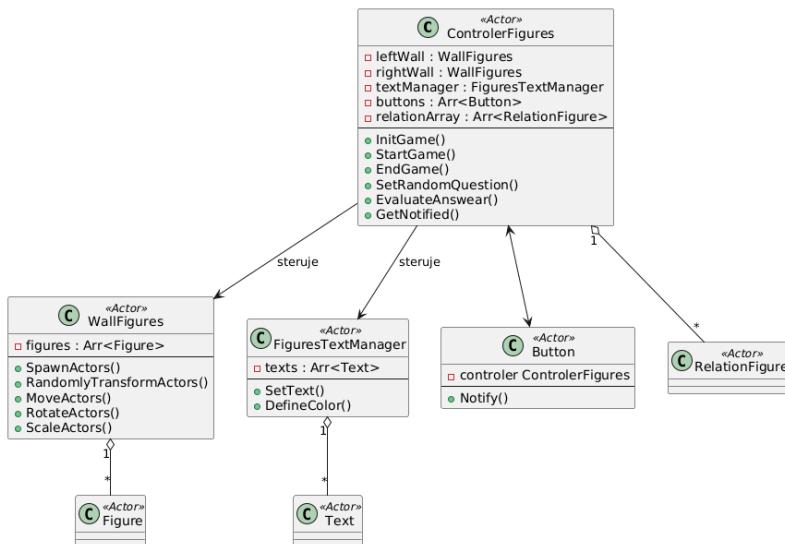
Aby zarządzać zagadką została zaimplementowana seria kontrolerów i menadżerów.

- `ControllerFigures` (główny kontroler)
- `FiguresTextManager`
- `WallFigures`

`WallFigures` to najprostszy z kontrolerów. Wykonuje polecenia głównego kontrolera. Jego zadaniem jest wyświetlanie żądanych figur i ich losowe ustawianie (obracanie, skalowanie, rozmieszczanie). W projekcie występują jego dwie instancje: kontroler lewy i prawy. Odpowiadają za odpowiednie ściany, na których wyświetlane są figury.

`FiguresTextManager` jest odpowiedzialny za zarządzanie tekstem wyświetlonym na ekranie. Wykonuje polecenia głównego kontrolera. Jego zadaniem jest odpowiednie wyświetlanie i kolorowanie tekstu.

`ControllerFigures` czyli kontroler główny jest najważniejszym elementem zagadki. Zawiera referencję do pozostałych kontrolerów, zarządza obiektami wyświetlonymi na scenie i kontroluje przebieg zadania. Dodatkowo zawiera dwustronną referencję z instancjami przycisków `Button`, które wysyłają do niego powiadomienia o tym, że zostały naciśnięte. Kontroler zawiera też predefiniowaną tablicę relacji typów figur opisanych wcześniej.



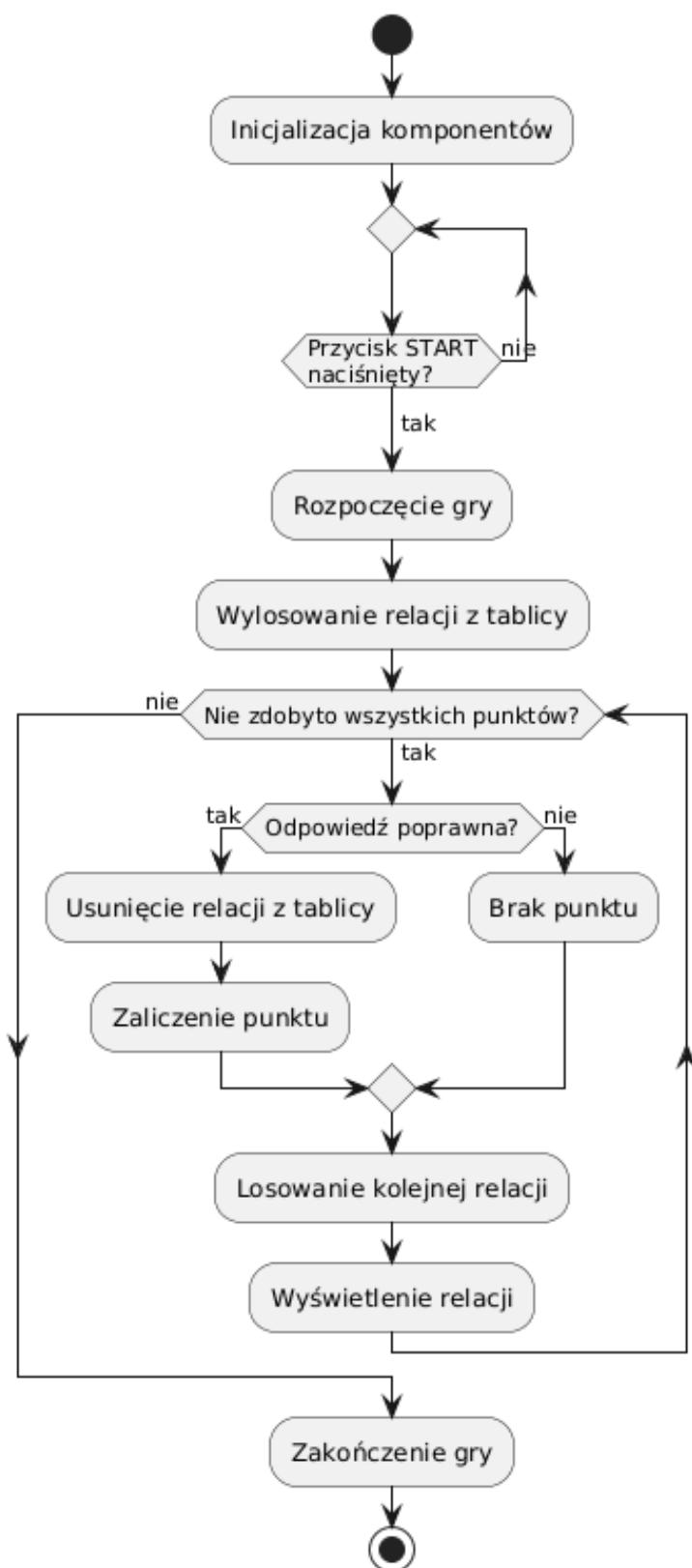
Rysunek 8.1: Diagram UML zawierający najważniejsze elementy kontrolerów dla zadania 2 – planimetria.

