

Ogólne wymagania funkcjonalne i techniczne dla gier edukacyjnych



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Spis treści

| | |
|--------------------------------------|----|
| Ogólne wymagania funkcjonalne..... | 3 |
| Ogólne wymagania techniczne | 6 |
| Odstępstwa od ogólnych wymagań | 15 |
| Dodatkowe informacje | 16 |
| Dopuszczenie gry do użytku | 17 |
| Wyjaśnienie pojęć | 18 |
| Główna oś gameplay | 18 |
| Filary gry | 18 |
| Mechaniki gry | 18 |

Ogólne wymagania funkcjonalne

1. Dostosowanie do grup docelowych i zrównoważenie rozgrywki:

- Mechanika, poziom trudności, narracja i grafika każdej gry muszą być dostosowane do docelowej grupy wiekowej, w szczególności dla edukacji wczesnoszkolnej (dla młodszych graczy bardziej intuicyjne i proste mechaniki).
- Gry muszą uwzględniać różnorodność demograficzną i społeczną graczy, w tym różne poziomy umiejętności i predyspozycji poznawczych.
- Każda gra musi być odpowiednio wyważona pod względem poziomu trudności, z mechanizmami, które stopniowo zwiększają poziom wyzwania.
- Gry edukacyjne powinny oferować wyzwania intelektualne i łamigłówki dostosowane do poziomu gracza, ale też zawierać opcjonalny system pomocy.
- W grach edukacyjnych należy wziąć pod uwagę dostosowanie nie tylko języka, ale również treści kulturowych.

2. Edukacja wpleciona w rozgrywkę

- Każda gra musi zawierać elementy edukacyjne. Wiedza przekazywana w grach powinna być wpleciona w rozgrywkę w sposób naturalny (np. poprzez interakcje, dialogi, wybory gracza, eksplorację).
- Kluczowe treści edukacyjne (np. geografia, ekologia, historia) powinny być prezentowane w sposób angażujący, przyciągający uwagę i odpowiedni dla grupy docelowej.

3. Platforma docelowa

- Gra powinna być zoptymalizowana na docelową platformę określoną w scenariuszu (np. Windows, WEB), z możliwością rozbudowy na inne platformy w przyszłości.

4. Interfejs użytkownika (UI/UX)

- **Kompatybilność i skalowanie:** interfejs użytkownika musi dynamicznie dostosowywać się do różnych rozdzielczości (HD, 4K) i typów ekranów (komputery, urządzenia mobilne). W przypadku gier WEB musi również obsługiwać ekrany dotykowe.
- **Personalizacja i intuicyjność:** interfejs musi być intuicyjny, przejrzysty i łatwy w nawigacji. Powinna istnieć możliwość jego personalizacji (np. układ panelu, skróty klawiszowe), aby dostosować go do preferencji użytkownika.
- **Wsparcie dla graczy:** interfejs powinien zawierać system podpowiedzi, interaktywne tutoriale oraz tooltipy, które pomagają początkującym graczom. Podpowiedzi kontekstowe powinny być aktywowane, gdy gracz najeżdża na element lub dotyka go na ekranie dotykowym.
- **Zgodność z WCAG:** interfejs użytkownika musi być zgodny z wytycznymi WCAG opisanymi w scenariuszu, aby zapewnić dostępność dla osób z niepełnosprawnościami.

- **Udogodnienia dla młodszych graczy:** przycisk i opcje interakcji dla młodszych graczy powinny być większe i łatwiej dostępne.
- **Testowanie:** interfejs musi być przetestowany pod kątem użyteczności na różnych urządzeniach i przeglądarkach.

5. Osoby z niepełnosprawnościami

- Gry muszą uwzględniać dostępność dla osób z niepełnosprawnościami, np. tryby dla osób z trudnościami wzrokowymi, słuchowymi czy ruchowymi (większe przyciski, napisy, ułatwienia w sterowaniu).

6. System narracji i dialogów

- Narracja musi reagować na decyzje gracza, zapewniając adaptacyjność fabuły oraz różne ścieżki zakończeń.

