|  |  |  |
| --- | --- | --- |
| Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej | **Narzędzia Pracy Grupowej** | 2022/2023 |
| **Skład grupy:**  Bartosz Pajor  Artur Olszówka  Kamil Nowobilski  Kacper Pelczarski | **Temat:**  Blackjack | **Nr Tematu:**  18 |

Cel projektu: Blackjack

Blackjack, znany również jako "21", to popularna gra karciana, w której celem jest zdobycie jak najbliżej (lub dokładnie) 21 punktów, nie przekraczając tej liczby. Gra odbywa się pomiędzy graczem a krupierem.

Zasady Blackjacka są dość proste. Każda karta ma określoną wartość punktową: karty od 2 do 10 mają wartość równą swojemu numerowi, walet, dama i król mają wartość 10 punktów. As może mieć wartość równą 1 lub 11, w zależności co jest lepsze dla gracza. Podczas rozdania dwie karty są rozdawane zarówno graczowi, jak i krupierowi. Obie karty gracza są widoczne, podczas gdy krupier odsłania tylko jedną kartę, pozostawiając drugą zakrytą. W tym momencie gracz ma dwie opcje do wyboru dla swojego kolejnego ruchu:

1. Może dobrać kolejną kartę, aby zwiększyć swoje szanse na zbliżenie się do 21 punktów.
2. Może pozostać przy swojej obecnej sumie punktów, jeśli uważa, że jest wystarczająco blisko 21 punktów i nie chce ryzykować dobierania kolejnych kart, które mogą przekroczyć tę wartość.

Backlog produktu:

1. Przygotowanie środowiska:
   1. Skonfigurować środowiska pracy grupowej oraz PyCharm.
   2. Zainstalować niezbędne bibloteki.
2. Definicja wymagań
   1. Sprecyzować funkcjonalność gry.
   2. Określić założenia dotyczące zachowań kart, tali, gracza i krupiera/
3. Stworzenie interfejsu graficznego
   1. Stworzenie okna i ustalenie jego rozmiarów.
   2. Stworzenie klasy Button
   3. Dodanie przycisku w menu „Graj”, „Reset” oraz „twórcy”.
4. Stworzenie głównej pętli programu
   1. Stworzyć funkcję odpowiedzialną za grę właściwą
   2. Stworzyć implementacją przycisków „Graj”, „Reset”.
5. Stworzyć klasy Talia oraz Karta
   1. Stworzyć metody odpowiedzialne za dodawanie oraz wyświetlanie kart
   2. Stworzyć metodę odpowiedzialną za „losowanie kart”
   3. zaimplementować metody pozwalające dynamicznie wczytywać
   4. Zdjęcia kart, które zostały wylosowane.
6. Stworzenie interfejsu odpowiedzialnego za „główną rozgrywkę”
   1. Stworzyć przyciski odpowiedzialne za dobór kart Gracza oraz za „Pass”
   2. Wyświetlić w odpowiednim miejscu wylosowane karty.
7. Stworzyć Krupiera
   1. Implementacja mechaniki dobierania kart przez krupiera
8. Mechanika rozstrzygająca o Zwycięstwie
   1. Stworzyć funkcję porównującą, ilość punktów gracza.
   2. Wywoływanie funkcji zwycięstwa porażki lub remisu.
9. Poprawki i udoskonalenia programu.
   1. Wprowadzenie licznika punktów, w czasie rzeczywistym
   2. Dodanie suwaka wskazującego liczbę talii wykorzystywanych podczas rozgrywki
   3. Ustalenie ścieżek względnych dla zdjęć talii
   4. Poprawienie czytelność kodu
10. Przygotowanie dokumentacji

10.1.Podliczenie statystyk pracy.

10.2.Przygotowanie sprawozdania indywidualnego.

10.3.Przygotowanie sprawozdania zbiorowego.

Opis organizacji pracy:

Podczas realizacji projektu wykorzystaliśmy różne narzędzia udostępnione przez platformy Asana i Github. Zadania zostały zrealizowane podczas 4 sprintów, dla każdego z nich przypisaliśmy innego scrum mastera.

**Pierwszy sprint**: Scrum masterem tego sprintu został Bartosz Pajor. W trakcie tego sprintu ustaliliśmy narzędzia, z których będziemy korzystać podczas projektu, zapoznaliśmy się z zasadami gry w Blackjack. Na ten sprint zostały ustalone dwa kamienie milowe: podzielenie zadań oraz stworzenie pierwszego programu gry Blackjack. Podczas sprintu nie wystąpiły żadne komplikacje i oba kamienie milowe osiągnęliśmy.