8.2.4. Przebieg zadania

Główny kontroler jest odpowiedzialny za inicjalizację zadania. Uruchamia on funkcję `InitBlueprint`, odpowiedzialną za wywołanie odpowiednich funkcji, z podlegnych elementów, takich jak: ustawienie tekstu, inicjalizacja ścian i przycisków. Ustawa także początkowe wartości zmiennych dla zadania.

Kontroler oczekuje na otrzymanie powiadomienia od przycisku odpowiedzialnego za uruchomienie gry. Kiedy otrzyma dane powiadomienie, za pośrednictwem odpowiedniej funkcji, uruchamia grę tj. wywołuje funkcje podrzędnych komponentów ustawiając im odpowiednią widoczność na scenie w grze.

Z predefiniowanej tablicy relacji zostaje wybrana jedna, która zostaje wyświetlona graczowi za pośrednictwem wywołania odpowiednich funkcji. Główny kontroler czeka na informacje od przycisków. Kiedy zostaje powiadomiony o tym, że jeden z nich, został naciśnięty sprawdza poprawność odpowiedzi. Jeśli odpowiedź jest poprawna, usuwa relację z tablicy i losuje kolejną. Jeśli nie jest poprawna, losuje kolejną relację do wyświetlenia graczowi i nie zalicza punktu. Gra kończy się po poprawnym uzupełnieniu wszystkich relacji.



Rysunek 8.2: Diagram przepływu dla zadania 2 – planimetria

8.3. Zadanie 3 – Nierówności (Autor)

8.4. Zadanie 4 – Funkcje (Autor)

8.5. Zadanie 5 – Geometria analityczna (Autor)

8.6. Zadanie 6 – Kombinatoryka (Autor)

8.7. Zadanie 7 – Liczby rzeczywiste i działania na zbiorach liczbowych

(Jan Walczak)

8.7.1. Problemy i różnice w realizacji zadania w praktyce

Podczas implementacji zadania, zgodnie z teorią opisaną w podrozdziale 7.7 musiałem wprowadzić, względem niej, kilka poprawek. Pierwotny mechanizm zakładał, że zamek, umieszczony na skrzyni, będzie wyposażony w pola na umieszczenie odpowiednich cyfr – liczby elementów danego podzbioru znajdujących się na niej. W praktyce okazało się to niemożliwe. Maksymalna liczba jednocyfrowa to 9, tym samym zadanie jest ograniczone do wyświetlania maksymalnie 9 liczb rzeczywistych w pokoju.

Postanowiłem zmienić zamysł i zamiast kłódki umieścić skrzynię z elektronicznym zamkiem i wyposażyć go w klawiaturę. W ten sposób dostajemy możliwość umieszczenia więcej niż 9 liczb w pokoju. Uczeń staje również przed dodatkowym zadaniem – wywnioskować kolejność wpisywanych liczb zgodnie z podpowiedzią, umieszczoną na ścianie (patrz rysunek 8.3).



Rysunek 8.3: Zrzut ekranu – skrzynia z zamkiem oraz podpowiedź na ścianie dla zadania 7.

Zmienione zostały również liczby, które są rozmieszczone w pokoju. Aktualnie pojawiają się liczby: $15; \pi; 4,5; \sqrt{2}; \frac{1}{4}; -3; 1; -7; \frac{1}{3} ; 9$ Tym samym kombinacja prezentuje się następująco (kolejność zgodna z podpowiedzią na rysunku 8.3)

- N (naturalne): 15; 1; 9 – 3 liczby,

- C (całkowite): 15; 1; 9; -3; -7 – 5 liczb,
- R (rzeczywiste): wszystkie liczby – 10 liczb,
- W (wymierne): 15; 1; 9; -3; -7; 4,5; $\frac{1}{4}$; $\frac{1}{3}$ – 8 liczb,
- NW (niewymierne): π ; $\sqrt{2}$ – 2 liczby.

Tym samym kombinacja do otwarcia sejfu to 351082.

Ostatnią rzeczą, która została zmieniona jest koncepcja drugiej części zadania. Niepotrzebny jest mechanizm obracania bloczków, ponieważ wybrane relacje są jednoznaczne – są czytane zawsze od lewej do prawej strony. Niepoprawny obrót bloczka od razu sugeruje błędą odpowiedź – tym samym jest to mechanizm bezcelowy.

Spośród wszystkich relacji zostały wybrane dwie:

- $N \subset Z \subset R$,
- $NW \cap W = \emptyset$,

Zostały one wybrane przez swoją charakterystykę. Pierwsza z nich pokazuje, jak kolejne zbiorы zawierają się w sobie nazwijem, a druga pokazuje rozłączność zbiorów liczb wymiernych i niewymiernych. Gracz ma do dyspozycji 6 bloczków:

- dwa bloczki z symbolem \subset ,
- dwa bloczki z symbolem \emptyset ,
- dwa bloczki z symbolem \cap

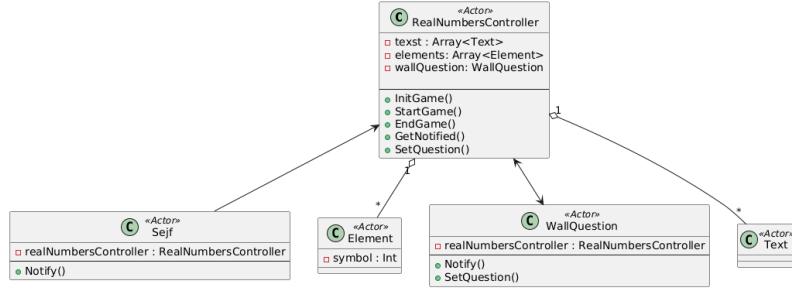
Mają mu posłużyć do uzupełnienia dwóch relacji wymienionych wcześniej. Po ułożeniu relacji poprawnie bądź nie bloczki pozostają do dyspozycji gracza.

