Optimized C++
Gam577/SE585

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

PA2 – Math_2.0

Student Information

Integrity Policy: All university integrity and class syllabus policies have been followed. I have neither given, nor received, nor have I tolerated others' use of unauthorized aid.

I understand and followed these policies:　　　　Yes　　　　No

　Name:

　Date:

Submission Details

　　Final ***Changelist*** number:

　　　　Verified build:　　　Yes　　　No

　　Number Tests Passed:

　　Required Configurations:

　　Discussion (What did you learn):

Optimized C++
Gam577/SE585

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

Verify Builds

- Follow the Piazza procedure on submission
    - Verify your submission compiles and works at the changelist number.
- Verify that only MINIMUM files are submitted
    - No – Generated files
        - *.pdb, *.suo, *.sdf, *.user, *.obj, *.exe, *.log, *.pdb, *.db, *.user
        - Anything that is generated by the compiler should not be included
    - No – Generated directories
        - /Debug, /Release, /Log, /ipch, /.vs
- Typical files project files that are required
    - *.sln, *.cpp, *.h
    - *.vcxproj, *.vcxproj.filters, CleanMe.bat

Standard Rules

Submit multiple times to Perforce
- Submit your work as you go to perforce several times (at least 5)
    - As soon as you get something working, submit to perforce
    - Have reasonable check-in comments
        - Points will be deducted if minimum is not reached

Write all programs in cross-platform C++
- Optimize for execution speed and robustness
- Working code doesn't mean full credit

Submission Report
- Fill out the submission Report
    - No report, no grade

Code and project needs to compile and run
- Make sure that your program compiles and runs
    - Warning level ALL …
    - NO Warnings or ERRORS
        - Your code should be squeaky clean.
    - Code needs to work "as-is".
        - No modifications to files or deleting files necessary to compile or run.
    - All your code must compile from perforce with no modifications.
        - Otherwise it's a 0, no exceptions

Project needs to run to completion
- If it crashes for any reason…
    - It will not be graded and you get a 0

No Containers
- NO STL allowed {Vector, Lists, Sets, etc...}
  - No automatic containers or arrays
  - You need to do this the old fashion way - *YOU EARNED IT*

Leave Project Settings
- Do NOT change the project or warning level
  - Any changing of level or suppression of warnings is an integrity issue

Simple C++
- No modern C++
  - No Lambdas, Autos, templates, etc…
  - No Boost
- NO Streams
  - Used fopen, fread, fwrite…
- No code in MACROS
  - Code needs to be in cpp files to see and debug it easy
- *Exception:*
  - implicit problem needs templates

Leaking Memory
- If the program leaks memory
  - There is a deduction of 20% of grade
- If a class creates an object using new/malloc
  - It is responsible for its deletion
- Any *MEMORY* dynamically allocated that isn't freed up is *LEAKING*
  - Leaking is *HORRIBLE*, so you lose points

No Debug code or files disabled
- Make sure the program is returned to the original state
  - If you added debug code, please return to original state
- If you disabled file, you need to re-enable the files
  - All files must be active to get credit.
  - Better to lose points for unit tests than to disable and lose all points

No Adding files to this project
- This project will work "as-is" do not add files…
- Grading system will overwrite project settings and will ignore any student's added files and will returned program to the original state

UnitTestConfiguration file (if provided) needs to be set by user
- Grading will be on the UnitTestConfiguration settings
  - Please explicitly set which tests you want graded… no regrading if set incorrectly

Optimized C++
Gam577/SE585

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

Due Dates

- See Piazza for due date and time
- Submit program perforce in your student directory assignment supplied.
- Fill out your this ***Submission Report*** and commit to perforce
    - ***ONLY*** use Adobe Reader to fill out form, all others will be rejected.
    - Fill out the form and discussion for full credit.

Goals

- Math Rework
    - Convert 475 math library
    - Mat4, Mat3, Vec4, Vec3

Assignments

- Separating Vector into Vec3 and Vec4
    - Vec3 is aligned to 16 Bytes for alignment reasons

    ```
    // ------------------------------------------------------------
    //
    // Vec3
    //   True 3 element vector...
    //     its stored in a SIMD friendly struct for cache reasons
    //     the "w" component is ignored and not set
    //
    //   v3  = | x  y  z  - |
    //
    // ------------------------------------------------------------
    ```

    - Vec4 is aligned to 16 Bytes for alignment reasons

    ```
    // ------------------------------------------------------------
    //
    // Vec4
    //   True 4 element vector...
    //     its stored in a SIMD friendly struct for cache reasons
    //
    //   v4  = | x  y  z  w |
    //
    // ------------------------------------------------------------
    ```

Optimized C++

Gam577/SE585

use Adobe Reader to complete

(Type in fields)

Submission Report

Keenan

- Separating Matrix into Mat3 and Mat4
  - Mat3 is a 3x3 matrix aligned to 16 Bytes for alignment reasons
    - Same footprint of Mat4
    - Only using the sub3x3 section

```
/*       | m0  m1  m2   0 | */
/* m = | m4  m5  m6   0 | */
/*       | m8  m9  m10  0 | */
/*       | 0   0   0   1 | */
```

  - Mat4 is a 4x4 matrix is aligned to 16 Bytes for alignment reasons
    - Same footprint of Mat4
    - Only using the sub3x3 section

```
/*       | m0  m1  m2   m3  | */
/* m = | m4  m5  m6   m7  | */
/*       | m8  m9  m10  m11 | */Type equation here.
/*       | m12 m13 m14  m15 | */
```

- Critical watch Week's 2 lecture
  - We are also using Affine transform's
    - Scale, Rot, Trans
    - That yields a better Inverse

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & b_x \\ a_3 & a_4 & a_5 & b_y \\ a_6 & a_7 & a_8 & b_z \\ c_x & c_y & c_z & d \end{bmatrix}$$

$$M^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -cA^{-1} & 1 \end{bmatrix}$$

- Lot's of cut and paste
  - Ask questions on Piazza
  - I supplied many unit tests
  - Tricky prototypes are provided
    - 232 unit test
    - 5404 individual checks
- Hints
  - I supplied the enum I used
  - You are welcomed to use that or your own scheme
  - Main thing… improve the inverses for common Affine transformations

Optimized C++
Gam577/SE585

use Adobe Reader to complete

(Type in fields)

Submission Report
Keenan

Validation

Simple checklist to make sure that everything is submitted correctly

- Is the project compiling and running without any errors or warnings?
- Does the project run **_ALL_** the unit tests execute without crashing?
- Is the submission report filled in and submitted to perforce?
- Follow the verification process for perforce
  - Is all the code there and compiles "as-is"?
  - No extra files
- Is the project leaking memory?

Hints

Most assignments will have hints in a section like this.

- This is pretty easy assignment
- Lot's of cut and paste
  - Make sure you look up a Matrix3x3 inverse and Determinant
  - Coding this is easy
    - The tests really help