**Dokumentacja dla klasy ai**

**Opis ogólny**

Klasa ai dziedziczy po klasie Car i reprezentuje pojazd sterowany przez sztuczną inteligencję (AI) w grze wyścigowej. Klasa ta zawiera logikę decyzyjną, która umożliwia sztucznej inteligencji ocenianie, jak najlepiej poruszać się po torze wyścigowym. Na podstawie oceny różnych punktów toru i analizowania sytuacji, AI podejmuje decyzję o sterowaniu pojazdem.

**Atrybuty:**

* **player**: Obiekt gracza, z którym AI będzie się ścigać lub będzie go starało się wyprzedzić.
* **already\_won**: Flaga, która określa, czy AI już wygrało wyścig. Ustawiana na True, jeśli AI ukończyło wyścig.
* **car\_id**: Identyfikator auta, który określa unikalny typ pojazdu AI. Może mieć różne algorytmy sterowania w zależności od tego identyfikatora.
* **stear**: Kąt, w którym pojazd AI ma skręcać (czyli kierunek, w którym pojazd będzie jechał).
* **joystick\_y**: Określa przyspieszenie lub opóźnienie pojazdu. 1 oznacza przyspieszenie, a -1 opóźnienie. Wartość 0 oznacza brak przyspieszenia.
* **acceleration**: Określa przyspieszenie pojazdu. Zwiększane w zależności od warunków wyścigu.

**Metody:**

1. **get\_real\_point(self, coords)**
   * **Opis**: Funkcja przekształca punkt na ekranie (koordynaty) na odpowiednie współrzędne wirtualnego świata gry, uwzględniając przesunięcie kamery. Umożliwia to prawidłowe przetwarzanie koordynatów po kliknięciu w ekran.
   * **Wejście**:
     + coords – krotka (x, y) reprezentująca punkt na ekranie.
   * **Wyjście**:
     + Zwraca krotkę (new\_x, new\_y), gdzie są to współrzędne wirtualnego świata gry.
2. **evaluate(self, pixel\_values, coords)**
   * **Opis**: Metoda ocenia, jak opłaca się dany punkt na torze na podstawie wartości pixel\_values (specjalnych zmiennych oznaczających przejechaną odległość oraz wychylenie od środka toru w bok) oraz realnych koordynatów piksela coords. Różne samochody mają różne algorytmy oceny, zależne od car\_id.
   * **Wejście**:
     + pixel\_values – wartości „zakrzywionej geometrii” (x,y), gdzie x oznacza pokonany dystans wyścigu, a y wychylenie od środka toru
     + coords – wartość piksela (x, y) reprezentująca punkt na torze.
   * **Wyjście**:
     + Zwraca ocenę danego punktu. Im wyższa wartość, tym bardziej preferowany punkt.
3. **check\_values(self, r)**
   * **Opis**: Funkcja wykonuje "skan" wokół pojazdu w określonym promieniu (r), oceniając wartości różnych punktów na torze w tym obszarze. Funkcja ta zwraca słownik z wynikami oceny.
   * **Wejście**:
     + r – promień skanu (domyślnie 40 lub 150).
   * **Wyjście**:
     + Zwraca słownik, w którym klucze to ocenione wartości, a wartości to krotki (kąt, współrzędne).
4. **enemy\_move(self)**
   * **Opis**: Metoda ocenia, w którą stronę powinna poruszać się sztuczna inteligencja. Wykorzystuje wcześniej ocenione wartości, aby podjąć decyzję o ruchu. W zależności od car\_id, AI wykonuje różne skany oraz reakcje.
   * **Wejście**: Brak.
   * **Wyjście**: Brak. Ta metoda aktualizuje atrybuty obiektu (stear, joystick\_y) w zależności od obliczeń.
5. **bonus\_conditions(self)**
   * **Opis**: Funkcja sprawdza specjalne warunki jazdy dla pojazdu AI. Na przykład, w zależności od car\_id, pojazd może mieć różne limity prędkości i przyspieszenie. Funkcja ta aktualizuje prędkość i przyspieszenie pojazdu na podstawie bieżących warunków.
   * **Wejście**: Brak.
   * **Wyjście**: Brak. Aktualizuje stan pojazdu (np. prędkość, przyspieszenie, kierunek).

**Logika działania AI**

1. **Ruchy AI**:
   * AI używa algorytmu oceny punktów toru na podstawie różnych funkcji i skanów. Każdy pojazd ma swoją strategię jazdy, zależną od car\_id.
   * evaluate ocenia, które punkty na torze są bardziej "opłacalne" do przejechania, a check\_values analizuje te punkty w danym promieniu od pojazdu.
   * AI wykonuje skan o promieniu 40-150 jednostek (zależnie od typu auta), po czym wybiera najlepszy kąt i kierunek.
   * Jeśli AI wykryje gracza w pobliżu, może zmienić swoje zachowanie, na przykład przyspieszając, aby go wyprzedzić.
2. **Dostosowanie strategii**:
   * W zależności od car\_id, AI może przyjmować różne strategie:
     + **Car 1**: Mistrz prostej. Na prostych odcinkach toru jeździ bardzo szybko, lecz na zakrętach bardzo hamuje.
     + **Car 2**: Dąży do jazdy centralnie w środku toru ze stałą prędkością.
     + **Car 3**: Dogania gracza i jak jest obok, to wbija gracza w barierę.
     + **Car 4**: Jeździ przed graczem nie dając się wyprzedzić, wymuszając kolizje gracza o swój tył.
3. **Unikalne wartości**:
   * -213700 to specjalna wartość, która wskazuje, że AI powinna jechać centralnie w stronę gracza w celu wbicia go w barierę.
4. **Bonusy i warunki specjalne**:
   * AI reaguje na różne zmienne, takie jak prędkość, kierunek skrętu, a także specjalne bonusy przyspieszenia, które są aktywowane w zależności od sytuacji na torze.

**Podsumowanie**

Klasa ai steruje samochodami w grze przy użyciu zaawansowanego algorytmu, który ocenia tor i podejmuje decyzje o ruchach w czasie rzeczywistym. AI uwzględnia różne czynniki, takie jak obecność gracza, aktualna prędkość, kierunek jazdy, a także specjalne bonusy, które pozwalają sztucznej inteligencji na lepszą adaptację do warunków wyścigu. Każdy pojazd AI posiada swoją unikalną strategię jazdy, która jest zależna od przypisanego car\_id.