MARKING CRITERIA for **IGB283 Assignment 2**

This assignment is worth 40% of your marks for the unit, including the specified practical tasks and the main coding project. This assignment will be marked out of 100 as follows.

Overall marking criteria

1. Well-written report (including answers to theoretical questions, user instructions, screenshots for the main application program, and a link to an online video that demonstrates the execution of your system).  
   Note1: You need to answer the specified theoretical questions and provide evidence that your code working.   
   Note2: The report must be in the format of PDF or WORD or scanned hand-writing on paper. Other formats will not be accepted.
2. Working code with specified functionalities and good programming practice.

Note1: Key bindings must be consistent with the specification e.g. s for forward.   
Note2: The main application program has to work. It should be well documented (comments); easy to understand (good names for variables and functions); and easy to read (proper use of indention and white space). You should use the same interface for communication with components of the same type (e.g. put all transformation functions in one class), reduce the size of methods, and split very large classes etc.

Note3: You need to use your own IGB283 Transform functions. You can reuse some of your Assignment1 code. You can reuse any code from the workshop activities.

**High marks will be given for a basic graphics application that implements the specified features and which is easy to follow. You will not get any extra credit for extravagant solutions.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grade | (1/2/3) | 4 (Satisfactory) | 5 | 6 | 7 (Excellent) |
| **Written report**  **(overall criteria)** | **No or poorly written report** | **Written Report containing fair amount of information required** | **Well-written report clearly describing the tasks/functionalities** | **Well-written report clearly describing the functionality with sufficient details.** | **Well-written report clearly describing the functionality, details and how problems were solved** |
| **Workshops, Theoretical questions, and Reports** | | | | | |
| 1.1a Completion of specified workshop tasks  (0.455 mark per activity, totalling **5 marks**) | Completed 1 or less tasks | Completed most tasks. Some small errors may exist. | Completed most tasks correctly. | Completed all tasks correctly with sufficient details. | Completed all tasks to excellent quality. |
| 1.1b Statement of completeness  **(3 mark)** | No Statement of completeness | Simple Statement of completeness |  | Statement of completeness with good details | Statement of completeness clearly describing the tasks or functionalities with details. |
| 1.1c a short online demo video (link in the report)  **(3 marks)** | No video | Video shows main functions with audio explanation |  |  | Video shows all functions with brief and clear audio explanation. |
| 1.1d Checklist  **(2 marks)** | No checklist |  |  |  | Honest, clear and complete checklist |
| **Programming**  **(overall criteria)** | **Code not working or with very limited functions** | **Working code with most functionalities. Some coding issues may exist.** | **Working code with all required functionalities. Minor issues may exist.** | **Working code with all required functionalities with no known issues.** | **Working code with all required functionalities and good coding practice.** |
| **Task1** |  |  |  |  |  |
| Completion of specified workshop tasks (**code in folders**) | | | | | |
| Completion of specified workshop tasks  (0.455 mark per basic activity, extras optional,  totalling  **5 marks**) | Completed 1 or less tasks | Completed most tasks. Some small errors may exist. | Completed most tasks correctly. | Completed all tasks correctly with sufficient details. | Completed all tasks to excellent quality. |
| **Task 2** | (1/2/3) | 4 | 5 | 6 | 7 |
| **Nodding, Walking and Combined** | | | | | |
| 2.1  Create graphical objects and save to memory in proper data structure (vertices, vertex indices, color etc)  **(8 marks)** | Object consists of less than 3 components | Object consists of  3 or more components | Object consists of 4 or more components with proper structure |  | Object looks pretty with proper shapes and colours. |
| 2.2 **Head noding** - Each object component has proper movement information stored (translation, rotation, scaling etc)  **(8 marks)** | No or wrong movement information stored | Most components have proper movement information stored |  |  | All components have proper movement information stored e.g head nodding angle within certain range, base can move left and right and change direction. |
| 2.3 **Walking** -Modelling the relationships between object’s components using tree  **(8 marks)** | Code not working or with very limited functions | Working code with most functionalities. Some coding issues may exist. | Working code with all required functionalities. Minor issues may exist. | Working code with all required functionalities with no known issues. | Elegant code for representing the parent-child relationships among object components |