**Newsletter pierwszego sprintu:**

Scrum Master- Bartosz Pajor

Cele:

1. Założenie repozytorium w GitHub oraz projektu w Asanie
2. Zaznajomienie się z zasadami gry w Blackjack
3. Podzielenie zadań

Kamienie milowe:

Utworzenie repozytorium oraz tablicy. Stworzenie pierwszego pliku projektu.

Zadania:

1. Założenie repozytorium w GitHub oraz dodanie pozostałych członków zespołu- Bartosz Pajor
2. Utworzenie tablicy w Asanie oraz dodanie pozostałych członków zespołu- Kacper Pelczarski
3. Wspólne zapoznanie z zasadami gry Blackjack- wszyscy
4. Stworzenie projektu

Zebrania:

Podczas tego sprintu zebraliśmy się wspólnie dwukrotnie. Pierwsze spotkanie dotyczyło omówienia ogólnych zasad gry, rozdzielenia zasad oraz organizacji pracy zespołu. Na drugim zebraniu przedyskutowano rozwiązania programistyczne, z których będziemy korzystać przy wykonaniu projektu.

**Drugi sprint:** Scrum masterem tego sprintu został Kamil Nowobilski.

W trakcie tego sprintu ustaliliśmy narzędzia, z których

będziemy korzystać podczas projektu, zapoznaliśmy się z

dostępnymi funkcjami dostępnymi w ramach biblioteki PyGame.

Na ten sprint zostały ustalone dwa kamienie milowe: Stworzenie głównego

zarysu programu oraz stworzyć dobrze działającego menu.

Podczas sprintu nie wystąpiły żadne komplikacje i oba kamienie

milowe zostały osiągnięte.

**Newsletter drugiego sprintu:**

Scrum Master – Kamil Nowobilski

Cele:

1. Zapoznanie towarzyszy odnośnie działania i funkcjonowania biblioteki PyGame.
2. poinstruowanie jak program ma wyglądać.

Kamienie milowe:

Stworzenie planu programu i jego poszczególnych funkcji.

Zadania:

1. Wyszukiwanie odpowiedniej dokumentacji biblioteki PyGame - Kamil Nowobilski
2. Zapoznanie się z jej treścią - Wszyscy
3. Stworzenie menu oraz zmiennych potrzebnych do prawidłowego funkcjonowania programu - Kamil Nowobilski
4. pierwsze uruchomienie programu - Bartek Pajor
5. efekty dźwiękowe - Artur Olszówka
6. Zdefiniowanie klas programu - Kacper Pelczarski

Zebrania:

Podczas tego sprintu zebraliśmy się w sumie raz. Na spotkaniu ustalono zadania, omówiono szczegóły realizacji i zidentyfikowano potencjalne przeszkody w trakcie sprintu. Omówiono wybór elementów interfejsu API do wykorzystania w programowaniu i ostateczny wygląd aplikacji. Po podziale zadań i wyznaczeniu kluczowych oraz krótkoterminowych celów spotkanie uznano za zakończone.

**Trzeci sprint** : Scrum masterem tego sprintu został Artur Olszówka. W trakcie tego sprintu rozpoczęliśmy początkowe prace nad programem. Na ten sprint zostały ustalone dwa kamienie milowe: Stworzenie talii i zaprojektowanie funkcji z nią związanych. Podczas sprintu wystąpiły drobne problemy, przez co część funkcji musieliśmy porzucić.

**Newsletter trzeciego sprintu:**

Scrum master – Artur Olszówka

Cele:

1. Rozpoczęcie prac nad programem
2. Stworzenie talii
3. Stworzenie funkcji tworzących rozgrywkę

Kamienie milowe:

Dołączenie grafik kart i nadanie im wartości. Napisanie funkcji wykorzystujących wcześniej utworzoną talię

Zadania:

1. Nadanie wartości poszczególnym kartom – Artur Olszówka
2. Import grafik – Bartosz Pajor
3. Spis funkcji rozgrywki – wszyscy
4. Stworzenie klasy talii – Kacper Pelczarski
5. Dobieranie kart – Kamil Nowobilski

Zebrania:

Podczas tego sprintu zebraliśmy się tylko raz. Samo spotkanie dotyczyło utworzenia podmiotu informatycznego zwanego dalej talią oraz napisanie funkcji, które nadają sens całej rozgrywce. Niestety nie starczyło czasu na napisanie wszystkich zaplanowanych funkcji.