7. Systemy nagród

- W grze powinien znajdować się system nagród, wirtualnych odznak lub medali, aby zachęcić graczy do kontynuowania rozgrywki i powrotu do gry po zakończeniu.
- Mechanika zdobywania doświadczenia powinna być przejrzysta i wspierać rozwój postaci (w grach opartych o rozwój postaci) oraz postępy w grze.
- System odznak i osiągnięć musi być kompatybilny z platformami takimi jak Steam, gdzie osiągnięcia zapisywane są w profilu użytkownika. W miarę możliwości powinien być również zastosowany **podobny mechanizm na platformie ZPE**, umożliwiający integrację z kontami użytkowników i zapisywanie osiągnięć na platformie edukacyjnej.

8. Interakcje społeczne i współpraca

- Gry wieloosobowe (lub z możliwością rozbudowy o multiplayer) powinny oferować narzędzia do komunikacji i współpracy między graczami.
- W grach z elementami kooperacyjnymi należy zadbać o systemy koordynacji działań drużynowych.

9. Instalator gry desktopowej

- **Wstawianie linków na ZPE:** w materiale edukacyjnym na ZPE należy umieścić dwa linki dla każdej gry:
 - [Pobierz z XXX] → otwiera stronę gry na platformie dystrybucyjnej (np. Steam)
 - [Pobierz instalator ZPE] → przekierowuje do strony pobierania gry z repozytorium ZPE
- **Obsługa platformy dystrybucyjnej (np. Steam)**
 - Kliknięcie w link otwiera stronę gry na platformie dystrybucyjnej (łącznie typu: steam://store/123456).
 - Gra na platformie dystrybucyjnej jest udostępniona nieodpłatnie.

- **Obsługa repozytorium ZPE**
 - Kliknięcie w link przekierowuje użytkownika na stronę pobierania (łącze typu: https://zpe.gov.pl/downloads/gra_XYZ_installer.exe).
 - Po pobraniu użytkownik ręcznie uruchamia instalator, który instaluje grę na komputerze.
- **Integracja z kontami ZPE (dla obu wersji)**
 1. Gry powinny wspierać logowanie użytkowników ZPE w celu personalizacji doświadczeń i ewentualnego zapisywania postępów.
 2. Dla wersji na platformie dystrybucyjnej (np. Steam): integracja może odbywać się poprzez powiązanie konta na tej platformie z kontem ZPE lub zalogowanie się do ZPE z poziomu gry.
 3. Dla wersji ZPE: gra może opcjonalnie wymagać logowania na konto ZPE przy pierwszym uruchomieniu, jeśli oferuje funkcje związane z personalizacją użytkownika lub synchronizacją postępów. Jeśli użytkownik zdecyduje się grać offline, powinien mieć możliwość kontynuowania rozgrywki, ale bez dostępu do funkcji wymagających połączenia z ZPE (np. synchronizacja postępów, leaderboardy).
 4. Zaleca się, aby gry były przygotowane do integracji z mechanizmem zarządzania kontami użytkowników na platformie ZPE, umożliwiającym m.in. integrację z profilem użytkownika, synchronizację postępów gry, zarządzanie wersjami i automatyczne aktualizacje, jeśli platforma ZPE będzie oferować taką funkcjonalność.

Ogólne wymagania techniczne

1. Modularność, skalowalność, wielojęzyczność

- Gra musi być zaprojektowana **modularnie**, aby umożliwiać przyszłą rozbudowę (np. dodawanie nowych poziomów, mechanik, scenariuszy, wydarzeń, postaci).
- System musi pozwalać na **dynamiczne dodawanie zawartości (DLC)** bez konieczności ingerencji w główny kod.
- **System aktualizacji** musi umożliwiać szybkie poprawki i dodawanie nowych funkcji bez modyfikacji całej gry.
- Należy zapewnić możliwość łatwego portowania gry na inne platformy (np. konsole, urządzenia mobilne), co wymaga **wstępnej optymalizacji kodu i architektury gry** z wykorzystaniem **modularnego silnika gry** i technologii umożliwiających łatwe dostosowanie interfejsu do różnych rozdzielczości i systemów sterowania (np. dotykowego w przypadku mobilnych wersji gry).
- Poziomy gry powinny być tworzone jako **niezależne moduły**, które można łatwo **dodawać, edytować i testować**.
- **System czasu w grze** (np. zmiany pór dnia, upływ lat) musi być na tyle elastyczny, aby można było łatwo wprowadzać nowe elementy narracyjne lub mechaniczne.
- **Beneficjent konkursowy musi zapewnić łatwe tłumaczenie gry** – interfejs, opisy misji, narracje i dialogi powinny być oddzielone od kodu źródłowego. System tłumaczeń powinien uwzględniać **różnice kulturowe i kontekstowe**. Na przykład, jednostki miar (metryczne/imperialne), waluty i specyficzne odniesienia historyczne powinny być dostosowane do wersji językowej gry.
- Wszystkie teksty powinny być przechowywane w **plikach JSON lub XML** w formacie UTF-8, aby umożliwić automatyczne tłumaczenie i aktualizację.
- **Modułowy system lokalizacji** powinien umożliwiać **dynamiczną zmianę języka** w trakcie gry.
- **Dostosowanie kulturowe**: w miarę możliwości treści gry powinny być dostosowane do różnych regionów (np. odniesienia historyczne, sposób narracji).