8.7.2. Implementacja zadania

W przeciwnieństwie do zadania drugiego, opisanego w 8.2, został zaimplementowany jedynie jeden, główny kontroler `RealNumbersController`. Zarządza on przebiegiem zadania, położeniem elementów na scenie oraz wyświetlaniem tekstu. Zawiera odpowiednie referencje do komponentów i odpowiada za poprawną inicjalizację.

Komponent `Element` jest bloczkiem z odpowiednim symbolem relacji, `Sejf` jest skrzynią, która przechowuje bloczki, a `WallQuestion` to płaska tablica (ścianka), na której wyświetlane są pytania w drugiej części zadania.

`Sejf` posiada jednostronną referencję do głównego kontrolera. Kontroler może zostać przez niego powiadomiony, że gracz poprawnie wpisał kod, tak aby kontroler mógł sterować dalszym przebiegiem zadania. `WallQuestion` zawiera dwustronną referencję do głównego kontrolera. Mechanizm powiadomień działa tutaj podobnie ale kontroler może odpowiadać na otrzymywane powiadomienia poprzez wywoływanie odpowiednich funkcji.



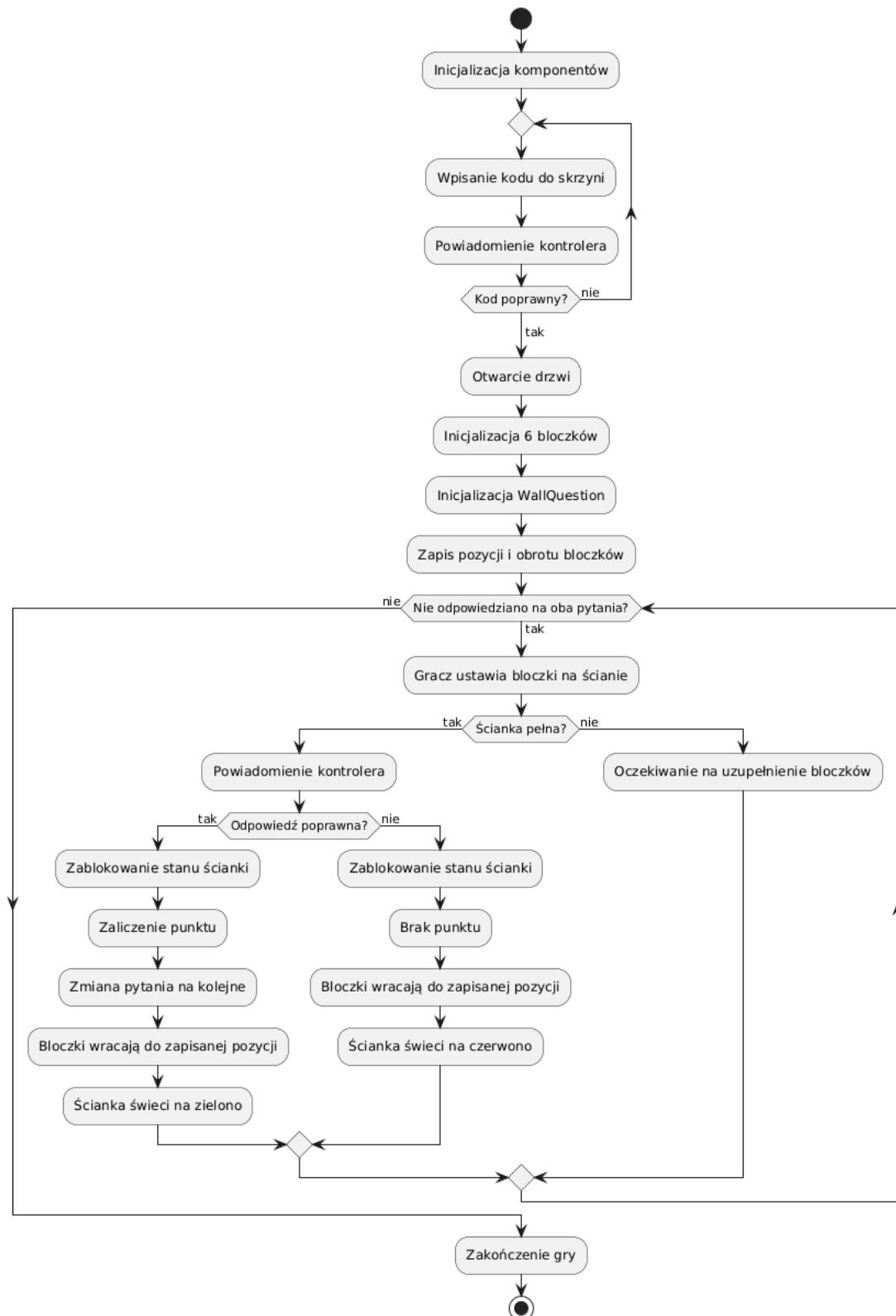
Rysunek 8.4: Diagram UML zawierający najważniejsze elementy dla zadania 7 – liczby rzeczywiste i działania na zbiorach liczbowych.

