```python
import sys

from pipelines.registry_pipelines import single_registry_pipeline, all_registries_pipeline
from utils.fms import init_registry_dirs, get_registry_codes
from utils.helpers import handle_cli_verbosity


def update_single_registry(argv):
    """
    Runs steps to extract aircraft registry data from a specific remote source, transform the extracted data into a
    standardized format, and load the transformed data by integrating it into tabular stores used to seed or update
    components of a data warehouse

    :param (list) argv: command line arguments - pass registry-code for the target remote source to update;
        pass 'verbose' to print progress messages
    :return: effect - creates [DATA_DIR]/world[REGISTRY.name] file;
        effect - creates [DATA_DIR]/aircrafts[TABLE.name] file;
        effect - creates [DATA_DIR]/registrants[TABLE.name] file;
        effect - creates [DATA_DIR]/registrations[TABLE.name] file;
        effect - creates [DATA_DIR]/[REGISTRY_INPUT_DIR]/[argv[1]]/[file or directory] asset(s);
        effect - creates [DATA_DIR]/[REGISTRY_OUTPUT_DIR]/[argv[1]]/[argv[1]][REGISTRY.name] file
    """

    # initialize registry data directories
    init_registry_dirs()

    if len(argv) <= 1:
        # handle when registry-code of remote source to process is missing
        sys.exit("Error: must pass registry-code of remote source to process")

    # assign registry-code of remote source to process
    registry_code = argv[1].lower()

    if registry_code.upper() not in get_registry_codes():
        # handle when registry-code of remote source to process is invalid
        sys.exit("Error: invalid registry-code of remote source to process")

    # assign whether progress messages should be printed
    verbose = handle_cli_verbosity(argv, 2)

    # extract, transform, and load remote registry data source
    single_registry_pipeline(registry_code, verbose=verbose)
    # compile world registry data
    all_registries_pipeline(verbose, read_only=True)


if __name__ == "__main__":
    update_single_registry(sys.argv)
```