

Dokumentation: Asteroid Avoidance Game

Idee des Spiels

Das Asteroid Avoidance Game ist ein einfaches Arcade-Spiel, das den Spieler vor die Herausforderung stellt, Asteroiden auszuweichen, während er versucht, so viele Punkte wie möglich zu sammeln. Die Idee hinter dem Spiel ist es, die Reaktionsfähigkeit und Geschicklichkeit des Spielers zu testen, während er versucht, so lange wie möglich zu überleben und seine Punktzahl zu maximieren.

Grobe Struktur des Spiels

Das Spiel besteht aus verschiedenen Klassen, die verschiedene Aspekte des Spiels repräsentieren, darunter das Hintergrundbild, das Spieler-Raumschiff, die feindlichen Asteroiden, Projektile und das Schwierigkeitsmenü. Die grundlegende Struktur des Spiels umfasst die Verwaltung der Spielobjekte, die Handhabung von Ereignissen, die Aktualisierung des Spielzustands und die Darstellung von Grafiken auf dem Bildschirm.

Klassenstruktur

1. Polygon

Repräsentiert ein Polygon-Objekt im Spiel.

Enthält Methoden zur Positionsbestimmung, Bewegung und Kollisionserkennung.

2. Enemy

Repräsentiert einen feindlichen Asteroiden im Spiel.

Bewegt sich vertikal auf dem Bildschirm und fällt von oben herab.

Wird zerstört, wenn es von einem Spielerprojektile getroffen wird.

3. Player

Repräsentiert das Spieler-Raumschiff im Spiel.

Kann sich horizontal bewegen und Projektile abfeuern, um Asteroiden zu zerstören.

4. Background

Verwaltet das Hintergrundbild des Spiels.

Bewegt sich, um den Eindruck von Bewegung zu vermitteln.

5. Bullet

Repräsentiert ein Projektil, das vom Spieler abgefeuert wird.

Bewegt sich nach oben und kann Asteroiden treffen und zerstören.

6. DifficultyMenu

Verwaltet das Schwierigkeitsmenü des Spiels.

Ermöglicht dem Spieler die Auswahl des Schwierigkeitsgrades vor Spielbeginn.

7. AsteroidAvoidanceGame

Die Hauptklasse, die das Spiel steuert und den Spielbildschirm darstellt.

Enthält die Hauptspiellogik, Ereignishandhabung und Spielinitialisierung.

APIs der Datenstrukturen

1. Polygon

`__init__(self, shape, position, rotation)`: Initialisiert das Polygon-Objekt mit einer Form, Position und Rotation.

`move(self, x=0, y=0)`: Bewegt das Polygon um die angegebenen x- und y-Koordinaten.

`rotate(self, degrees=0)`: Rotiert das Polygon um den angegebenen Winkel.

`contains(self, point)`: Überprüft, ob ein Punkt innerhalb des Polygons liegt.

2. Enemy

`__init__(self, polygon, speed, screen_width, screen_height)`: Initialisiert den feindlichen Asteroiden mit einer Polygonform, Geschwindigkeit und Bildschirmabmessungen.

`move(self)`: Bewegt den feindlichen Asteroiden nach unten auf dem Bildschirm.

`draw(self, surface)`: Zeichnet den feindlichen Asteroiden auf der gegebenen Oberfläche.

`contains(self, point)`: Überprüft, ob ein Punkt innerhalb des feindlichen Asteroiden liegt.

3. Player

`__init__(self, speed, bullet_speed, screen_width, screen_height)`: Initialisiert das Spieler-Raumschiff mit Geschwindigkeit, Projektilgeschwindigkeit und Bildschirmabmessungen.

`update_position(self)`: Aktualisiert die Position des Spieler-Raumschiffs basierend auf Benutzereingaben.

`draw(self, surface)`: Zeichnet das Spieler-Raumschiff auf der gegebenen Oberfläche.

shoot(self): Feuert ein Projektil ab.

4. Background

__init__(self): Initialisiert das Hintergrundbild des Spiels.

update(self): Aktualisiert die Position des Hintergrundbildes.

render(self, surface): Rendert das Hintergrundbild auf der gegebenen Oberfläche.

5. Bullet

__init__(self, position, velocity): Initialisiert ein Projektil mit Position und Geschwindigkeit.

move(self): Bewegt das Projektil nach oben auf dem Bildschirm.

draw(self, surface): Zeichnet das Projektil auf der gegebenen Oberfläche.

contains(self, point): Überprüft, ob ein Punkt innerhalb des Projektils liegt.

6. DifficultyMenu

__init__(self, options): Initialisiert das Schwierigkeitsmenü mit einer Liste von Optionen.

show(self): Zeigt das Schwierigkeitsmenü an und wartet auf Benutzereingaben.

7. AsteroidAvoidanceGame

__init__(self): Initialisiert das Spielobjekt.

handle_events(self): Behandelt die Benutzereingaben und Ereignisse.

update(self): Aktualisiert den Spielzustand basierend auf den Ereignissen.

render(self): Rendert die Spielobjekte auf dem Bildschirm.

Benutzung des Programms

Starten Sie das Spiel, indem Sie die **main.py-Datei** ausführen.

Wählen Sie den gewünschten Schwierigkeitsgrad aus dem Menü aus.

Steuern Sie das Spieler-Raumschiff mit den Pfeiltasten und feuern Sie Projektile mit der Leertaste ab.

Versuchen Sie, so lange wie möglich zu überleben und so viele Asteroiden wie möglich zu zerstören, um Punkte zu sammeln.

Wenn Sie es Eine Minute überleben könnten, dann wird die Geschwindigkeit der Asteroiden erhöht und das wiederholt sich jede Minute.

Das Spiel endet, wenn ein Asteroid das Spieler-Raumschiff trifft.

Vergleichen Sie Ihre Punktzahl mit Ihrem persönlichen Bestwert und versuchen Sie, ihn zu übertreffen.

Fazit

Das Asteroid Avoidance Game ist ein unterhaltsames und herausforderndes Spiel, das Geschwindigkeit, Reaktion und Geschicklichkeit erfordert. Mit seiner einfachen Bedienung, aber dennoch anspruchsvollen Spielmechaniken bietet es Spielern jeden Alters eine spannende Spielerfahrung.

Viel Spaß beim Spielen!