Dokumentacja dla klasy **Turn**, **Forward** oraz funkcji get\_level:

**Klasa Turn**

Klasa Turn dziedziczy po klasie Sprite i reprezentuje zakręt na torze. Jest to specjalny typ sprite'a, który zawiera dodatkowe funkcje związane z obliczeniami geometrycznymi oraz przechowywaniem specyficznych informacji o rodzaju zakrętu. W zależności od typu zakrętu, obliczane są unikalne wartości dla każdego piksela w tym zakręcie.

**Atrybuty:**

* **types**: Słownik definiujący różne typy zakrętów, ich kąty obrotu i punkty miernicze, w zależności od tego, z jakiej strony podchodzi się do zakrętu (np. "NE" oznacza podejście z góry i wyjazd z prawej strony).
* **values**: Słownik, w którym przechowywane są specjalne wartości dla każdego typu zakrętu, obliczone na podstawie punktu mierniczego i specyficznych obliczeń geometrycznych.

**Metody:**

* **set\_pixel\_values(measure\_point, reverse, height, width)**:
  + Funkcja oblicza specjalne wartości geometryczne dla każdego piksela w obrębie zakrętu.
  + **measure\_point** – punkt, względem którego będą liczone wartości (np. (130, 0) dla zakrętu).
  + **reverse** – określa, czy obliczenia mają być odwrócone (dotyczy obrotu).
  + **height i width** – wymiary toru (domyślnie 130x130).
* **\_\_init\_\_(self, window, coords, image, type, id)**:
  + Konstruktor tworzący obiekt zakrętu w oparciu o parametry takie jak okno gry, współrzędne, obrazek zakrętu, typ (np. "NE") oraz identyfikator.
* **scale(self, factor)**:
  + Skaluje zakręt oraz jego hitbox, zachowując jednocześnie jego proporcje.
* **get\_pixel\_values(self, coords)**:
  + Funkcja zwraca unikalne wartości x i y dla danego punktu (koordynaty), obliczając je na podstawie wartości zdefiniowanych w values.

**Klasa Forward**

Klasa Forward również dziedziczy po Sprite i reprezentuje prostą (odcinek toru) w grze. Podobnie jak w przypadku zakrętu, klasa ta zawiera funkcje obliczające specjalne wartości dla każdego piksela na prostym torze.

**Atrybuty:**

* **types**: Słownik definiujący różne typy prostych torów, takie jak "WE" (zachód-wschód), "SN" (północ-południe), itd.
* **all\_types\_values**: Słownik, w którym przechowywane są specjalne wartości dla każdego typu toru (np. "WE", "SN"), obliczone na podstawie unikalnych obliczeń geometrycznych dla tego typu.

**Metody:**

* **set\_pixel\_values(type, width, height)**:
  + Funkcja oblicza wartości dla każdego piksela w obrębie toru. W zależności od typu toru, wartości są obliczane w różny sposób.
* **\_\_init\_\_(self, window, coords, image, type, id)**:
  + Konstruktor tworzy obiekt toru prostego na podstawie parametrów: okno gry, współrzędne, obrazek toru, typ (np. "WE") oraz identyfikator.
* **scale(self, factor)**:
  + Skaluje prostą oraz jej hitbox, zachowując proporcje.
* **get\_pixel\_values(self, coords)**:
  + Funkcja zwraca wartości x i y dla danego punktu, obliczone na podstawie wartości przechowywanych w all\_types\_values.

**Funkcja get\_level(file\_name, window, turn\_image, forward\_image)**

Funkcja ta wczytuje poziom z pliku tekstowego i generuje listę obiektów reprezentujących poszczególne elementy toru (zakręty i proste). Funkcja analizuje zawartość pliku, przetwarza ją i na tej podstawie tworzy odpowiednie obiekty typu Turn lub Forward.

**Parametry:**

* **file\_name** – nazwa pliku z poziomem (np. "level1.txt").
* **window** – okno gry, w którym będą rysowane obiekty.
* **turn\_image** – obrazek reprezentujący zakręt.
* **forward\_image** – obrazek reprezentujący prostą toru.

**Zasada działania:**

1. Funkcja otwiera plik z poziomem, a następnie wczytuje wszystkie linie.
2. Dla każdej linii (reprezentującej jedno pole toru) rozdziela dane na: typ obiektu (zakręt lub prosta), współrzędne (x, y), typ zakrętu/prostej (np. "NE", "WE"), oraz identyfikator.
3. Na podstawie tych danych, tworzy odpowiedni obiekt (Turn lub Forward) i dodaje go do listy.
4. Po zakończeniu przetwarzania pliku zwraca listę wszystkich obiektów.

**Podsumowanie**

* Klasy **Turn** i **Forward** pozwalają na reprezentację elementów toru w grze, zarówno zakrętów, jak i prostych odcinków.
* Każdy obiekt przechowuje informacje o swoim typie, kącie obrotu oraz unikalnych wartościach geometrycznych dla pikseli w obrębie swojego kształtu.
* Funkcja get\_level umożliwia wczytanie poziomu z pliku i generowanie odpowiednich obiektów toru.