Assignment 5 — End Game

Lund University Graphics Group

In this concluding assignment you will apply techniques from previous assignments to a bigger, more unified context in the form of a game. By using a conjunction of hierarchical transform, tessellated objects, animation, shaders etc., this serves as a summary of the topics visited throughout the course.

1 Game

You are free to start with the basic code found in `src‣EDAF80‣assignment5.cpp` and `src‣EDAF80‣assignment5.hpp` and populate it with the techniques you have used from the first four assignments.

The objective is to create a program based on topics and techniques of previous assignments with some simple game mechanic. Ideas, guidelines and requirements are provided in the preceding seminar. You are free to chose either of the suggested set-ups (*Asteroids* and *Torus Ride*) — or pursue an idea of your own.

Do not hesitate to post questions or experiences useful for others on Canvas (as well as a presentation of your game once it's done).

A short outline of the suggested set-ups; please see the seminar for more details.

Asteroids

- Set-up: Asteroid field with randomised trajectories and animation
- Objective: Avoid asteroids (penalise collisions) and/or shoot down (reward hits)

Torus Ride

- Set-up: Path of rings made out of tori
- Objective: Fly through the rinds (reward hits and/or penalise misses)

2 Requirements for approval

Your game will be expected to meet the requirements listed in the seminar. Though minor bugs are of course tolerated, the program should be stable enough to be run and tested.

Furthermore, once your game has "Gone gold" you should post a short presentation of it on Canvas in the "End Game Gallery" discussion. The discussion description details what should be included in your post, so please read it, but overall it can be summarised as: title, authors, features, and screenshots/ videos.

A Framework controls

The framework uses standard key bindings for movement, such as W, A, S, and D. But there are also custom key bindings for moving up and down, as well as controlling the UI. All those key bindings are listed in Table 1.

There is only one action currently bound to the mouse, and that is rotating the camera. To do so, move the mouse while holding the left mouse button.

GUI elements can be toggled being a collapsed and expanded state by double clicking on their title bar. And they can be moved around the window by dragging their title bar wherever desired (within the window).

Table 1: Various controls when running an assignment."Reload the shaders" is not available in assignments 1 and 2 of EDAF80, while "Toggle fullscreen mode" is missing from assignment 2 of EDAN35.

Action	Shortcut
Move forward	W
Move backward	S
Strafe to the left	A
Strafe to the right	D
Move downward	Q
Move upward	E
"Walk" modifier	⇧
"Sprint" modifier	Ctrl
Reload the shaders	R
Hide the whole UI	F2
Hide the log UI	F3
Toggle fullscreen mode	F11

B IDE key bindings

To help with getting certain tasks done more efficiently, Table 2 lists key bindings of different IDEs for several common actions.

Table 2: Various keyboard shortcuts for Visual Studio 2019 and 2017, and Xcode.

Action	Shortcut	
	Visual Studio	Xcode
Build	Ctrl + B	⌘ + B
Run (with the debugger)	F5	⌘ + R
Run (without the debugger)	Ctrl + F5	
Toggle breakpoint at current line	F9	⌘ + \
Stop debugging	⇧ + F5	⌘ + .
Continue (while in break mode)	F5	ctrl + ⌘ + Y
Step Over (while in break mode)	F10	F6
Step Into (while in break mode)	F11	F7
Step Out (while in break mode)	⇧ + F11	F8
Comment selection	Ctrl + K, Ctrl + C	⌘ + /
Uncomment selection	Ctrl + K, Ctrl + U	⌘ + /
Delete entire row	Ctrl + X	