2. Zarządzanie zasobami i dokumentacja

- **Kod gry musi być dobrze udokumentowany**, aby ułatwić przyszłe aktualizacje i rozwój.
- **Wymagana jest pełna dokumentacja techniczna**, zawierająca schematy UML, opis kluczowych komponentów, mechanik gry oraz procedury wdrażania aktualizacji i rozszerzeń.
- Wszystkie zasoby gry (grafika, dźwięk, animacje, skrypty) muszą być przechowywane w sposób umożliwiający łatwe **zarządzanie aktualizacjami**.
- Wszelkie zasoby multimedialne (modele 3D, tekstury, dźwięki) **muszą być zawarte w paczce instalacyjnej gry** – gra nie może wymagać dostępu do zewnętrznych zasobów podczas działania.

- **Beneficjent konkursowy musi zapewnić pełny dostęp do plików źródłowych gry** (modele, tekstury, skrypty) na osobnym nośniku lub w chmurze.

3. Multiplatformowość

- Gry muszą działać na platformach sugerowanych w scenariuszach, ale beneficjent konkursowy musi zapewnić możliwość portowania na inne systemy i platformy, jeżeli będzie to wymagane w przyszłości.
- Zastosowanie silników takich jak Unity czy Unreal Engine umożliwi łatwe portowanie na różne platformy.

4. Kodowanie i struktura projektu

- **Język programowania:** kod gry musi być napisany w językach wspieranych przez wybrany silnik gry, np. C# dla Unity lub C++ dla Unreal Engine.
- **Czystość kodu:** kod musi być napisany w sposób przejrzysty, dobrze udokumentowany i strukturalnie podzielony na moduły. Dokumentacja techniczna powinna obejmować opis architektury kodu oraz komentarze w kluczowych miejscach.
- **Komentarze i dokumentacja:** kod musi być odpowiednio udokumentowany, z opisami kluczowych funkcji i klas. Każda część systemu, w szczególności mechaniki związane z interakcją i zagadkami, powinna być opisana w dokumentacji technicznej, którą beneficjenci konkursowi dostarczą z projektem.

5. Rozwój i przyszła rozbudowa

- Gra musi być zaprojektowana tak, aby umożliwiać łatwe wprowadzenie aktualizacji, rozszerzeń i dodatkowych treści. Kod musi być dobrze zorganizowany, a zasoby odpowiednio zarchiwizowane i oznaczone, aby inne zespoły mogły z łatwością kontynuować pracę nad projektem.

6. Zarządzanie zasobami gry

- Wszystkie zasoby muszą być zoptymalizowane pod kątem wydajności, aby gra działała płynnie nawet na średniej klasy sprzęcie.
- Zasoby graficzne, dźwiękowe i tekstowe muszą być przechowywane w sposób modułowy i umożliwiać łatwe aktualizacje.
- System zarządzania zasobami powinien umożliwiać automatyczną kompresję nowych materiałów w celu zmniejszenia przestrzeni dyskowej.

7. Platforma docelowa

- Wszystkie gry (z wyjątkiem gier WEB) są przeznaczone na komputery PC z systemem Windows 10 i nowszym oraz możliwością rozbudowy (portowania) na inne platformy w przyszłości.
- Wszystkie gry WEB są przeznaczone na przeglądarki na komputerach osobistych oraz na urządzeniach mobilnych (tablety, smartfony) i powinny zostać wykonane jako komponent interaktywny ZPE.

