

PA5 – SIMD

Due Date

- See Piazza for any changes to due date and time
- Submit program to perform in your student directory
 - Sub directory called:
 - /PA5/...
 - Fill out your **PA5 Submission Report.pdf**
 - Place it in the same directory as your solution
 - Enter the final **Changelist** number of your submission
 - Enter the number of problems passed
 - Write up a quick discussion in the report
 - What you learned from this PA

Goals

- Learn
 - SIMD, Intrinsics
 - Show off, you can program vector code!
- MMX - SSE4.1 allowed
 - No AVX or more advance SIMD intrinsics allowed

Assignments

1. Finish implementing class **Vect4D**

- Required methods are prototyped out
 - Make sure you add const and missing methods such as the big4
 - Make everything constant when possible
- Vect4D - just like a 3D vector, except you add, sub, mult, div ... all 4 elements {x,y,z,w}
 - Multiply element by element { a.x*b.x -> c.x, a.y*b.y->c.y, ...}
- Look at lecture for many of the math operations
 - Confused? -> start a post

2. Convert a given class **Vect4D** to **Vect4D_SIMD**

- Convert all methods to use intrinsics SIMD instructions
- Test the code with data provided that
 1. Modify and altered class (Vect4D).
 2. Modify and compile the new SIMD class (Vect4D_SIMD).
 3. Please verify that the new class creates the same output.
- Run the tests in Debug and Release

3. Convert a given class *Matrix* to *Matrix_SIMD*

- Convert all methods to use intrinsics SIMD instructions
- Test the code with data provided that
 1. Altered class (*Matrix*).
 2. Modify and compile the new SIMD class (*Matrix_SIMD*).
 3. Please verify that the new class creates the same output.
- Run the test in Debug and Release

4. Convert method (*Vect4D * Matrix*) to (*Vect4D_SIMD * Matrix_SIMD*)

- Convert all methods to use intrinsics SIMD instructions
- Test the code with data provided that
 1. Altered method (*Vect4D * Matrix*).
 2. Modify and compile the new SIMD class (*Vect4D_SIMD * Matrix_SIMD*).
 3. Please verify that the new class creates the same output.
- Run the test in Debug and Release

5. Convert method (*Matrix * Vect4D*) to (*Matrix_SIMD * Vect4D_SIMD*)

- Convert all methods to use intrinsics SIMD instructions
- Test the code with data provided that
 1. Altered method (*Matrix * Vect4D*).
 2. Modify and compile the new SIMD class (*Matrix_SIMD * Vect4D_SIMD*).
 3. Please verify that the new class creates the same output.
- Run the test in Debug and Release

6. Convert static method *LERP()* to *LERP_SIMD()*

- Convert all methods to use intrinsics SIMD instructions
- Test the code with data provided that
 1. Altered method (*LERP*).
 2. Modify and compile the new SIMD class (*LERP_SIMD*).
 3. Please verify that the new class creates the same output.
- Run the test in Debug and Release

7. Optimized *Row and Col Major* programs.

- Both the Row and Col should have roughly the same time in Release.
 - This might require a proxy before you do the SIMD conversion.
 - Feel free to create a special proxy for Row Major and another for Col Major before you write the respective SIMD code.
 1. That should help greatly
 2. Share your times on Piazza

General:

- Write all programs in cross-platform C or C++.
 - Optimize for execution speed and robustness.
- Create a programming file for each problem, for example
 - Student directory
 - /PA5/...
 - Make sure that each problem can be compiled and run through the checked in solution
- Write all programs in cross-platform C or C++.
 - Optimize for execution speed and robustness.
- Do all your work by yourself
 - Feel free to talk with others about ideas on Piazza
 - You are 100% responsible for all code
 - See syllabus about collaboration rules
- Fill out the submission Report
 - No report, no grade
- Submit your work as you go to perform several times (at least 5)
 - As soon as you get something working, submit to perform
 - Have reasonable check-in comments
 - Seriously, I'm checking
- Make sure that your program compiles and runs
 - Warning level ALL
 - NO Warnings or ERRORS
 - Your code should be squeaky clean.
- NO STL allowed {Vector, Lists, Sets, etc...}
 - No automatic containers or arrays
 - You need to do this the old fashion way - **YOU EARNED IT**
- No modern C++
 - Lambdas, Autos, templates, etc...

Validation

Simple checklist to make sure that everything is submitted correctly

- Did you do all SIMD problems?
- Do they compile and run without any errors?
- Warning level All free?
- Did you fill out the submission report?
- Submitted it into /PA5 directory?
- Can you delete your local drive, regrab the /PA5 directory?
 - Is all the code there?
 - Does it compile?
- Did you check in your text files?

Hints

Most assignments will have hints in a section like this.

- Do many little check-ins
 - Iteration is easy and it helps.
 - Perforce is good at it.
- Look at the lecture notes!
 - A lot of good ideas in there.
 - The code in the examples work.
- It's a puzzle
 - Keep trying to work at piecing the instructions together
 - Hunt the manual for ideas...
 - <http://msdn.microsoft.com/en-us/library/y0dh78ez%28v=vs.100%29.aspx>
 - Amazing manual
 - https://software.intel.com/sites/landingpage/IntrinsicsGuide/#expand=2775,4511,4509,757,244,3536,3882,4959,5400,5399,3026,88,2114,2775&text=hadd&techs=MMX,SSE,SSE2,SSE3,SSSE3,SSE4_1
- Use the FORUMs
 - This is much harder than the last assignment.
 - See me during office hours.
 - Read, explore, ask questions in class