CAMERA CONTROL & OPTIMIZATION

DAY 3 OF
"ADVANCED UNITY PROGRAMMING" (2015)
BY MARTIN KRAUS, AAU

CAMERA CONTROL

Task for the morning:

- **Import Unity's standard assets "Cameras" and "Characters" and try out the "FPSController" prefab and the "ThirdPersonController" prefab with the "FreeLookCameraRig" and the "MultipurposeCameraRig" prefabs.**
- **Try to navigate through a narrow, winded corridor using "ThirdPersonController" and "MultipurposeCameraRig" with and without the "ProtectCameraFromWallClip" script. Try to define a camera with a Nav Agent component that is better at following the player through narrow corridors. Which game object should it target? Where should that game object be? Which rotation should the camera use?**
- **Optional: Read Level 6 (about the design of camera controls) of "Level Up! The Guide to Great Video Game Design", 2nd Edition, by Scott Rogers.**

OPTIMIZATION

Task 1 for the afternoon:

- **Instantiate 10,000 Quad game objects at random positions and script their movement at constant velocity to the ground; whenever a Quad reaches the ground, instantiate a new one. Note the performance from the "Stats" display.**
- **Object Pool: Instead of destroying the fallen Quads and instantiating new ones, simply move the fallen Quads up. Note the performance from the "Stats" display.**
- **Data Locality and Flyweight: Instead of using 10,000 game objects, render 20,000 triangles of the same size and shape (in one mesh) and script the same movement by updating their vertex positions. Note the performance from the "Stats" display.**
- **Use Unity's Profiler to get more insights about the differences.**

OPTIMIZATION

Task 2 for the afternoon:

- **Dirty Flag: Compute a large L-system with one or more continuous parameter (e.g. an angle of a rotation) in the Update function. Use a dirty flag to update the L-system only if one of the parameters has changed.**
- **Instead of computing the L-system in one call of the Update function (and blocking the rest of the application), use coroutines to compute the L-system over several frames.**
- **Optional (and not related to optimization but to software architecture): Read about Singletons (in particular as managers), which are very common to introduce a central interface for functionality under development.**