Copyleaks

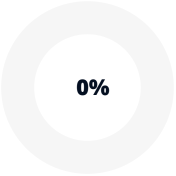
Plagiarism report

clouds.py

Scan details

Scan time:	Total Pages:	Total Words:
April 13th, 2024 at 9:29 UTC	4	829

Plagiarism Detection



Types of plagiarism		Words
● Identical	0%	0
● Minor Changes	0%	0
● Paraphrased	0%	0
○ Omitted Words	0%	0

AI Content Detection

Text coverage

● AI text

○ Human text

Plagiarism Results: No results found!

```python
"""
This module is designed to handle cloud entities within the game environment. It provides functionality
to create, update, and render clouds, adding to the visual aesthetics of the game's backgrounds or skies.
Cloud entities are created with varying speeds and depths to simulate realistic movement and parallax effec
"""

import random

class Cloud:
    """
    Represents an individual cloud in the game with its own position, image, speed, and depth.

    Attributes:
        pos (list of float): The x and y position of the cloud in the game world.
        img (pygame.Surface): The image representing the cloud.
        speed (float): The speed at which the cloud moves, simulating wind.
        depth (float): The depth of the cloud, used to calculate parallax effects and simulate distance.
    """

    def __init__(self, pos, img, speed, depth):
        """
        Initializes a new Cloud instance with the specified attributes.

        Parameters:
            pos (list of float): The initial position of the cloud.
            img (pygame.Surface): The image representing the cloud.
            speed (float): The cloud's movement speed.
            depth (float): The cloud's depth relative to other entities in the game world.
        """
        self.pos = list(pos)
        self.img = img
        self.speed = speed
        self.depth = depth

    def update(self):
        """
        Updates the cloud's position based on its speed.

        This method should be called once per frame to move the cloud across the game environment.
        """
        self.pos[0] += self.speed

    def render(self, surf, offset=(0, 0)):
        """
        Renders the cloud to a specified surface, adjusting its position based on the provided offset
        and its depth to create a parallax effect.

        Parameters:
            surf (pygame.Surface): The Pygame surface to draw the cloud on.
            offset (tuple of float): An (x, y) offset to apply to the cloud's position, used for camera mov
                                     or scrolling environments.
        """
        render_pos = (self.pos[0] - offset[0] * self.depth, self.pos[1] - offset[1] * self.depth)
        surf.blit(self.img, (render_pos[0] % (surf.get_width() + self.img.get_width()) - self.img.get_width
                             render_pos[1] % (surf.get_height() + self.img.get_height()) - self.img.get_hei

class Clouds:
    """
    Manages a collection of Cloud instances, providing methods to update and render all clouds as a group.

    Attributes:
        clouds (list of Cloud): A list of Cloud objects representing individual clouds in the game environm
    """

    def __init__(self, cloud_pngs, count=16):
        """
        Initializes a new Clouds instance, creating a specified number of Cloud objects with random attribu

        Parameters:
            cloud_pngs (list of pygame.Surface): A list of images that can be used to represent the clouds.
            count (int, optional): The number of cloud instances to create. Defaults to 16.
        """
        self.clouds = [Cloud((random.random() * 99999, random.random() * 99999),
                             random.choice(cloud_pngs),
                             random.random() * 0.05 + 0.05,
                             random.random() * 0.6 + 0.2) for _ in range(count)]
        self.clouds.sort(key=lambda cloud: cloud.depth)
    def update(self):
        """
        Updates all cloud instances, moving them based on their individual speeds.

        This method should be called once per frame to animate the clouds across the game environment.
        """
        for cloud in self.clouds:
            cloud.update()

    def render(self, surf, offset=(0, 0)):
        """
        Renders all cloud instances to a specified surface, applying a global offset to each cloud
        and adjusting for parallax effects based on cloud depth.

        Parameters:
            surf (pygame.Surface): The Pygame surface to draw the clouds on.
            offset (tuple of float): A global (x, y) offset to apply to all clouds, used for camera movemen
                                     or scrolling environments.
        """
        for cloud in self.clouds:
            cloud.render(surf, offset=offset)
```