8. Kompatybilność z platformą docelową

- Wszystkie gry muszą być kompatybilne z docelową platformą, czy to komputerami stacjonarnymi (Windows), czy przeglądarkami internetowymi (web), zarówno na komputerach jak i urządzeniach przenośnych oraz smartfonach. Muszą być odpowiednio zoptymalizowane do działania na docelowych urządzeniach, zachowując wysoką wydajność.
- Wszystkie gry tworzone jako komponenty interaktywne ZPE muszą być zgodne z „**Dokumentacją techniczną komponentów interaktywnych**”.
- **Integracja z kontami ZPE:** w miarę możliwości wszystkie gry będące komponentami interaktywnymi ZPE powinny wykorzystywać system kont użytkowników ZPE. Gry muszą wspierać logowanie poprzez konta ZPE oraz umożliwiać przechowywanie postępów graczy na platformie, co może obejmować API lub inny dostępny mechanizm autoryzacji oferowany przez ZPE. **W miarę możliwości, gry desktopowe powinny być zintegrowane z kontami ZPE.**

9. Zgodność z wytycznymi ZPE

- Gry muszą być zgodne z wszystkimi wytycznymi technicznymi ZPE dotyczącymi integracji, zarządzania danymi użytkowników, bezpieczeństwa oraz ochrony prywatności.
- Gry, które są tworzone jako komponenty interaktywne ZPE, muszą spełniać techniczne wymagania tej platformy, w tym brak konieczności ładowania danych z zewnętrznych źródeł. Dodatkowo zaleca się możliwość współpracy z mechanizmami edukacyjnymi ZPE, takimi jak przypisywanie wyników do kont użytkowników, raportowanie postępów i konfiguracja przez nauczyciela.

10. Zarządzanie kontami ZPE

- Gry, które są zintegrowane z systemem kont użytkowników ZPE, muszą zapewniać płynne przechowywanie i synchronizację postępów gracza oraz spełniać wymagania platformy ZPE w zakresie dostępu i autoryzacji.

11. Użycie edytora

- Osadzanie gier oraz materiałów na platformie ZPE wykonane zostanie za pomocą edytora oraz narzędzi opisanych w „**Dokumentacji technicznej komponentów interaktywnych**”.

12. Silnik gry

- Wybrany silnik (np. Unity, Unreal Engine, PlayCanvas, Godot) musi zapewniać wysoką wydajność oraz wsparcie dla dynamicznej grafiki 2D/3D oraz umożliwiać dynamiczne zmiany w otoczeniu, oferować elastyczność w zakresie animacji postaci, interakcji w środowisku oraz wsparcie dla rozwoju na komputery i inne platformy.
- Silnik musi wspierać technologie otwarte (np. OpenGL, Vulkan), aby umożliwić działanie gry na różnych platformach.
- Jeżeli przewidziano w scenariuszu, silnik musi umożliwiać zarówno tworzenie gry w trybie standardowym, jak i jej przyszłe rozwinięcie do wersji VR.

13. Audio i dźwięk

- **Format dźwięków:** wszystkie dźwięki muszą być dostarczone w formacie WAV lub MP3 w wysokiej jakości (minimum 16-bit, 44.1kHz).
- **Dźwięk przestrzenny:** wersja standardowa musi zawierać dźwięki przystosowane do implementacji systemu przestrzennego audio (3D sound), co jest kluczowe dla przyszłej wersji VR.
- **Nagrania głosowe:** głosy postaci muszą być nagrane w profesjonalnej jakości, bez zakłóceń, z uwzględnieniem różnych wersji językowych gry.

