* Setup Euclidean Distance Project as CMake Project
  + - Added vcpkg for dependencys
    - Added dependency to OpenCV
    - Linked against OpenCV
    - Added static lib for core
    - Linked Against static lib
    - Linked lib against OpenCV
* Learnings
  + Vcpkg can be added via root cmakelists or cmakepresets (preferred)
    - Add to Presets to cache variables
      * "CMAKE\_TOOLCHAIN\_FILE": "$env{VCPKG\_ROOT}/scripts/buildsystems/vcpkg.cmake"
    - Or add to lists
      * set(CMAKE\_TOOLCHAIN\_FILE "$ENV{VCPKG\_ROOT}/scripts/buildsystems/vcpkg.cmake" CACHE STRING "")
  + Use find\_package(OpenCV REQUIRED) for packages
  + Use include\_directories(${OpenCV\_INCLUDE\_DIRS}) for header access
  + Use target\_link\_libraries(euclidean\_distance\_transform\_app PUBLIC ${OpenCV\_LIBS}) for Linking (public will also include if you include this project, but options are also PRIVATE or INTERFACE)
  + Split up Librarys into src and include directorys
  + Add interface headers to include directory
  + If you link against the library make sure the include directorys are set aswell
  + Use target\_include\_directories for the lib to find its own headers in cmakelists
* 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
  + 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
  + Implement CMake Project