Wprowadzenie

Celem projektu „2048” jest odtworzenie oryginalnej gry, z wykorzystaniem podstawowych bibliotek. Projekt ten w głównej mierze bazuje na module „pygame”. Środowisko wykorzystywane do napisania gry to Visual studio 2022. Aby przeprowadzić kompleksową analizę wymagań dla tego projektu, skoncentrujmy się na głównych aspektach funkcjonalnych, struktury danych oraz interakcji między komponentami. Oto podsumowanie kluczowych punktów:

Funkcjonalności

1. Inicjalizacja Pygame i ustawienia początkowe:
    - Inicjalizacja biblioteki Pygame.
    - Definicja stałych gry, takich jak rozmiar ekranu, rozmiar planszy, rozmiar kafelka, kolory tła i kafelków, ikona programu oraz czcionka i wynik.
2. Generowanie nowego kafelka:
    - Losowe dodawanie nowego kafelka (o wartości 2 lub 4) na pustym miejscu na planszy.
3. Sprawdzanie możliwości połączenia kafelków:
    - Sprawdzanie, czy sąsiednie kafelki mogą być połączone (czy mają takie same wartości).
4. Sprawdzanie zakończenia gry:
    - Warunki zakończenia gry, gdy nie ma wolnych miejsc i nie można połączyć żadnych kafelków.
5. Sprawdzanie osiągnięcia 2048:
    - Warunek do wygrania gry, wymagane jest zdobycie kafelka 2048.
6. Aktualizacja wyniku:
    - Zwiększanie wyniku po każdym połączeniu kafelków.
7. Łączenie kafelków:
    - Mechanizmy łączenia kafelków w lewo, prawo, górę lub dół w odpowiedzi na wejście użytkownika.
8. Obsługa wejścia użytkownika:
    - Reakcja na naciśnięcia klawiszy przez użytkownika do sterowania kafelkami.
    - Reakcja na naciśniecie spacji przy ekranie początkowym do rozpoczęcia gry.
    - Reakcja na naciśniecie entera przy ekranie końcowym do zresetowania gry.
9. Rysowanie interfejsu użytkownika:
    - Wyświetlanie ekranu startowego, kafelków, wyniku i komunikatu o zakończeniu gry (wygranej i przegranej).
10. Resetowanie gry:
    - Wyczyszczenie wyniku, kafelków. Przywrócenie gry do stanu startowego.
11. Pętla główna gry:
    - Obsługa zdarzeń, rysowanie interfejsu użytkownika i aktualizacja stanu gry.

Struktura danych

1. **Plansza gry:** Używa tablicy numpy o wymiarach BOARD_SIZE x BOARD_SIZE do przechowywania wartości kafelków. Jest to podstawowa struktura danych, która umożliwia łatwą manipulację i sprawdzanie stanu gry.

2. **Stałe:** Zdefiniowane stałe, takie jak rozmiar ekranu, kolory i inne, używane do konfiguracji wizualnej i logiki gry.
3. **Score:** Zmienna przechowująca aktualny wynik gry.

Interakcje między komponentami

1. Inicjalizacja i UI:
   - W momencie uruchomienia gry inicjalizowane są Pygame, ekran główny, ekran gry oraz początkowy stan planszy. Następnie w pętli głównej gry aktualizowany jest interfejs użytkownika w reakcji na akcje gracza.
2. Obsługa zdarzeń:
   - Główna pętla gry nasłuchuje na zdarzenia, takie jak rozpoczęcie gry, naciśnięcie klawisza, resetowanie gry, zamknięcie okna i odpowiednio reaguje na nie.
3. Logika gry:
   - Funkcje takie jak add_new_tile, can_combine_tiles, is_game_over, has_won, move_tiles i combine_tiles są ściśle powiązane, współpracując w celu aktualizacji stanu gry zgodnie z regułami "2048". Wymagają one ciągłej komunikacji i wymiany danych między sobą, głównie poprzez manipulację globalną tablicą board i zmienną score.
4. Rysowanie UI:
   - Rysowanie stanu gry, takiego jak plansza i wynik, jest bezpośrednio zależne od aktualnego stanu tablicy board i wartości score. Każda zmiana w logice gry jest odzwierciedlana na ekranie za pomocą tej funkcji.