14. Optymalizacja wydajności

- Gra musi działać płynnie na średniej klasy komputerach (4 GB RAM, zintegrowana karta graficzna z obsługą DirectX 11, procesor Intel i5 ósmej generacji lub odpowiednik).
- Dynamiczne skalowanie rozdzielczości i jakości tekstur powinno być zaimplementowane, aby zapewnić płynność działania.
- Minimalna liczba klatek na sekundę to 60 FPS na średnich ustawieniach.
- Gra musi być również skalowalna w górę, aby w pełni wykorzystać możliwości mocniejszych urządzeń.
- Należy dokonać optymalizacji plików instalacyjnych, aby gra nie zajmowała zbyt dużo przestrzeni dyskowej na komputerach użytkowników.
- Gra musi być zoptymalizowana pod kątem minimalnych czasów ładowania oraz płynnego działania, szczególnie podczas rozwiązywania zagadek i przechodzenia między różnymi lokacjami.
- Optymalizacja dla mobilnych urządzeń (dotyczy gier WEB):
 - **Responsive Design:** gra musi działać zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych. Interfejs musi być zaprojektowany w sposób, który pozwala na intuicyjne sterowanie dotykowe na mniejszych ekranach (np. tablety, smartfony).
 - **Touch Controls:** wersja mobilna musi obsługiwać sterowanie dotykowe (przesuwanie palcem, kliknięcie, przeciąganie), przy czym gra na desktopie powinna wspierać również tradycyjne sterowanie myszą i klawiaturą.
 - Gry WEB powinny wykorzystywać techniki kompresji zasobów, takie jak WebP dla obrazów, preloading assetów oraz lazy loading, aby zapewnić płynność działania w przeglądarkach.

15. Przygotowanie pod różne rozdzielczości

- Interfejs i grafika muszą być skalowalne do różnych rozdzielczości ekranów – od smartfonów po monitory 4K.
- Każda gra powinna automatycznie dostosowywać się do zmiany rozdzielczości, z uwzględnieniem odpowiedniej wielkości tekstów i elementów interaktywnych.

16. Responsywność

- Gry muszą być responsywne, zarówno na urządzeniach dotykowych, jak i w tradycyjnym sterowaniu na PC.
- System sterowania powinien obsługiwać dotyk, mysz i klawiaturę, a także inne urządzenia wejściowe (np. gamepady, gdy jest to wskazane).

17. Zabezpieczenia i stabilność

- Gry muszą być stabilne, z minimalnym ryzykiem błędów, które mogłyby uniemożliwić dalszą rozgrywkę.
- Systemy zapisywania postępu muszą być bezpieczne i niezawodne, zapewniając ciągłość rozgrywki nawet po ponownym uruchomieniu gry.

18. System zapisu gry

- Gra musi posiadać zarówno automatyczny, jak i manualny system zapisu.
- Obsługa synchronizacji zapisów z chmurą (np. Steam Cloud) powinna być brana pod uwagę jako opcja przyszłej rozbudowy.
- Gra musi obsługiwać automatyczny i manualny system zapisu stanu rozgrywki. Wszystkie dane zapisu powinny być przechowywane w sposób zgodny z wymaganiami platformy, na której gra działa:
 - w przypadku gier desktopowych, pliki zapisu muszą być przechowywane lokalnie, w standardowym katalogu użytkownika (np. AppData dla Windows);
 - w przypadku gier WEB, stan gry powinien być zapisywany w chmurze lub w profilu użytkownika ZPE, jeśli jest to technicznie możliwe i zgodne z zasadami bezpieczeństwa. W przypadku gier multiplayer, system zapisu stanu gry musi umożliwiać synchronizację z serwerem, aby zapewnić ciągłość rozgrywki.

19. Zarządzanie zasobami

- Należy zadbać o odpowiednie zarządzanie zasobami graficznymi, dźwiękowymi i pamięcią w trakcie rozgrywki, aby gra była płynna i nie przekraczała optymalnych limitów użycia pamięci na różnych urządzeniach.
- Ważne jest zachowanie równowagi między jakością a wydajnością gry.
- Zasoby graficzne, dźwiękowe i tekstowe powinny być zorganizowane w struktury katalogów z opisami, co ułatwi ich edycję i aktualizację.
- Wszystkie zasoby muszą być odpowiednio skompresowane, aby minimalizować zajmowaną przestrzeń dyskową.

20. Integracja z usługami zewnętrznymi

- Gra musi być gotowa do dystrybucji na platformach cyfrowych takich jak Steam. Beneficjent konkursowy zobowiązany jest dostarczyć wszystkie wymagane pliki instalacyjne oraz materiały promocyjne (zrzuty ekranu, opisy, grafiki) zgodne z wytycznymi wybranych platform.