8.7.3. Przebieg zadania

Zadanie rozpoczyna się od pierwszej części – wpisania kodu do skrzyni. Kontroler inicjalizuje komponenty, a na ścianach i na skrzyni znajdują się wyświetlane liczby opisane w 8.7.1. Gracz powinien policzyć ile liczb i z jakiego zbioru jest wyświetlanych w pokoju i poprawnie wpisać kod, według podpowiedzi znajdującej się na ścianie. Po poprawnym wpisaniu kodu, kontroler główny zostaje powiadomiony przez instancję skrzyni, że zadanie może przejść do następnego etapu. Drzwi się otwierają a kontroler inicjalizuje 6 bloczków, opisanych w 8.7.1 i ściankę z pytaniami `WallQuestion`. Zapisuje pozycję oraz obrót bloczków w świecie, tak aby możliwy był powrót do stanu początkowego, między udzielanymi odpowiedziami.

Ścianka wykrywa kolizję z bloczkami, które gracz na niej umieszcza. W momencie kiedy zostaje uzupełniona w całości tj. bloczki wypełniają miejsca na odpowiedzi, to powiadamia kontroler o rozwiązaniu zadania. Kontroler decyduje czy odpowiedź gracza jest poprawna. W przypadku gdy jest, to przesyła do `WallQuestion` powiadomienie o zmianie pytania na kolejne i zalicza punkt. Jeżeli odpowiedź nie jest poprawna, to również powiadamia o tym ściankę, ale nie zalicza punktu. W obu przypadkach bloczki wracają na wcześniej zisaną pozycję. W zależności od poprawności udzielonej odpowiedzi ścianka rozświetla się kolorem zielonym lub czerwonym, tak żeby gracz miał pewność, że udzielił dobrej odpowiedzi. Podczas sprawdzania stan ścianki tj. ustalenie bloczków i pytanie zostaje zablokowany i gracz nie może go modyfikować (dokładać lub zabierać bloczków).

Kiedy gracz odpowie na oba pytania poprawnie poziom zostaje uznany za zaliczony.



Rysunek 8.5: Diagram przepływu dla zadania 7 – liczby rzeczywiste i działania na zbiorach liczbowych.

8.7.4. Implementacja mechanizmu uzupełniania ścianki

Ścianka, czyli `WallQuestion`, wykrywa kolizję z bloczkami, czyli `Element` w dwóch miejscach, w których będą one docelowo umieszczone. Kiedy gracz przytrzyma bloczek nad jednym z miejsc uruchamiana jest odpowiednia funkcja `StartCollision()`. W zależności, od tego gdzie został umieszczony bloczek, sprawdzany jest aktualny stan ścianki – czy w tym miejscu gracz zdążył już umieścić bloczek. Jeśli tak, to ścianka nie pozwoli mu umieścić w tym miejscu kolejnego bloczka, dopóki nie wyjmie poprzedniego. Dzieje się tak za sprawą dwóch zmiennych wartości logicznych: `ActorHovered` i `ActorLockedIn`. Oznaczają one kolejno: stan, w którym gracz trzyma bloczek nad ścianką i stan, w którym gracz umieścił bloczek na ściance. W przypadku gdy `ActorLockedIn` jest ustawione na wartość 0, to zapisywana jest referencja bloczku, który został właśnie umieszczony.

Dodatkowo informacja o tym, że gracz puścił bloczek, czyli zamierza go umieścić w danym miejscu, jest odpowiednio zapisywana w bloczku. Kiedy bloczek znajduje się w ręce gracza otrzymuje sygnaturę `held`. Kiedy go puści, sygnatura jest odbierana. Wyżej wymienione wartości logiczne są ustawiane dopiero wtedy, kiedy sygnatura najpierw istniała w instancji bloczka, a potem została z niej usunięta.

W przypadku, gdy gracz chce usunąć bloczek ze ścianki sytuacja jest analogiczna. Tym razem jednak, sprawdzamy czy bloczek, który zabiera gracz, jest tym znajdującym się na ściance, poprzez porównanie referencji.

8.8. Zadanie 8 – Znaki funkcji trygonometrycznych (Autor)

8.9. Zadanie 9 – Ciągi liczbowe (Jan Walczak)

8.9.1. Problemy i różnice w realizacji zadania w praktyce

W przeciwieństwie do zadań opisanych w podrozdziałach 8.2 i 8.7 w tym zadaniu nie pojawiło się dużo rozbieżności między teorią, a praktyczną implementacją.

Miejsce, w którym wprowadziłem modyfikację, to warunki zakończenia gry. Zadanie, przez swój projekt, stawia bardziej na szybką analizę, niż na długie zastanawianie się nad odpowiedzią. Jest bezpośrednio inspirowane prawdziwą grą, w której największą wartością, wyróżniającą gracza, jest jego czas reakcji. Z tego powodu zdecydowałem się na usunięcie limitu czasowego. Gra przyspiesza liniowo, razem z postępem tj. liczbą zestrzelonych przez gracza kaczek.

Dodatkowo zwiększyłem liczbę życia tj. możliwości na pomyłkę gracza do 4 zamiast 3. Jest to spowodowane wprowadzeniem dodatkowej mechaniki gry – jeśli kaczka wyleci poza zasięg gracza (za ścianę) gracz traci życie. Gra kończy się jedynie w momencie utraty wszyskich życia – co musi nastąpić, ponieważ prędkość kaczek będzie rosła tak długo, jak długo toczy się rozgrywka.

8.9.2. Implementacja pistoletu laserowego

Implementacja pistoletu laserowego była kluczowym punktem, potrzebnym do stworzenia całego poziomu.

