Graphics and Multimedia (COMP3419)

Assignment 1-b Specification

1 Key Information

- The mark of **"COMP3419 Assignment 1-b Building 3D Animation Scene"** will be given on canvas submission. Due time: before 23:59, Friday of Week 11 (2020-11-13).
- This individual assignment is worth **8%** of your final assessment.
- **Submission Deliverables**: Students are asked to create a **zip** file of all deliverables (several image files + 1 .pde file + 1 README.txt). Please be aware of the following submission **restrictions** that **(1)** this zip file should be named as "SIDxxxxxxxxx_Assignment1b.zip" where xxxxxxxxx denotes the student ID (e.g., "SID450003419_Assignment1b"). **(2)** this zip file should contain one single folder named as "SIDxxxxxxxxx_Assignment1b", which includes all other deliverables, and **(3)** the processing file should be named as "SIDxxxxxxxxx_Assignment1b.pde" as well. Failing to follow these restrictions would cause a 2 marks penalty immediately.
- Students' assignments **will only be marked if** all deliverables can be **accessed** from the Canvas System, and they can be **runnable** following instructions provided in README txt file. Once plagiarism detected by the Canvas system, the student will receive no mark immediately, as well as other related penalties from university.
- A **README** txt file (to describe the steps/instructions regarding how to get their source code running to derive the expected outputs) would help markers get familiar with the submission.
- Within the **README** file, students should also add a **one-sentence summary** to claim and specify their completed requirement IDs (see Sec. 4), and tutors would then only reproduce the corresponding results (that claimed as completed by students) from the canvas submission (zip) file, and further check the implementation correctness of these claimed requirements. An example of this one-sentence summary is as "*Completed Requirements: 2/3/4/6*" (if requirement #1, #5, and #7 are incomplete).

2 General Marking Policy

Late Submission Policy
For the late submission cases, penalties will be assigned according to the university wide late penalties for assignment Clause 7A of the Assessment Procedures.
Special Consideration and Arrangements
While you are studying, there may be circumstances or essential commitments that impact your academic performance. Our special consideration and special arrangements process is there to support you in these situations. More information on how to lodge the special consideration application, can be found from this webpage.

3 Building 3D Animation Scene (Assignment 1-b)

Before you start to program this assignment, please make sure you understand the basics of 3D graphics from lab on week 8 and finish the orbit simulator.

In this assignment, program an interactive 3D animation with the following scenarios:

- When you click mouse at (x, y) on your screen shoot a 3D ball with a random texture (randomly load a texture from your texture pool). The texture pool (several image files) should be included in canvas submission zip file.
- This ball flies away from the screen along the Z axis with a random direction according to the XY plane. It means this ball does not go straight ahead, but instead is shot away in a random direction. The gravity always effect the movement of this ball as long as it is in screen.
- While the ball moves away from the screen, it falls down to the ground due to the gravity.
- The space is constrained by 5 walls (left, right, ceiling, floor and a wall far ahead facing the screen). When the ball touches any of the wall, it bounces back and its new direction is computed considering its previous direction. The 6th wall (the wall where the camera is placed) is free to design (either bounce the ball back or make the ball disappear when the ball comes towards this wall).
- You can shoot new balls into the screen while the previous balls are travelling.
- The ball spins while it is travelling. The direction of spinning is determined by its original direction and the bounces.
- When the balls touch each other, they bounce back as well.
- The potential energy (speed and height) of the ball decays according to the travelling time as well as the number of bounces.

R

- Hint: It would be easier if you model the balls with a class.
- Nice to embed some of the Kinematic equations when you calculate the speed and direction of the balls. For equations of motion, please refer to `https://en.wikipedia.org/wiki/Equations_of_motion`

4 Detailed Requirements/Expectations

IDs	Requirement/Expectations	Marks
1	Random textures (at least 5 different textures)	1
2	Ball shot away in random direction	1
3	Bouncing works between balls and walls	1
4	The speed of the balls changes as time goes on, due to the gravity	1
5	Balls eventually fall to the ground and stop	1
6	Efficient implementation (modelling balls with a object class) - no latency or delay observed (after *a series of mouse clickings*): a. when over 10 balls flying smoothly in the air (full mark) b. when over 5 balls flying smoothly in the air (half mark) Note: the latency or delay observed (during *a series of mouse clickings*) is accepted	1
7	Collision works between balls without abnormal behaviors *(hidden testing cases)*	2