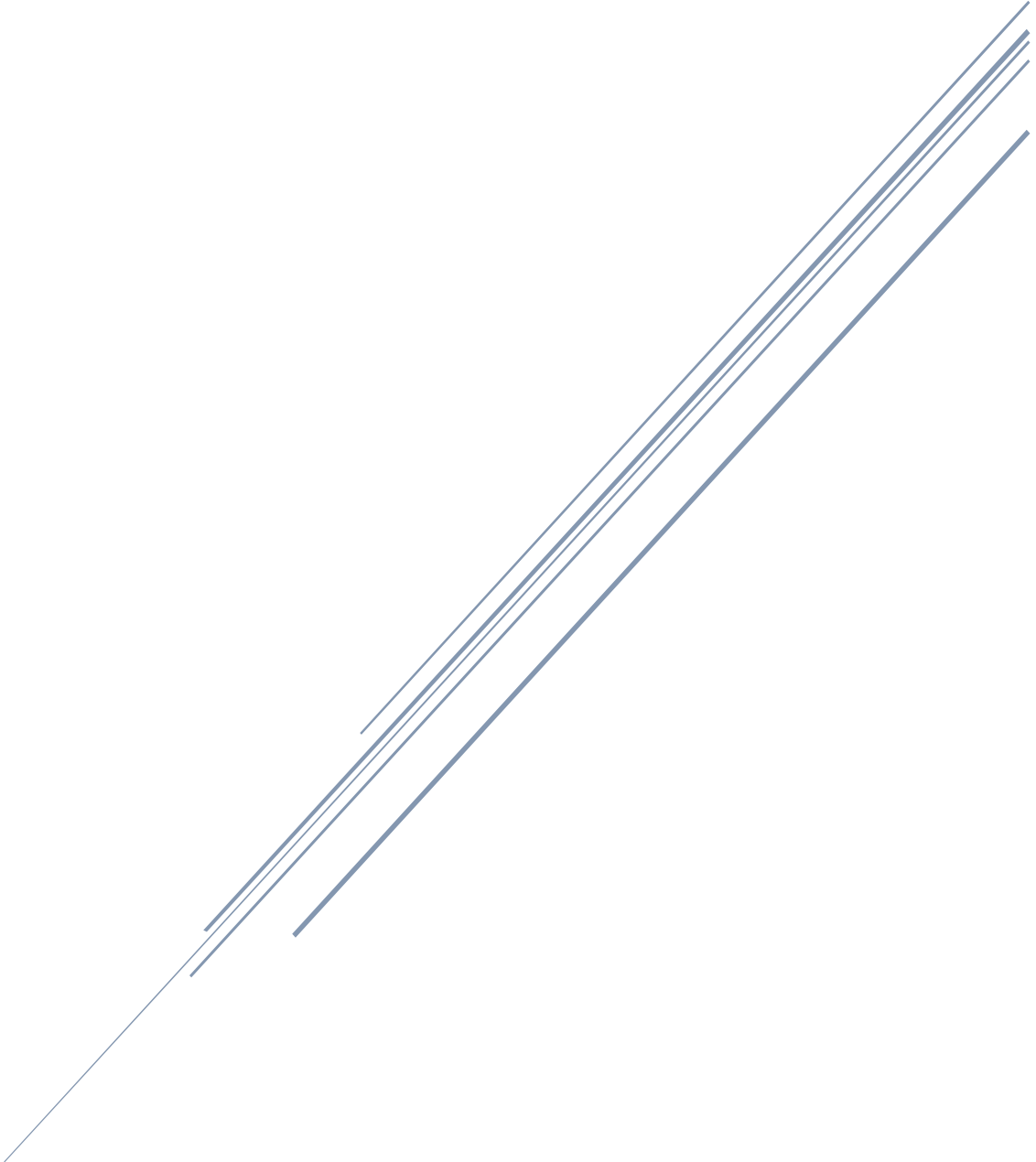


# Cechy wyróżniające projekt

Coin Hunter



Uniwersytet Jagielloński, Wydział Matematyki i Informatyki  
Inżyniera Oprogramowania

# Spis treści

1. Wstęp.....	2
2. Cechy wyróżniające .....	3
3. Spełnienie kryteriów oceny projektu.....	4
3.1. Opis projektu od strony biznesu, tzn. wymagań biznesowych (co klient chce osiągnąć poprzez wykorzystanie tego produktu).....	4
3.2. Przełożenie wymagań biznesowych na wymagania użytkownika, a następnie - wymagań użytkownika na wymagania funkcjonalne .....	4
3.3. Zadbanie o wymagania niefunkcjonalne (wydajność, użyteczność, przenaszalność i inne charakterystyki np. z modelu ISO 25010) .....	4
3.4. Jakość specyfikacji wymagań, kompletność, brak błędów typu: niejednoznaczności, sprzeczności, braki, nadmiarowość.....	4
3.5. Jakość projektu architektury: dobre praktyki (interfejsy, odwracanie zależności, dziedziczenie, wzorce projektowe itd...) .....	5
3.6. Jakość projektu niskopoziomowego i jego implementacji (zasady SOLID, czysty kod, adaptacyjny kod) .....	5
3.7. Jakość i raportowanie testów .....	5
3.8. Prezentacja demo projektu pod koniec semestru .....	5
3.9. Wymogi formalne (dokumentacja, jakość i kompletność) .....	5

# 1. Wstęp

Niniejszy dokument przedstawia spis cech wyróżniających projekt „*Coin Hunter*” oraz listę powodów, dla których powinien on otrzymać maksymalną liczbę punktów.