Aby poprawnie zaimplementować taki laser potrzebne są w sumie trzy wartości:

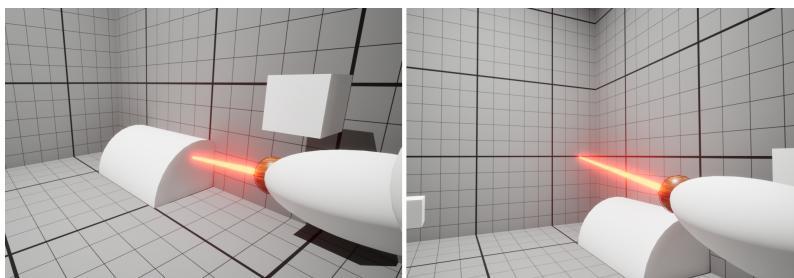
- punkt początkowy, od którego zaczynamy rysowanie lasera
- punkt końcowy, do którego laser będzie rysowany
- wektor kierunkowy, według którego laser będzie rysowany

Za punkt początkowy został obrany koniec ręki gracza tj. kontrolera którym steruje. Kolejno, z ustawienia gracza można wyznaczyć wektor kierunkowy. Wystarczy pobrać z kontrolera jego wartość przy użyciu systemowej funkcji Unreal Engine `Get Forward Vector`. Należało go zapisać, razem z punktem początkowym. Obliczenie punktu końcowego jest nieco bardziej skomplikowane. Polega na początkowym pomnożeniu wektora kierunkowego przez inny wektor, o dużych wartościach poszczególnych współrzędnych, a następnie dodaniu wyniku tego działania do współrzędnych punktu początkowego. W projekcie zastosowano wektor $(1000, 0; 1000, 0; 1000, 0)$. Tak obliczony punkt należało chwilowo zapisać jako punkt końcowy.

Następnie, przy użyciu systemowej funkcji `Line Trace by Channel`, należało wyznaczyć punkt, w którym linia, poprowadzona od punktu początkowego do końcowego, przecina się ze światłem przedstawionym w symulacji. Funkcja ta zwraca wynik w postaci wartości typu struktury `Hit Result`, który można rozbić na składowe i pozyskać z niego `Impact Point`, czyli rzeczywisty punkt końcowy oraz `Distance`, czyli wartość zmienoprzecinkową reprezentującą długość wyznaczonej linii.

Z obliczonymi wartościami można narysować laser, przebiegający od kontrolera gracza, do pierwszego napotkanego obiektu w grze. W tym celu kontroler ma na stałe przypisany obiekt `Static Mesh` z siatką statyczną w kształcie cylindra, który jest rozciągany według obliczonych wartości. Nałożony jest na niego materiał emitujący czerwone światło. Na końcu, rozciągniętej siatki statycznej, umieszczona jest niewidzialna kula, złączona z systemem kolizji. Dzięki niej można wykrywać, w jaki komponent na scenie, aktualnie celuje gracz.

Na rysunku 8.6 widać, że laser faktycznie wykrywa otoczenie, w które celuje gracz i zmienia odpowiednio swoją długość. Zmianie ulega również kierunek, w którym celuje gracz.



Rysunek 8.6: Porównanie rysowania lasera w przypadku kontaktu z przykładowym elementem otoczenia i ścianą.

8.9.3. Implementacja zadania

Podobnie jak w zadaniu opisanym w podrozdziale 8.7 zaimplementowany został jeden, główny kontroler `SequenceController`. Jego zadaniem jest inicjalizacja gry, zarządzanie przebiegiem gry, wyświetlanie tekstu i zarządzanieinstancjami kaczek. Kontroler wybiera pseudolosowo jeden z trzech dostępnych wzorów na kolejny wyraz ciągu arytmetycznego, według którego będzie wybierał kolejne elementy:

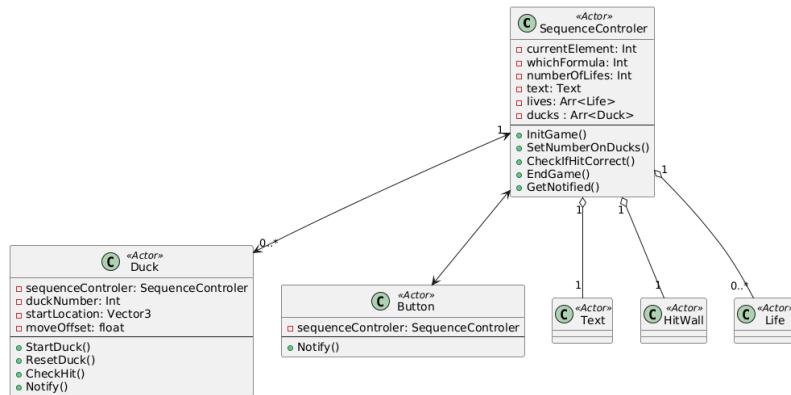
- $a_n = 3n + 4$
- $a_n = 2n + 3$
- $a_n = 5n + 7$

Komponent `Duck` implementuje funkcjonalności związane ze sterowaniem pojedynczej kaczki. Zawiera dwustronną referencję do obiektu głównego kontrolera i przesyła powiadomienia, kiedy

zostanie zestrzelony przez gracza, przy użyciu pistoletu laserowego. W jej instancji zapisywana jest pozycja początkowa, numer, który aktualnie jest na niej wyświetlany i wartość zmienoprzcinkowa, czyli droga, którą pokona kaczka w każdej klatce wykonywania się programu.

Gracz zostaje również zapoznany z zasadami działania gry. Kiedy jest gotowy, do rozpoczęcia gry, naciska przycisk podobny do tego opisanego w podrozdziale 8.2. Działa na tej samej zasadzie i również posiada dwustronną referencję do instancji kontrolera.

W zagadce występują jeszcze dwa komponenty: `HitWall` i `Life`. Główny kontroler zawiera tablicę referencji do 4 instancji obiektów `Life`. Reprezentują ilość pozostałych życ gracza i nie pełnią w logice działania programu żadnej istotnej roli, poza tą wizualną. `HitWall` jest obiektem niewidocznym, znajdującym się poza zasięgiem gracza. Leczące kaczki będą wykrywać ten obiekt, po przeleceniu przez ścianę i sygnalizować to głównemu kontrolerowi tak, jakby zostały trafione. Jeżeli kaczka uderzy w ten komponent to wyśle specjalny typ powiadomienia, mówiący o tym, że graczowi nie udało się jej zestrzelić. Jeśli kaczka zawierała poprawny wyraz ciągu to kontroler oznaczyto jako błąd i odejmie graczowi jedno życie. W takim przypadku, jeżeli gracz ma jeszcze pozostałe życie, wyświetli się kolejny wyraz ciągu, a kaczki zaczyną lecieć od początku.



