Title: Tiny Tanks

Start Date: Monday, 14 July 2014

Assessment Day: Friday, 7 November 2014

Assessable units of competency

PGDMTH601A – Apply fundamental games programming mathematical skills
ICAPRG416A - Manage a software component reuse library

General description

You are to create a 2D tank game similar to the flash game "Tiny Tanks" or the Wii game "Wii Tanks". In order to do this you will need to create an accompanying Maths Library to include into the game project.

The player will be able to control a tank's movement and turret direction independently of one another – which means the turret may be firing backwards while the tank is moving forwards.

There will be targets throughout the scene that the player needs to shoot.

Please use the following controls: wasd for tank movement, q and e for rotation of the turret. space to shoot.

The movement of the tanks must be handled by matrices AND vectors in the code.

To do this, you will need to create a maths library. This library will need a 2D vector class, a 3D vector class and a 3 by 3 matrix class.

You must ensure that the data in your program is memory-aligned and optimised. And that all data is safely type-cast.

You must also document both the library and game with internal (code) and external (PDF) documentation. The external documentation must contain the appropriate UML diagrams.

Knowledge and skills

Listed here is the knowledge and skills you'll be learning and on which you will be assessed.

- Setting up a component library
- Documenting a component library
- Determine memory footprint of binary data
- Use of vectors
- Use of matrices

Evidence specifications

This is the specific evidence you must prepare for and present on assessment day to demonstrate you have competency in the above knowledge and skills. The evidence must conform to all the specific requirements listed below.

The program should have the following attributes:

1. Maths library
2. External and internal documentation
3. A memory efficient 2D top-down tank game

Your roles and responsibilities as a candidate

- Understand and feel comfortable with the assessment process
- Know what evidence you must provide during your assessment
- Take an active part in the assessment process
- Be ready for the assessment at the nominated time

Assessment instructions for candidate

METHOD OF ASSESSMENT

Assessment will be conducted by you personally presenting evidence that demonstrates your competence in a short interview with your assessor. The evidence you must prepare and present is described above in this assessment criteria document. Assessments will be conducted on a specific day recorded above in this assessment criteria document.

ASSESSMENT CONDITIONS

You will have approximately 10 mins to present your evidence that demonstrates your competence. It is your responsibility to be prepared. If you have forgotten something or made a small mistake you may correct it, however the assessor may choose to assess other candidates who are better prepared and return to you if time permits. Upon completion of the assessment you will be issued with feedback and a record of the assessment, which you will need to acknowledge that you have accepted the result. If you are absent on the nominated assessment day (without prior agreement or a sufficient documented excuse) you will be assessed as not yet competent.

GRADING

The assessment you are undertaking will be graded as either *competent* or not *yet competent*.

REASSESSMENT PROCESS

If you are assessed as being not yet competent you will receive clear, written and oral feedback on what you will need to do to achieve competence. You will have one (1) week to prepare your evidence for a reassessment. You will be given only one reassessment opportunity. If you are unsuccessful after your reassessment you will be required to attend an intervention meeting with your Head of School to discuss your progress.

REASONABLE ADJUSTMENTS

We recognise the need to make reasonable adjustments within our assessment and learning environments to meet your individual needs. If you need to speak confidentially to someone about your individual needs please contact your teacher.

Assessment rubric

This table defines exactly what is required to be successfully deemed competent.

Evidence	Definition of Competent for Tiny Tanks

1. Maths library	Competence requires you to develop a maths library separate to your game project that adheres to OOP design principles and promotes reuse and portability.
	The library must have a logical, consistent structure that another developer could easily understand and it should not have any redundant code.
	The library may be static or dynamic.
	The maths library must include the following aspects:
	• 2 dimensional and 3 dimensional vectors
	  ○ Dot product
	  ○ Normalisation
	  ○ Cross Product
	  ○ Magnitude
	  ○ Operator overloading
	• Matrix 3x3
	  ○ Operator overloading
	  ○ Translate
	  ○ Rotate
	  ○ Scale
	• Common mathematical functionality

2. External and internal documentation	You must submit a PDF file documenting the functionality and purpose of your accompanying maths library. This must include: • UML diagrams explaining the structure of your library. • Example code demonstrating the use of library components. • Info about public constants, data structures, component interfaces and limitations • Installation information including system and software requirements All code must adhere to AIE's coding standard and applied internal documentation (comment code).
3. A memory efficient 2D top-down tank game	You must submit a project to implement a 2D top-down tank game using the Maths Library you created. The tank and turret must be controlled independently and move as expected. Matrices and vectors are to be used in all sprite transforms. The compiled tank game must: • Be created with OpenGL in C++ • Have no compiler warnings or errors • Not crash or exit during execution • Contain code that aligns data in memory. • Use safe methods only when type-casting is performed. • Optimise data arrangement in memory to avoid cache miss where possible.