## **2. Cechy wyróżniające**

Projekt „Coin Hunter” powstał dzięki współpracy osób za niego odpowiedzialnych, ich organizacji oraz pełnemu zaangażowaniu. Poniżej przedstawiono listę cech, którymi projekt się wyróżnia.

## **3. Spełnienie kryteriów oceny projektu**

### **3.1. Opis projektu od strony biznesu, tzn. wymagań biznesowych (co klient chce osiągnąć poprzez wykorzystanie tego produktu)**

W dokumentacji zostały przedstawione różnorodne aspekty obecności produktu na rynku, a także zagrożenia i korzyści wynikające z jego specyfiki. Zostały one uzasadnione analizą konkurencyjnych rozwiązań oraz potencjału branży. Wnikliwie przeprowadzona analiza biznesowa umożliwia klientowi podejmowanie świadomych i przemyślanych decyzji odnośnie zarządzania implementowanym przez nas projektem.

### **3.2. Przełożenie wymagań biznesowych na wymagania użytkownika, a następnie - wymagań użytkownika na wymagania funkcjonalne**

Zrealizowane zostały główne wymagania wyróżniające produkt na rynku, t.j. implementacja warstwy społecznościowej jako integralnej części gry, oraz możliwość zapisania swoich postępów. Możliwe jest korzystanie z aplikacji bez konieczności pobierania dodatkowych programów. Dzięki sukcesywnej implementacji wymagań biznesowych wyróżniających nasz produkt na rynku prawdopodobieństwo osiągnięcia sukcesu (określonego kryteriami zawartymi w pliku "Dokument wizji i zakresu") określamy jako duże.

### **3.3. Zadbanie o wymagania niefunkcjonalne (wydajność, użyteczność, przenaszalność i inne charakterystyki np. z modelu ISO 25010)**

Aplikacja stworzona została według powszechnie przyjętych standardów projektowania gier przeglądarkowych w JavaScript, co zapewnia jej wydajność i efektywność. Taki model rozwiązania z racji swojej powszechności jest dobrze sprawdzony i łatwo utrzymywalny. Aplikacja wykazuje się wysoką przenaszalnością z racji słabej zależności od środowiska, w którym jest uruchamiana (aplikacja przeglądarkowa).

### **3.4. Jakość specyfikacji wymagań, kompletność, brak błędów typu: niejednoznaczności, sprzeczności, braki, nadmiarowość**

Wymagania przyjęte przez zespół są spójne, pokrywają różnorodne aspekty projektu, dzięki czemu uzupełniają się wzajemnie. Nie wykryto sprzeczności, niejednoznaczności, nadmiarowości ani braków w specyfikacji wymagań.

### **3.5. Jakość projektu architektury: dobre praktyki (interfejsy, odwracanie zależności, dziedziczenie, wzorce projektowe itd...)**

Patrz niżej.

### **3.6. Jakość projektu niskopoziomowego i jego implementacji (zasady SOLID, czysty kod, adaptacyjny kod)**

Przykłady zastosowania zasad SOLID oraz wzorców projektowych zostały przedstawione w pliku „Metodologia, architektura i wzorce” w folderze „Documentation”. Kod jest przejrzysto zorganizowany, posiada standardowo praktykowaną strukturę plików i katalogów. Spełniona jest zasada DRY, klasy posiadają adekwatną hermetyzację, a nazwy zmiennych i metod są sugestywne oraz łatwe do rozróżnienia. Funkcje wielozadaniowe rozdzielone zostały na mniejsze metody, dzięki czemu zachowana jest czytelność i przejrzystość. Dzięki zastosowaniu klas abstrakcyjnych oraz dziedziczenia kod jest łatwo rozszerzalny o dodatkowe funkcjonalności.

### **3.7. Jakość i raportowanie testów**

Wykonane zostały wszystkie typy testów (jednostkowe, integracyjne, systemowy, akceptacyjne). Dokumentacja testowa jest kompletna i należy również zwrócić uwagę na różnorodność i ilość przeprowadzonych testów. Szczegółowy i kompletny proces testowy umożliwił wyeliminowanie kluczowych błędów, dzięki czemu produkt dostarczony do klienta jest jakościowy, wolny od defektów oraz spełnia zadane mu wymagania.

### **3.8. Prezentacja demo projektu pod koniec semestru**

Prezentacja projektu Coin Hunter wyróżniała się zastosowaniem środków multimedialnych (prezentacja PowerPoint), jak również zawarciem szczegółowego opisu technicznego wykorzystywanych technologii. Zadbano także o detale wizualne, takie jak ogólna przejrzystość oraz nawiązania do stylistyki gry. W prezentacji zawarto krótkie wypowiedzi reprezentantów każdego z zespołów tworzących poszczególne części projektu. Dodatkowym atutem prezentacji była możliwość przetestowania gotowego produktu przez uczestników (demonstracja działania gotowej aplikacji).

### **3.9. Wymogi formalne (dokumentacja, jakość i kompletność)**

Dokumentacja projektu powstawała przy ciągłej komunikacji z zespołem programistów. Jest kompletna, zawiera detaliczny i poprawny opis zastosowanych rozwiązań. Całość dostarczonej dokumentacji jest spójna stylistycznie, przejrzysta oraz zorganizowana w sposób umożliwiający łatwą i szybką nawigację. Wyczerpujący opis pełnej gamy aspektów projektu dostarcza wszelkich potrzebnych informacji na temat kodu, zastosowanych rozwiązań oraz weryfikuje wywiązanie się z postawionych zespołowi wymagań.