Rysunek 8.7: Diagram UML zawierający najważniejsze elementy dla zadania 9 – Ciągi liczbowe.

8.9.4. Przebieg zadania

Zadanie rozpoczyna się od inicjalizacji komponentów przez główny kontroler. Początkowo graczowi zostaje wyświetlona instrukcja do zadania. Kiedy gracz będzie gotowy może rozpocząć grę naciskając przycisk „Start“. Kontroler zostaje w takiej sytuacji powiadomiony przez instancję przycisku i rozpoczyna grę. W sposób pseudolsowy zostaje wybrany jeden z trzech wzorów na kolejny n -ty wyraz ciągu arytmetycznego. Ustawia również poprawny wyraz ciągu, początkowo $n = 0$, na jednej z trzech kaczek. Pozostałe kaczki otrzymują wyrazy, większe lub mniejsze od poprawnego wyrazu, również w sposób pseudolosowy, biorąc pod uwagę to, że wybrane wyrazy mogą być odległe jedynie o 2 jednostki na osi współrzędnych, od poprawnego wyrazu.

Kaczki pokonują stałą jednostkę odległości w każdej jednostce czasu wykonywania się programu, wyrażoną za pomocą liczby zmienoprzcinkowej. Jednostka ta jest tym większa, im większy jest aktualny numer wyrazu ciągu arytmetycznego. Oznacza to, że będą przyspieszać razem z przebiegiem gry. Do tak obliczonej, stałej wartości, dodawana jest pseudolosowa wartość zmienoprzcinkowa wybrana z przedziału $[0,5; 0,7]$.

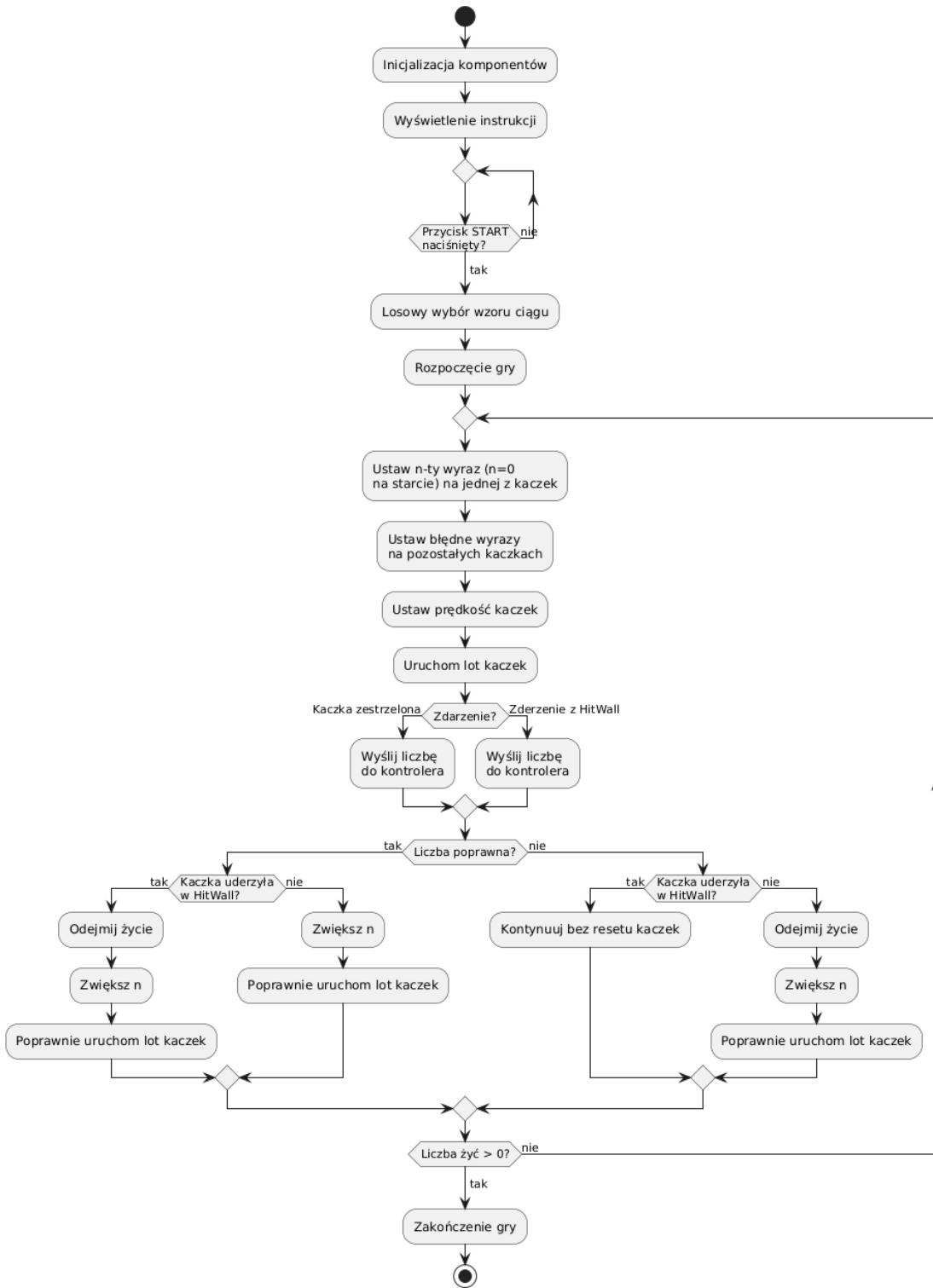
Kaczki poruszają się od lewej do prawej strony ekranu, aż do zderzenia się z instancją obiektu `HitWall`. Kaczki mogą zostać zestrzelone przez gracza, zanim zderzą się z wyżej wymienionym obiektem. W obu

W powiadomieniu przekazana jest również wartość stałoprzecinkowa, reprezentująca liczbę, która znajdowała się na danej kaczce.

