Plik Sprite.py

Klasa Sprite została stworzona na wzór składni dostępnej w „Scratchu” po to, aby umożliwić tworzenie i manipulowanie obrazkami (tzw. sprite'ami) w grze stworzonej przy pomocy biblioteki pygame. Klasa ta pozwala na obracanie, przesuwanie, skalowanie oraz sprawdzanie kolizji z innymi obiektami. Oto szczegółowe omówienie jej funkcji:

**Główne atrybuty:**

1. **window**: Okno gry w pygame, w którym sprite będzie rysowany.
2. **coords**: Pozycja sprite'a na ekranie w postaci współrzędnych (x, y).
3. **angle**: Kąt obrotu sprite'a.
4. **base\_img**: Obrazek, który jest używany jako źródło dla sprite'a.
5. **mask**: Maska stworzona na podstawie obrazu sprite'a, używana do detekcji kolizji.
6. **centre\_point**: Punkt centralny dla rotacji (zwykle jest to środek obrazu).
7. **scale\_factor**: Współczynnik skali, który pozwala na zmianę rozmiaru sprite'a.
8. **is\_player**: Flaga boolean, która wskazuje, czy sprite jest kontrolowany przez gracza.

**Główne metody:**

1. **change\_position(x, y)**: Przemieszcza sprite'a o wartości (x, y).
2. **set\_position(coords)**: Ustawia pozycję sprite'a na konkretne współrzędne (x, y).
3. **forward(distance)**: Przemieszcza sprite'a do przodu o określoną odległość, w kierunku jego aktualnego kąta obrotu.
4. **rotate(angle\_deg)**: Obraca sprite'a o zadany kąt w stopniach.
5. **set\_angle(angle\_deg)**: Ustawia kąt obrotu sprite'a na konkretną wartość.
6. **point\_at(coords)**: Obraca sprite'a w kierunku konkretnego punktu (x, y).
7. **draw(relative\_x, relative\_y)**: Rysuje sprite'a na ekranie, uwzględniając przesunięcie kamery (jeśli jest).
8. **isColliding(oSprite)**: Sprawdza, czy sprite koliduje z innym sprite'em, porównując ich maski (hitboxy).
9. **reset\_absolute\_values()**: Aktualizuje pozycję absolutną sprite'a, uwzględniając jego obraz oraz punkt centralny.
10. **scale\_by(factor)**: Skaluje sprite'a o dany współczynnik i aktualizuje odpowiednio wszystkie atrybuty.

**Zastosowanie w grze:**

* Sprite może być przesuwany, obracany i skalowany, co jest użyteczne do tworzenia obiektów lub postaci, które mogą wchodzić w interakcje z innymi elementami w grze.
* Detekcja kolizji pozwala na sprawdzenie, czy dwa sprite'y nachodzą na siebie, co może być przydatne np. w mechanice gry, gdy postacie lub obiekty wchodzą w interakcję.
* Funkcje obrotu i ruchu umożliwiają symulację różnych działań, takich jak obracanie postaci, pojazdów lub innych obiektów w grze.

**Uwagi do kodu:**

1. **Detekcja kolizji** jest wykonywana przy użyciu pygame.mask, co pozwala na dokładną detekcję kolizji na poziomie pikseli.
2. Funkcje **rotacji i skalowania** aktualizują zarówno obraz sprite'a, jak i wewnętrzne obliczenia związane z jego punktem centralnym i wektorem, zapewniając dokładne pozycjonowanie.

**Podsumowanie:**

Klasa Sprite jest podstawowym narzędziem do tworzenia obiektów w grze 2D w pygame. Dzięki niej możemy łatwo manipulować postaciami, obiektami, rysować je na ekranie, obracać, przesuwać oraz wykrywać kolizje. To potężne narzędzie w tworzeniu gier 2D.