- Wymagana jest integracja z systemami DRM (Digital Rights Management), w celu ochrony przed piractwem i nieautoryzowanym dostępem.
- Wersja instalacyjna gry musi być gotowa do publikacji na platformach cyfrowych, spełniając wszystkie techniczne wymagania platform (np. Steam). Ponadto, gra powinna wspierać mechanizmy umożliwiające integrację z systemem ZPE, w tym opcję synchronizacji postępów, osiągnięć oraz aktualizacji wersji gry w sposób analogiczny do wymagań platform dystrybucji cyfrowej.
- Beneficjent konkursowy musi zapewnić możliwość przyszłych aktualizacji gry, poprawek i rozszerzeń (DLC), które można wprowadzać bez potrzeby większych zmian w architekturze gry.

21. Bezpieczeństwo danych

- Gra powinna zapewniać bezpieczne przechowywanie danych graczy, w tym postępów w grze i ustawień. Jeśli gra wymaga połączenia z siecią (np. dla aktualizacji czy multiplayera), dane te muszą być odpowiednio chronione.

22. Dostępność w trybie offline

- Wszystkie gry desktopowe muszą oferować pełną rozgrywkę w trybie offline, bez konieczności łączenia się z zewnętrznymi serwerami (poza opcjonalnymi funkcjami online, jeśli dotyczy).

23. Niezależność od zewnętrznych źródeł danych

- Gry nie mogą wymagać zewnętrznych połączeń do danych spoza środowiska ZPE, aby zagwarantować pełną kontrolę nad treścią oraz bezpieczeństwem użytkowników.

24. Optymalizacja dla przeglądarek (dotyczy gier WEB)

- Gra musi być zoptymalizowana pod kątem działania w popularnych przeglądarkach (Chrome, Firefox, Safari, Edge). Wydajność gry na każdej z przeglądarek powinna być przetestowana, aby zapewnić płynność rozgrywki na różnych urządzeniach.
- Gra musi działać zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych. Interfejs musi być zaprojektowany w sposób, który pozwala na intuicyjne sterowanie dotykowe.
- Wspieranie starszych przeglądarek: w ramach komponentu interaktywnego ZPE gra powinna być kompatybilna także z przeglądarkami o starszych wersjach, aby zachować dostępność dla szerokiej grupy odbiorców.

25. Testowanie i debugging

- Gra musi przejść pełen cykl testów stabilności na różnych konfiguracjach sprzętowych oraz systemowych (w tym Windows 10 i nowsze).
- Każda wersja kodu musi przejść testy jednostkowe oraz regresyjne przed wdrożeniem. W przypadku gier przeglądarkowych wymagane jest testowanie pod kątem optymalizacji zasobów i wydajności renderowania w różnych przeglądarkach.

- Beneficjent konkursowy musi przeprowadzić testy kompatybilności z różnymi rozdzielczościami i ustawieniami graficznymi, aby zapewnić stabilność rozgrywki na szerokiej gamie sprzętowej.
- System testów automatycznych musi obejmować kluczowe mechaniki gry (np. interakcje, narrację, łamigłówki) oraz wydajność na różnych konfiguracjach sprzętowych.
- Testy powinny obejmować funkcjonalność i bezpieczeństwo danych, eliminując błędy krytyczne, problemy z wydajnością, stabilnością oraz mechaniką rozgrywki.
- Automatyczne testy muszą obejmować co najmniej 80% kluczowych funkcji gry, w tym interakcje, narrację i łamigłówki. Raporty z testów należy dostarczyć w formacie ustandaryzowanym, ułatwiającym interpretację wyników przez zespół ekspertów ORE, np. PDF, XML lub DOCX.