Kontroler po otrzymaniu w powiadomieniu liczby oraz informacji, czy dana kaczka została zestrzelona, podejmuje decyzję:

1. Jeśli liczba jest poprawnym wyrazem ciągu – zalicza punkt i ustawia ponownie kaczki.
2. Jeśli liczba nie jest poprawna – odejmuje jedno życie, ale nie ustawia ponownie kaczek.
3. Jeśli kaczka, przechowująca poprawny wyraz ciągu, uderzy w obiekt `HitWall` (wyloci poza zasięg gracza) – odejmuje jedno życie i ponownie ustawia kaczki.

Ponowne ustawienie kaczek polega na obliczeniu kolejnego wyrazu ciągu arytmetycznego, ustawieniu go na jednej z nich, ustawieniu dwóm pozostałym kaczkom błędного wyrazu oraz na cofnięciu ich do stanu początkowego. Gra kończy się w momencie, gdy gracz straci wszystkie 4 życia.



Rysunek 8.8: Diagram przepływu dla zadania 9 – Ciągi liczbowe

8.10. Zadanie 10 – Prawdopodobieństwo (Autor)

8.11. Zadanie 11 – Optymalizacja i rachunek różniczkowy (Autor)

8.12. Zadanie 12 – Układy równań (Autor)

8.13. Zadanie 13 – Stereometria (Jan Walczak)

8.13.1. Problemy i różnice w realizacji zadania w praktyce

Zadanie nie odbiega za bardzo od wcześniej opracowanego opisu teoretycznego. Pierwszą różnicą jest ilość wyświetlanych brył geometrycznych. Według opisu graczowi miała być przedstawiana tylko jedna bryła. W praktyce takie zadanie byłoby bardzo krótkie. Zdefiniowano więc pulę 5 brył:

1. kulę,
2. stożek,
3. walec,
4. sześciian,
5. ostrosłup prawidłowy czworokątny.

Do każdej z wymienionych brył wybrano 3 pytania. Dotyczą one kolejno:

- rodzaju bryły,
- wzoru na obliczenie powierzchni bryły,
- wzoru na obliczenie objętości bryły.

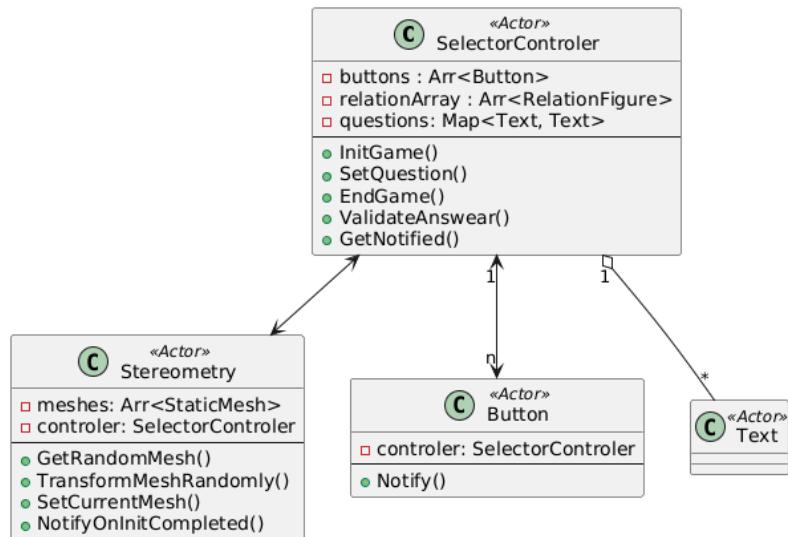
Gracz musi odpowiedzieć na wszystkie 3 pytania prawidłowo. Jeśli się pomyli, ze zbioru wybierana jest inna bryła, inna niż ta, o którą pytano go przed chwilą. W ten sposób uniknięto sytuacji, w której gracz odpowiadałby losowo i przechodził zadanie metodą eliminacji.

8.13.2. Implementacja zadania

W zadaniu został zaimplementowany jeden główny kontroler `SelectorController`. Jego zadaniem jest inicjalizacja gry, zarządzanie przebiegiem gry, wyświetlanie tekstu i wybór odpowiednich pytań, wyświetlanych graczowi. Zawiera on dwustronną referencję do obiektu `Stereometry` oraz dwustronną referencję do trzech obiektów `Button` (takich, jak te opisane w podrozdziałach 8.2 i 8.13). Zawiera także predefiniowany zbiór pytań, dotyczących każdej z brył, reprezentowany przez mapę typu `pytanie:poprawna odpowiedź`.

Obiekt `Stereometry` odpowiada za poprawne wyświetlanie brył. Zawiera 5 obiektów typu `Static Mesh`, które wyświetla w świecie gry, wybierając bryłę odpowiadającą aktualnie zadawanemu pytaniu. Jest też odpowiedzialny za pseudolosowy mechanizm wyboru brył. Przed wyświetleniem obiekt `Stereometry`, w sposób pseudolosowy, obraca bryłę, zmienia jej rozmiar, oraz położenie w świecie gry. W tym przypadku nie było potrzeby implementowania mechanizmu powiadomień, ponieważ rodzaj komunikatów wysyłanych pomiędzy obiektami `Stereometry` i `SelectorController` jest prosty i zawsze jednakowy (żądanie wyświetlenia nowej bryły). Dodatkowo, obiekt `Stereometry` zawiera referencję do kontrolera tylko dlatego, że jego inicjalizacja jest dłuża (prawdopodobnie przez wczytywanie siatek statycznych brył, z nałożoną teksturą odbijającą światło). Musi więc powiadomić

kontroler główny, że zakończył swoją inicjalizację, aby ten mógł zacząć wykonywać na nim operacje, przez wywołanie odpowiednich funkcji.