**Czwarty sprint:** Scrum masterem czwartego sprintu został Kacper Pelczarski. Celem była implementacja dalszych funkcjonalności związanych z zasadami gry, animacji oraz klasy przycisku, za kamień milowy uznano uruchomienie działającego programu.

**Newsletter czwartego sprintu:**

Scrum master – Kacper Pelczarski

Cele:

1. Stworzenie klasy Button
2. Implementacja gry w pliku main.py
3. Funkcje gry na czas, zapisu punktów po wyłączeniu programu

Kamienie milowe:

Uruchomienie funkcjonującego programu

Zadania:

1. Stworzenie klasy Button – Kacper Pelczarski
2. Stworzenie menu i logiki programu w main.py – Kamil Nowobilski
3. Statystyki gracza w czasie rzeczywistym – Barosz Pajor
4. Animacje zwycięstwa, remisu i porażki – Artur Olszówka
5. Uruchomienie programu – wszyscy

Zebrania:

Podczas tego sprintu zebraliśmy się dwa razy. Spotkania dotyczyły implementacji poszczególnych funkcji i ich integracji w działający program. Zabrakło czasu na implementację funkcji gry na czas, cofania ruchu oraz efektów dźwiękowych

**Sprawozdania indywidualne:**

**Bartosz Pajor**

Adres e-mail: bapajor@student.agh.edu.pl

**Zrealizowane zadania:**

Projektowe:

* Utworzenie repozytorium w GitHub
* Zapoznanie z zasadami gry Blackjack
* Gra testowa z komputerem
* Pierwsze uruchomienie programu
* Udział w zebraniach

Programistyczne:

* Import wszystkich potrzebnych grafik
* Implementacja ścieżek do wymaganych zdjęć
* Edycja pliku README
* Edycja main.py
* Edycja definicja\_klas.py
* Utworzenie przycisku do wyboru liczby talii

Organizacyjne:

* Zarządzanie pierwszym sprintem
* Utworzenie repozytorium w GitHub
* Dodanie pozostałych członków zespołu do repozytorium
* Przypisanie zadań do poszczególnych członków zespołu

Dokumentacyjne:

* Edycja sprawozdania grupowego
* Wypełnienie sprawozdania indywidualnego
* Uzupełnienie tabeli ze statystykami pracy

|  |  |  |
| --- | --- | --- |
| Aspekt | Parametry | Wkład |
| Role | Scrum Master, Administrator repozytorium, Deweloper |  |
| Kodowanie | Liczba linii kodu | 277 ( 245++ , 32-- |
|  | Funkcje | Import wszystkich potrzebnych grafik  Implementacja ścieżek do wymaganych zdjęć  Edycja pliku README  Edycja main.py  Edycja definicja\_klas.py  Utworzenie przycisku do wyboru liczby talii |
| Repozytorium | Liczba commit-ów | 50 |
|  | Liczba utworzonych gałęzi | 1 |
|  | Gałąź (używana – nazwa) | Bartek (usunięta) |
|  | Liczba połączonych gałęzi | 1 |
| Dokumentowanie | Liczba standup-ów | 4 (1 prowadzony) |
|  | Opisy na Wiki | 0 |
| Aktywność | Liczba zrealizowanych zadań | 11 |
|  | Szacowana liczba godzin | 10 |
|  | Ocena procentowego wkładu | 25% |

**Kamil Nowobilski**

Adres e-mail: [nowobilski@student.agh.edu.pl](mailto:nowobilski@student.agh.edu.pl)

**Zrealizowane zadania:**

Projektowe:

* Dołączenie do przestrzeni roboczej zespołu (GitHub, Asana, Discord).
* Zrozumienie zasad gry w Blackjacka.
* Konfiguracja środowiska programistycznego PyCharm.
* Udział w spotkaniu organizacyjnym.
* Motywowanie zespołu.

Programistyczne:

* Stworzenie głównego menu
* Stworzenie zmiennych do prawidłowego funkcjonowania programu
* Naprawa problemów z importowaniem bibliotek.
* Warunki po doborze 4 karty

Organizacyjne:

* Udzielanie instrukcji dotyczących problemów z biblioteką.
* Dostarczanie linków do dokumentacji odpowiednich bibliotek.
* Pomoc w konfiguracji środowiska Python.

