

main.py

This file processes game statistics by extracting usernames and heroes from a game screenshot, retrieving player statistics, and placing the stats on the image.

Dependencies:

- utils.get_heroes
- utils.ocr_extractor
- utils.player_stats
- utils.put_stats_on_image
- utils.screenshot

Usage:

1. Ensure that the necessary dependencies are installed.
2. Execute the file.

utils/get_heroes.py

This file downloads hero avatars from the Omeda City dashboard and converts them to PNG format.

Dependencies:

- - os
- - requests
- - PIL (Python Imaging Library)
- - bs4 (BeautifulSoup)

Usage:

1. Ensure that the required dependencies are installed.
2. Execute the file.

utils/ocr_extractor.py

This file extracts usernames and corresponding hero names from a game screenshot using image processing and OCR techniques.

Dependencies:

- - os
- - cv2 (OpenCV)
- - easyocr
- - PIL (Python Imaging Library)

Usage:

1. Ensure that the required dependencies are installed.
2. Provide the path to the game screenshot image.
3. Execute the file.

utils/player_stats.py

This module extracts usernames and corresponding hero names from a game screenshot using image processing and OCR techniques.

Dependencies:

- os: The operating system interface for handling file paths and directories.
- cv2 (OpenCV): The computer vision library for image processing and analysis.
- easyocr: The OCR library for text extraction from images.
- PIL (Python Imaging Library): The library for image processing and manipulation.

Usage:

1. Ensure that the required dependencies are installed. You can install them using package managers like pip.
2. Provide the path to the game screenshot image. Update the `image_path` variable with the correct path.
3. Execute the file to extract usernames and hero names from the screenshot.

utils/put_stats_on_image.py

This module extracts usernames and corresponding hero names from a game screenshot using image processing and OCR techniques.

Dependencies:

- os: The operating system interface for handling file paths and directories.
- cv2 (OpenCV): The computer vision library for image processing and analysis.
- easyocr: The OCR library for text extraction from images.
- PIL (Python Imaging Library): The library for image processing and manipulation.

Usage:

1. Ensure that the required dependencies are installed. You can install them using package managers like pip.
2. Provide the path to the game screenshot image. Update the `image_path` variable with the correct path.
3. Execute the file to extract usernames and hero names from the screenshot.

utils/screenshot.py

This module provides functionality to capture a screenshot of a specific application window using automation techniques.

Dependencies:

- time: The module for time-related functions.
- pyautogui: The library for GUI automation and screen capture.

Usage:

1. Ensure that the required dependencies are installed. You can install them using package managers like pip.
2. Adjust the application window title in the `getWindowsWithTitle` function to match the desired application.
3. Set an appropriate delay in seconds using the `time.sleep` function if needed to wait for the application to be ready before capturing the screenshot.
4. Execute the file to capture the game screenshot.

structures/stat_classes.py

This module provides abstract base classes and concrete implementations for player statistics in a game.

Dependencies:

- abc (Abstract Base Classes): The module for defining abstract base classes.

Usage:

1. Import the required classes from this module.
2. Create instances of HeroPlayerStats and OverallPlayerStats with appropriate values for the player statistics.
3. Access the attributes and methods of the instances as needed.