COSC 3406, Fall 2025 - Checklist for the project

The maximum number of points for an item is between square brackets [ ]. The numbers that appear in the assignment feedback column follow this breakdown.

Proposal [5]

The proposal discusses all the required points.

Each point is discussed with enough detail.

Progress report [5]

The report discusses all the required points.

Each point is discussed with enough detail.

Presentation [10]

The presentation included all the required points.

Each point was discussed with enough detail.

Group members were able to answer questions about their project.

Final report [10]

The report discusses all the required points.

Each point is discussed with enough detail.

Required points:

- Briefly describe the idea of the game.
- Game manual: describe how the game is played, what are the objectives, etc.
- Describe how the technical requirements of the project were integrated into the game; mention what required features could not be added to the game.
- Add a few screenshots of the game.
- Technical notes: describe the architecture of the game, including the main classes and functions, and how these are integrated together.
- Note that, although the report is "only worth 10 points", it also contributes to understanding your game and defining the grade below.

The game [70]

Requirements: the game included all the technical requirements [40]

Both first-person and third-person camera [4]

Free flying with full ability to turn along 3 axes; the game could constrain some of the degrees of freedom to resemble flying a real helicopter [4]

Textured objects [4]

Collision detection between player helicopter and other game objects [4]

Hierarchical helicopter design with independently moving parts [5]

An armada of computer-controlled enemies; at least 3 enemy types with some basic AI behavior [5]

Weapons

- Weapons fire and destroy objects; At least 3 different weapon types [5]
- Weapons have distinct visual effects (particle filter effects) [5]

Spline trajectories for some objects [4]

Playability [15]

The game is functional: one is able to control the player, fly around, explore the game world, etc. [5]

The game enables meaningful play; there is some challenge and enjoyment while playing the game. For example, the player has clear objectives to achieve; the player can shoot at objects, which are then destroyed. [5]

Visuals and aesthetics: some care was given to the appearance of the environment and entities (player, enemies, skybox) in the game, to create a feeling of immersion [5]

Code [15]

The game compiles and runs; all the necessary libraries were included or documented in the report [5]

Good coding practices: the classes in the game are well organized; repetition of code is avoided; comments were added to the code [10]

Bonus for additional features [up to +10]

Scene graph: up to +2.5. A fully-functional scene graph stores the game world and is used to control the hierarchical transformation of objects.

Terrain: up to +2.5. A terrain that is more than just a plane with a color is used to model the game world. For example, a terrain with varying elevation or varying textures (distinct regions with vegetation, desert, etc.). For games that do not take place in an outside environment, similarly, this additional feature consists in an environment composed of varying geometry or textures.

Enemy AI: up to +2.5. The enemies have a non-trivial AI, for example, based on a state machine, or enemies move in somewhat complex trajectories.

Music and HUD: up to +2.5, depending on the level of complexity.

Other features: up to +2.5, (as long as they are meaningful to the game and require some considerable work in terms of their coding or understanding).

Total: 100 points + 10 bonus.