**Dokumentacja klasy Car w pliku Car.py**

**Klasa Car**

Klasa Car reprezentuje samochód w grze. Odpowiada za sterowanie pojazdem, jego fizykę ruchu, detekcję kolizji, rysowanie na ekranie, a także różne interakcje z otoczeniem, takie jak rozlewanie oleju, ślizganie się po nim i kontrolowanie liczby okrążeń na torze.

**Inicjalizator \_\_init\_\_(self, window, track, coords, image, centre\_point, angle, car\_id)**

Inicjalizuje instancję pojazdu.

**Parametry:**

* window: Obiekt okna gry, na którym rysowane są elementy gry.
* track: Lista elementów toru, które są wykorzystywane do detekcji kolizji i sprawdzania pozycji pojazdu.
* coords: Początkowe współrzędne pojazdu na ekranie.
* image: Obrazek reprezentujący pojazd.
* centre\_point: Punkt środkowy samochodu (używany do kolizji).
* angle: Początkowy kąt pojazdu (w stopniach).
* car\_id: Unikalny identyfikator pojazdu (0 dla gracza, inne liczby dla sztucznej inteligencji).

**Atrybuty:**

* car\_id: Identyfikator samochodu (0 dla gracza, inne wartości dla AI).
* track: Tor, na którym znajduje się pojazd.
* stear: Kąt skrętu pojazdu.
* joystick\_x: Ruch w poziomie (lewo/prawo).
* joystick\_y: Ruch w pionie (przód/tył).
* last\_pos: Ostatnia zapisana pozycja pojazdu (x, y, kąt).
* is\_braking: Flaga informująca, czy pojazd hamuje.
* pressed\_keys: Zbiór naciśniętych klawiszy.
* acceleration: Przyspieszenie pojazdu.
* Xacceleration: Przyspieszenie w kierunku osi X.
* Yacceleration: Przyspieszenie w kierunku osi Y.
* velocity: Prędkość pojazdu.
* Xvelocity: Prędkość w kierunku osi X.
* Yvelocity: Prędkość w kierunku osi Y.
* drift: Wartość driftu pojazdu (0-100).
* tile\_id: Identyfikator pola toru, na którym znajduje się pojazd.
* laps: Liczba ukończonych okrążeń.
* laps\_times: Czas okrążeń.
* best\_lap\_time: Najlepszy czas okrążenia.
* oil\_cooldown: Czas do ponownego rozlania oleju.
* oil\_coords: Współrzędne rozlania oleju.
* oil\_radius: Promień plamy oleju.
* oiled\_time: Czas, przez który pojazd jest śliski z powodu oleju.
* hitbox: Obiekt hitboxu pojazdu do detekcji kolizji.

**Metody:**

**draw(self, x, y)**

Rysuje samochód na ekranie. Dodatkowo, jeżeli pojazd rozlał olej, rysuje plamę oleju.

**Parametry:**

* x, y: Współrzędne, na których pojazd ma zostać narysowany.

**get\_pressed\_keys(self, pressed\_keys)**

Aktualizuje informacje o naciśniętych klawiszach i odpowiednio ustawia zmienne dotyczące ruchu pojazdu.

**Parametry:**

* pressed\_keys: Zbiór aktualnie naciśniętych klawiszy.

**spill\_oil(self)**

Rozlewa olej na torze i ustawia czas aktywności plamy oleju. Można to wywołać tylko przez gracza.

**check\_oil\_collision(self, player)**

Sprawdza, czy pojazd wpadł w plamę oleju, którą rozlał inny pojazd (np. gracz).

**Parametry:**

* player: Pojazd, który może rozlać olej.

**Zwraca:**

* True, jeśli pojazd dotknął plamę oleju, False w przeciwnym razie.

**isColliding(self, oSprite)**

Sprawdza, czy samochód koliduje z innym obiektem (np. innym samochodem lub torowiskiem).

**Parametry:**

* oSprite: Obiekt, z którym sprawdzamy kolizję.

**Zwraca:**

* True jeśli doszło do kolizji, False w przeciwnym razie.

**turn(self)**

Aktualizuje kąt skrętu pojazdu, bazując na wejściu z joysticka i bieżącym stanie hamulca.

**move(self)**

Aktualizuje ruch pojazdu, przyspieszenia, prędkości oraz obrót. Wprowadza fizykę ruchu i driftu, w tym również interakcje z olejem i innymi debuffami.

**back(self)**

Cofnięcie pojazdu do poprzedniej zapisanej pozycji, jeśli doszło do kolizji.

**is\_next\_lap(self, oTrack)**

Sprawdza, czy pojazd przejechał przez pole toru, które oznacza rozpoczęcie nowego okrążenia.

**Parametry:**

* oTrack: Obiekt toru, który może być aktualnym polem toru, w którym znajduje się pojazd.

**Zwraca:**

* 1 jeśli pojazd jest na nowym okrążeniu, 0 w przeciwnym razie, -1 jeśli pojazd wrócił na poprzednie okrążenie.

**set\_laps(self)**

Aktualizuje liczbę ukończonych okrążeń i czas okrążeń na podstawie pozycji pojazdu.

**find\_tile(self, coords)**

Znajduje pole toru, na którym znajduje się punkt o współrzędnych coords.

**Parametry:**

* coords: Współrzędne punktu, dla którego szukamy odpowiedniego pola toru.

**Zwraca:**

* Obiekt toru, do którego należy dany punkt, lub None, jeśli taki punkt nie należy do żadnego pola toru.

**find\_pixel\_values(self, coords)**

Znajduje wartości „zakrzywionej geometrii” toru w zależności od współrzędnych podanego punktu.

**Parametry:**

* coords: Współrzędne punktu, dla którego szukamy wartości pikseli.

**Zwraca:**

* Wartości „zakrzywionej geometrii” punktu, jeśli punkt należy do toru. Jeśli nie, zwraca None.

**Zastosowanie klasy Car:**

Klasa Car jest wykorzystywana do modelowania pojazdów w grze, zarówno dla gracza, jak i dla sztucznej inteligencji. Pozwala na pełną kontrolę nad pojazdem, w tym obsługę fizyki, kolizji, skrętu, przyspieszenia, hamulców oraz interakcji z innymi obiektami (np. rozlewaniem oleju). Dzięki tej klasie pojazdy mogą poruszać się po torze, uczestniczyć w wyścigach i wchodzić w interakcje z innymi elementami gry.