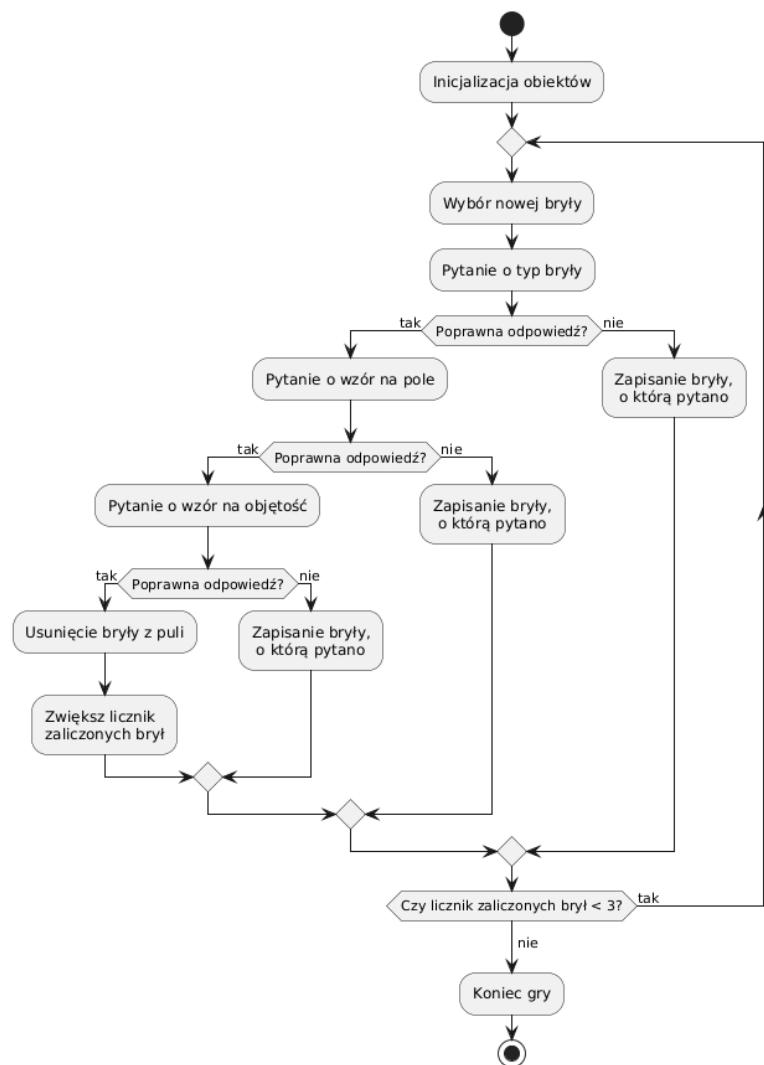


Rysunek 8.9: Diagram UML zawierający najważniejsze elementy kontrolerów dla zadania 13 – stereometria.

8.13.3. Przebieg zadania

Kontroler inicjalizuje podrzędne mu obiekty i rozpoczyna grę. W sposób pseudolosowy, wybierana jest jedna z 5 predefiniowanych brył, poprzez wywołanie odpowiedniej funkcji obiektu Stereometry. Gracz ma do wyboru 3 przyciski, które zawierają odpowiedzi A, B lub C. Pytania są wyświetlane sekwencyjnie – najpierw zadawane jest pytanie o typ bryły, potem o wzór na obliczenie jej pola powierzchni, a na końcu o wzór na obliczanie jej objętości.

Jeśli gracz pomyli się, podczas udzielania odpowiedzi, kontroler ponownie wywoła funkcję odpowiedzialną za wybór kolejnej bryły. W takim przypadku może się ona pojawić ponownie w zadaniu, ale nie od razu po udzieleniu błędnej odpowiedzi. Jeśli gracz odpowie na wszystkie 3 pytania prawidłowo, bryła zostaje uznana za zaliczoną i jest usuwana z puli dostępnych do wylosowania brył. Gra kończy się kiedy gracz poprawnie odpowie na pytania dotyczące 3 różnych brył.



Rysunek 8.10: Diagram przepływu dla zadania 13 – stereometria

9. PODSUMOWANIE

9.1. *Ocena realizacji celów*

9.2. *Propozycje rozwoju projektu*

9.3. *Wnioski końcowe*

SPIS RYSUNKÓW

7.1 Przykładowy wygląd tablicy z przecinającymi się wykresami	28
7.2 gra Duck Hunt	31
8.1 Diagram UML zawierający najważniejsze elementy kontrolerów dla zadania 2 – planimetria.	34
8.2 Diagram przepływu dla zadania 2 – planimetria	36
8.3 Zrzut ekranu – skrzynia z zamkiem oraz podpowiedź na ścianie dla zadania 7.	37
8.4 Diagram UML zawierający najważniejsze elementy dla zadania 7 – liczby rzeczywiste i działania na zbiorach liczbowych.	39
8.5 Diagram przepływu dla zadania 7 – liczby rzeczywiste i działania na zbiorach liczbowych.	40
8.6 Porównanie rysowania lasera w przypadku kontaktu z przykładowym elementem otoczenia i ścianą.	42
8.7 Diagram UML zawierający najważniejsze elementy dla zadania 9 – Ciągi liczbowe.	43
8.8 Diagram przepływu dla zadania 9 – Ciągi liczbowe	45
8.9 Diagram UML zawierający najważniejsze elementy kontrolerów dla zadania 13 – stereometria.	47
8.10 Diagram przepływu dla zadania 13 – stereometria	48

SPIS TABEL

7.1 Przykładowa tabela zawierająca zależności między figurami geometrycznymi. . . . 25