The project implements a simplified Pong game environment using the OpenAI Gym framework (mini_pong.py) and generates a dataset of pixel representations (sprites.py). The environment, represented by the MiniPongEnv class, simulates a miniature Pong game where a paddle moves left or right and the ball bounces within a confined space. The code imports necessary libraries like gymnasium, numpy, and matplotlib to set up and visualize the environment. The environment is initialized with attributes such as level, size, and normalization parameters that control its dynamics and scale. The action space includes three discrete actions: no movement, moving left, or moving right, while the observation space varies based on the level, influencing the state representation format.

The reset method initializes the game state, including ball and paddle positions.
The step method processes actions to update the game state, checks for collisions, and determines rewards or terminal conditions. The environment's observation is managed through the _get_observation method, which returns the state as either a pixel grid or a numerical array depending on the chosen level. The _to_pixels method generates pixel representations of the current state for visualization, while the render method uses matplotlib to display the game state.

In sprites.py, the environment is instantiated with level=0 to obtain pixel-based observations for dataset generation. The script iterates through all possible (x, y, p) state combinations (representing ball and paddle positions), manually sets these states in the environment, and captures the corresponding pixel grids. These are stored as pixel data (allpix) and labels (alllabels). The data is then randomized and saved as CSV files for further use.

The script also visualizes and saves a few example pixel images from the dataset to illustrate the environment states. Finally, the dataset is split into training and testing sets (80% and 20%, respectively), with each saved as separate CSV files. This setup enables further machine learning tasks, such as predicting paddle and ball positions based on pixel data, providing a simplified model for reinforcement learning applications. The code demonstrates how an environment can be built, visualized, and transformed into structured data for training purposes.

```
import gymnasium as gym
from gymnasium import spaces
import numpy as np
import random
import matplotlib.pyplot as plt
```

- gymnasium: To create the environment framework.
- spaces: Defines the action and observation space of the environment.
- numpy: For numerical operations.
- random: For randomizing initial game states.
- matplotlib.pyplot: To visualize the environment’s state.