Dokumentacyjne:

* Tworzenie części sprawozdania.
* Edycja grupowego sprawozdania.
* Rejestracja wkładu pracy.
* Poprawa formatowania sprawozdania.

|  |  |  |
| --- | --- | --- |
| Aspekt | Parametry | Wkład |
| Role | Informator, deweloper, sacrum master |  |
| Kodowanie | Liczba linii kodu | **(**210 ++7 --) |
|  | Funkcje (wymienić) | Reformatowanie kodu  Operacja na wszystkich funkcjach z biblioteki pygame. Stworzenie menu oraz głównego zarysu programu.  Stworzenie zmiennych potrzebnych do prawidłowego działania programu, implementacja funkcji stworzonych przez towarzyszy, stworzenie menu oraz w głównej funkcji odpowiedzialnej za grę (level\_one), Sprawdzenie wszystkich warunków po doborze 4 karty krupiera |
| Repozytorium | Liczba commit-ów | 7 |
| Dokumentowanie | Liczba standup-ów | 2 |
|  | Opisy na Wiki | 0 |
| Aktywność | Liczba zrealizowanych zadań | 7 |
|  | Szacowana liczba godzin | 11 |
|  | Ocena procentowego wkładu | 30% |

**Artur Olszówka**

Adres e-mail: [artols@student.agh.edu.pl](mailto:artols@student.agh.edu.pl)

**Zrealizowane zadania:**

Projektowe:

* Zaznajomienie zespołu z zasadami gry w Blackjacka
* Dołączenie do grupy projektowej (Messenger, Asana, Github, Discord)
* Testowanie programu
* Konfiguracja IDE PyCharm
* Wstępny podział obowiązków w grupie projektowej

Programistyczne:

* Inicjalizacja programu
* Funkcja dobierania kart przez gracza
* Funkcja licząca obecną punktację
* Funkcje opisujące wynik rozgrywki
* Zapis statystyk gracza

Organizacyjne:

* Pomoc przy podziale obowiązków między projektantami
* Ewentualne wsparcie przy cudzych zadaniach
* Sumienne wykonywanie narzuconych poleceń

Dokumentacyjne:

* Opis wkładu własnego w projekt
* Edycja sprawozdania grupy projektowej

|  |  |  |
| --- | --- | --- |
| Aspekt | Parametry | Wkład |
| Role | Wymienić | Scrum master, deweloper, bard |
| Kodowanie | Liczba linii kodu | 195 (193 ++, 2 --) |
|  | Funkcje (wymienić) | Napisanie funkcji: wygrales  Napisanie funkcji: przegrales  Napisanie funkcji: remis  Napisanie funkcji: get\_wartosc  Napisanie funkcji: wkzdj  Edycja błędów stylistycznych |
| Repozytorium | Liczba commit-ów | 8 |
|  | Liczba utworzonych gałęzi | 0 |
|  | Gałąź (używana – nazwa) | main |
|  | Liczba połączonych gałęzi | 0 |
| Dokumentowanie | Liczba standup-ów | 3 |
|  | Opisy na Wiki | 0 |
| Aktywność | Liczba zrealizowanych zadań | 4 |
|  | Szacowana liczba godzin | 10 |
|  | Ocena procentowego wkładu | 20% |

**Kacper Pelczarski**

Adres e-mail: pelczarskik@student.agh.edu.pl

**Zrealizowane zadania:**

Projektowe:

* Utworzenie Asany
* Przydzielenie zadań
* Testowanie programu
* Udział w zebraniach

Programistyczne:

* Zdefiniowanie klas programu
* Implementacja logiki gry
* Edycja main.py
* Edycja definicja\_klas.py
* Kontrola czytelności pisanego kodu

Organizacyjne:

* Zarządzanie czwartym sprintem
* Utworzenie Asany
* Przypisanie zadań w systemie zarządzania zadaniami

Dokumentacyjne:

* Edycja sprawozdania grupowego
* Wypełnienie sprawozdania indywidualnego

|  |  |  |
| --- | --- | --- |
| Aspekt | Parametry | Wkład |
| Role | Scrum Master repozytorium, Deweloper |  |
| Kodowanie | Liczba linii kodu | 352++ , 101-- |
|  | Funkcje | Naprawa błędów  Mechanika przycisku  Sprawdzanie warunków po doborze trzech kart krupiera  Edycja main.py  Edycja definicja\_klas.py  Tworzenie klas |
| Repozytorium | Liczba commit-ów | 21 |
|  | Liczba utworzonych gałęzi | 0 |
|  | Gałąź (używana – nazwa) | main |
|  | Liczba połączonych gałęzi | 0 |
| Dokumentowanie | Liczba standup-ów | 3 |
|  | Opisy na Wiki | 0 |
| Aktywność | Liczba zrealizowanych zadań | 5 |
|  | Szacowana liczba godzin | 10 |
|  | Ocena procentowego wkładu | 25% |