# Assignment Sheet

|  |  |
| --- | --- |
| **Course and instructor name** | GAME Game Dynamics 3 –  Dr. Umer Noor |
| **Assignment name** | Assignment 3 – Implementing GJK |
| **Grade value** | 20%  Rubric attached |
| **Due date** | In class |
| **Individual or group assignment** | Group |
| **Submission instructions** | Show in class and submit on Blackboard |
| **Targeting these learning outcomes from course outline** | Defend decisions made whilst creating computer programs that implement the physics of 3D motion and simple 3D collisions.  Produce computer programs that implement the physics of 3D motion and simple 3D collisions.  Assemble a collision detection algorithm to be used with simple 3D shapes. |

# Instructions

Follow on from your previous assignment:

## Part 1 - Simplex Class

* Create a Simplex class that does the following:
  + Tracks 4 points in total (3D vectors)
  + You can remove one point and put in a different one
* Using your Simplex class, create a program that basically follows the “GJK Worked Example” slides posted on Blackboard. You will need Scott’s Vector and Matrix libraries or the ones provide with your game engine.
* Check your numbers match mine and that you find the same result as my slides

## Part 2 - Implement GJK Collision detection

* Incorporate your Simplex class with the simulation of two tetrahedrons approaching each other
* Create a Boolean flag that shows “true” when a collision occurs
* Test that your GJK collision detection algorithm works as expected

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rubric | | | | | |
|  | 5% | 4% | 3% | 2-1% | 0% |
| **Simplex Class** | Coded extremely well and wonderful comments. Output matches worked example. | Coded to specifications | Missing or broken functionality | Missing lots of detail | Largely empty |
| **Collision Detection** | Test case coded well. Simulation behaves clearly as expected | Coded to specifications | Missing or broken functionality | Missing lots of detail | Largely empty |
| **Overall code quality** | Coded extremely well and wonderful comments | Coded to specifications | Missing or broken functionality | Missing lots of detail | Largely empty |
| **Presentation** | Able to explain code very well in person and handle live-coding easily | Satisfactory presentation of work done | Not able to explain all code or cannot make certain changes | Somewhat confusing or vague | Very confusing. Unsure student understands any code |
| **Process** | Pieces of the assignment have been steadily building throughout the time given. Creativity used to solve key problems. Able to reflect on what they could improve on. | Assignment completed satisfactorily | Somewhat of a rush to complete assignment | Obviously has been a big last-minute rush to complete assignment. No time for creativity or reflection | Student has not spent any time grappling with concepts, reaching out for help, or reflecting on topic |

## Rubric

## Grading standard

20/20 - Work so amazing the instructor would only see this once in a lifetime

18/20 – Exceptional work, rare

15/20 - Great work, student has full command of the topic.

12/20 - Minor errors

10/20 - Errors and perhaps a major error

7/20 - Regular and consistent major errors. Lack of understanding

5/20 - Largely empty