* Setup Euclidean Distance Project as Solution
  + Setup a static lib for core logic
    - Setup strategy pattern for switching between different distance transform algorithms
    - Implemented a strategy for OpenCV Euclidean Distance Transform
  + Setup a Unit Test Project for Testing
  + Setup a App for Testing
    - Added Basic Testcases for small images
* Learnings
  + C++ Solution in base directory (don’t check box when creating)
  + Change Project Settings for all Configurations and Output Systems always
  + Empty Project can be changed to static lib
  + Referencing in a solution should link to a static lib automatically.
  + If it does not – “Use Library Dependency Inputs” to Yes
  + .lib File might not be created for some reason
  + Add additional include paths for easy use of #include without relative paths
  + $(SolutionDir) is a macro for the solution directory in propertys of vs
  + Vcpkg works with vs and you can #include headers – linking will be setup automatically
  + When creating a class within the header call it .hpp instead of .h (convention)
* TODO
  + Setup more Tests for both the Unit Test Framework and visual tests for the app.
  + Add some more strategys to the pattern and tests for those
  + Add Performance Testing
    - Research best method
  + Try Setting up the Project with cmake instead for learning purposes
  + Create map class
    - Think about Border / padding for edt with roads
      * Find out worse case requirements
    - Think about general best implementation including inheritance and so on
    - Implement