26. Repozytorium kodu i dokumentacja techniczna

- **Beneficjent konkursowy jest zobowiązany do dostarczenia pełnego repozytorium kodu źródłowego gry oraz powiązanych zasobów, zgodnie z poniższymi wymaganiami:**
 - **Repozytorium kodu**
 1. Cały kod źródłowy musi być przechowywany w systemie kontroli wersji (np. Git, Mercurial), a każda wersja gry powinna być archiwizowana.
 2. Kod źródłowy musi być przechowywany w repozytorium z obsługą **wersjonowania i automatycznego testowania (CI/CD)**. Wymagane jest stosowanie **oddzielnych gałęzi dla testów oraz stabilnych wersji produkcyjnych**. Przed wdrożeniem nowych funkcji powinna być możliwość automatycznej walidacji integralności kodu.
 3. Repozytorium musi zawierać historię commitów i dokumentację zmian w kodzie (wymagana czytelna historia commitów z opisami).
 4. Struktura projektu powinna być czytelna i modularna, z podziałem na główne komponenty gry (np. silnik gry, UI, baza danych).
 5. Beneficjent konkursowy jest zobowiązany do przekazania instrukcji dotyczących buildowania i uruchamiania gry, w tym listy zależności (np. wersje bibliotek, wymagane pakiety).
 6. Struktura katalogów powinna być ujednolicona i dobrze udokumentowana. Zaleca się oddzielenie zasobów (grafika, dźwięk), kodu źródłowego oraz plików konfiguracyjnych, co ułatwi przyszłe aktualizacje i rozwój gry.
 7. Kod powinien zawierać komentarze oraz dokumentację API kluczowych modułów.

- **Dokumentacja techniczna**
 1. Pełna dokumentacja kodu źródłowego (opis architektury, struktura plików, zależności między komponentami).
 2. Diagramy UML dla głównych systemów gry (np. schematy baz danych, interakcje między modułami).
 3. Opis mechanizmów zarządzania zasobami i optymalizacji.
 4. Instrukcja wdrażania i konfiguracji gry (dotyczy zarówno wersji desktopowych, jak i webowych).
 5. Dokumentacja procesów testowania i debugowania (wymagane narzędzia, zakres testów, raporty wydajnościowe).

27. Zgodność z przepisami dotyczącymi ochrony danych

- Gry muszą być zgodne z przepisami dotyczącymi ochrony danych osobowych, w tym z RODO w przypadku przetwarzania danych użytkowników z UE.

28. Multiplayer i dodatkowa infrastruktura

- W przypadku gier multiplayer, konieczne jest zapewnienie zewnętrznej infrastruktury obsługującej połączenia w czasie rzeczywistym między graczami, ponieważ ZPE nie wspiera natywnie multiplayer.
 - Na etapie opracowywania harmonogramu produkcji danej gry beneficjent konkursowy musi zadeklarować potrzeby w zakresie infrastruktury (np. serwery, usługi chmurowe) wraz z oszacowaniem kosztów ich utrzymania.
 - Do 6. miesiąca od rozpoczęcia projektu, beneficjent konkursowy jest zobowiązany do przekazania szczegółowej dokumentacji dotyczącej infrastruktury potrzebnej do utrzymania gry, w tym jej konfiguracji, kosztów oraz specyfikacji technicznej.
 - Wszystkie usługi muszą być zgodne z RODO, szczególnie jeśli dotyczą przechowywania danych użytkowników (np. stany gry, konta graczy).
- Dla gier WEB komponenty multiplayer muszą być zgodne z „Dokumentacją techniczną komponentów interaktywnych”, a mechanizmy połączeń w czasie rzeczywistym muszą być testowane pod kątem wydajności i stabilności.

29. Wsparcie powdrożeniowe

- **Beneficjent konkursowy jest zobowiązany do świadczenia wsparcia technicznego przez 12 miesięcy od momentu publikacji na ZPE finalnej wersji gry.**
- **Wsparcie powdrożeniowe obejmuje:**
 - usuwanie błędów krytycznych i awarii uniemożliwiających korzystanie z gry;
 - optymalizację wydajności gry w zakresie poprawy stabilności i płynności działania;

- aktualizację komponentów w przypadku niezgodności ze zmianami systemowymi (np. nowe wersje systemów operacyjnych, przeglądarek internetowych);
 - wsparcie w zakresie instalacji i konfiguracji gry, jeśli zajdzie taka potrzeba.
- **Wsparcie powdrożeniowe nie obejmuje:**
- dodawania nowych funkcji i mechanik do gry poza zakresem uzgodnionym w dokumentacji konkursowej;
 - modyfikacji treści edukacyjnych oraz aktualizacji scenariuszy;
 - rozbudowy trybów rozgrywki i dostosowywania systemów do nowych wymagań użytkowników.
- **Każda aktualizacja** w ramach wsparcia powdrożeniowego **powinna być dokumentowana**, a zgłoszenia błędów rejestrowane w systemie śledzenia zgłoszeń (np. JIRA, Redmine, GitHub Issues).