| 2.4 **Combined Articulation** -Object continually moves left or right while its head is nodding. Articulated movement achieved by traversing the tree, updating information and then rendering by again traversing the tree to draw the component objects.  **(4 marks)** | Code not working or with very limited functions | Working code with most functionalities. Some coding issues may exist. | Working code with all required functionalities. Minor issues may exist. | Working code with all required functionalities with no known issues. | Fully functional and elegant code to  continuously move the object’s components |
| 2.5 Use your own transformation functions to achieve 2.2-2.4  **(4 marks)** | Did not use your own transformation functions | Your own transformation functions mostly working, some coding issues may exist. | Working code with all required functionalities. Minor issues may exist. | Working code with all required functionalities with no known issues. | Your own transformation functions are fully functional and elegant |
| **Note1: You can assume the object base has wheels so it can move left and right and turn around.** |  |  |  |  |  |
| **Note2: head nodding should be within certain range, rather than rotating 360 degrees.** |  |  |  |  |  |
| **Note3: When objects moves horizontally to the boundary, it can change its direction and continues to move.** |  |  |  |  |  |
| Task 3 | (1/2/3) | 4 | 5 | 6 | 7 |
| **Jumping** |  |  |  |  |  |
| 3.1 Add a ground to your scene  **( 5 marks)** | No visible ground in the scene. | Simple, visible ground in the scene. Some minor issues may exist. | Visible ground has color and pattern. No known issues. | Realistic ground in the scene e.g. grass, road, table etc. | Meaningful and realistic ground that fits well with the story |
| 3.2 object continually jumps up and down like Luxo Jr. crashing the ball  **(5 marks)** | Not implemented |  |  |  | object continually moves up and down like Luxo Jr jumps on the ball. |
| 3.3 object continually leaps forward like Luxo Jr. **(5 marks)** | Not implemented | Object jumps forward (i.e., up, forward and down) continuously. May contain unforeseen mistakes. |  |  | Object jumps forward (i.e., up, forward and down) continuously like Luxo Jr. |
| **Note 1: action 3.3 and 3.4 should happen at different times.** |  |  |  |  |  |
| **Note2: Jumps should happen when the object’s base is on the floor.** |  |  |  |  |  |
| **Note 3: No user interaction needed in this task.** |  |  |  |  |  |
|  |  |  |  |  |  |
| Task4 | (1/2/3) | 4 | 5 | 6 | 7 |
| **Avatar controls** | | | | | |
|  |  |  |  |  |  |
| 4.1 steer your object using the keyboard.   key ’a’ for left, ’d’ for right   key ’w’ for jumping up, key ’s’ for jumping forward  **(4 marks)** | Code not working | All four keys function correctly most times | All four keys function correctly all the time |  | All four keys function correctly with good coding practice |
| 4.2 When no key is pressed , your object should continue its current motion (move at a constant speed in the direction of its motion, no moving backward) and never escape the world view.  **(2 marks)** | Code not working or object not moving | Object moves as specified. | Object moves at a constant speed but may escape from the view. | Object moves as specified and never escape from the view. | Fully functional as specified and best coding practice |
| 4.3 Add functionality so that when the ’z’ key is hit, keyboard control stops working and the avatar collapses to the ground.  **(2 marks)** | ’z’ key not working. | ’z’ key works and object falls somehow. |  | ’z’ key works as specified. | ’z’ key works as specified and good coding practice |
| 4.4 After landing on the ground, QUT Jr lies slumped momentarily before rising to stand again. After this process, standard avatar controls may resume.  **(2 marks)** | Not working or object broken into pieces. | Object lies on the ground but may dis-shape in places. | Object lies on ground and stands straight again. | Working as specified. Coding may have minor deficiencies. | Working as specified and good coding practice. |
|  |  |  |  |  |  |
| Task 5 | (1/2/3) | 4 | 5 | 6 | 7 |
| **Two Player Mode** | | | | | |
| 5.1  create a secondary playable avatar **(4 marks)** | No Inclusion of second avatar | Second avatar created. | Second avatar created and distinct enough from the first. |  | Second avatar is controllable independent of the first. |
| 5.2  A camera system that accommodates the two avatars  **(3 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |
| 5.3 Functional two player mode **(3 marks)** | Attempted but not working | Working with some issues | Working with minor issue. | Working with no known issue | Both avatars working properly and usage instructions documented |
|  |  |  |  |  |  |
| Task 6 | (1/2/3) | 4 | 5 | 6 | 7 |
| **Advanced Articulated** **Motion** | | | | | |
| 6.1 create a complex avatar with non-linear structure e.g. two legs/two arms/more  **(3 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |
| 6.2 complex avatar walking  **(3 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |
| 6.3 complex avatar jumping  **(3 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |
| 6.4 complex avatar control with keys  **(2 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |
| 6.5 Two player-mode involving complex avatar  **(2 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |
| 6.6 shading  **(2 marks)** | Feature not implemented | Feature implemented and functional with some issues. | Feature implemented and fully functional with minor issue. | Feature implemented and fully functional with no known issue. | Feature implemented and fully functional with good coding practice. |