Odstępstwa od ogólnych wymagań

Możliwość modyfikacji wymagań w uzasadnionych przypadkach

W wyjątkowych sytuacjach, gdy ogólne wymagania funkcjonalne i techniczne mogłyby ograniczać walory edukacyjne gry, dopuszcza się możliwość ich dostosowania. Każda zmiana musi być uzasadniona i zatwierdzona w drodze trójstronnej konsultacji z udziałem beneficjenta konkursowego, eksperta merytorycznego ORE oraz informatyka ds. integrowania e-materiałów pod względem technologicznym po stronie projektu niekonkurencyjnego. Odstępstwa nie mogą wpływać negatywnie na dostępność, kompatybilność oraz jakość techniczną gry.

Decyzja o zmianie wymagań musi zostać udokumentowana w formie pisemnej (np. protokołu lub raportu), zawierającego uzasadnienie oraz rekomendacje ekspertów ORE.

Dodatkowe informacje

Współdzielenie zasobów i optymalizacja budżetu

Beneficjent konkursowy, tworząc gry edukacyjne, ma możliwość współdzielenia zasobów cyfrowych, takich jak modele 3D, grafiki, elementy UI, animacje, efekty dźwiękowe czy fragmenty kodu, pomiędzy różnymi grami w ramach danego obszaru, z wyłączeniem zasobów unikalnych dla konkretnych funkcjonalności.

Korzyści płynące ze współdzielenia zasobów:

- **Optymalizacja budżetu** – ograniczenie kosztów produkcji poprzez ponowne wykorzystanie już stworzonych elementów.
- **Skrócenie czasu realizacji** – szybsze wdrażanie gier poprzez użycie gotowych elementów.

Warunki współdzielenia zasobów:

- Wszystkie współdzielone zasoby muszą być kompatybilne z wymogami technicznymi i funkcjonalnymi określonymi dla aplikacji.
- Elementy wizualne i interaktywne powinny zachować taką samą jakość i dostosowanie do wymagań dostępności (WCAG).
- Współdzielenie nie może ograniczać unikalności i walorów edukacyjnych poszczególnych gier – każda gra powinna mieć indywidualne treści i funkcjonalności merytoryczno-dydaktyczne.

Dopuszczenie gry do użytku

Aplikacja zostanie dopuszczona do użytku po spełnieniu wszystkich wymagań:

| Kryterium | Spełnia (TAK/NIE) |
|--|--------------------------|
| Gra działa na docelowych platformach | |
| Gra jest zgodna z GDD | |
| UI/UX zgodne z wytycznymi | |
| Testy funkcjonalne zakończone pozytywnie | |
| Optymalizacja pod kątem wydajności | |
| Gra spełnia wymagania WCAG | |
| System aktualizacji działa poprawnie | |
| Wszystkie zasoby są zawarte w paczce instalacyjnej | |

Wyjaśnienie pojęć

Główna oś gameplay

Główna oś gameplay jest kluczowym elementem, który nadaje sens i kształtuje doświadczenie gracza w danej grze. Termin „główna oś gameplay” odnosi się do kluczowego elementu lub mechaniki w grze, której doświadczenie koncentruje się wokół konkretnej koncepcji lub celu. Jest to fundamentalna część rozgrywki, która definiuje, jak gracze będą oddziaływać z grą i jakie będą mieli cele.

Filary gry

Filary gry to kluczowe elementy, zasady lub aspekty, które stanowią fundament danej gry i mają istotny wpływ na doświadczenie gracza. Są to podstawowe elementy, na których opiera się cała struktura i koncepcja danej gry.

Mechaniki gry

Mechaniki gry to zestaw zasad, interakcji i systemów, które kształtują sposób, w jaki gra działa i jak gracze oddziałują z nią. Są to fundamentalne elementy, które stanowią podstawę rozgrywki.