Day One

What is Computer Graphics?

There are several different definitions of computer graphics, but it generally means *any* use of computers to create and manipulate images. The term *computer graphics* was coined in 1960 by William Fetter to describe a new design method used at Boeing for cockpit ergonomics.

What are the branches of Computer Graphics?

There is no one way for splitting the field of Computer Graphics, and it generally depends on how you approach it. For the purposes of this course, we will take a computer generated image and figure out what we need to generate it. This will yield the branches of computer graphics.

1. **Computer Modelling:**
   - Polygonal modelling: uses polygons to approximate the surface of a model. Typically consists of triangles, though quads and other regular n-gons are also used. The main advantage is that graphics hardware is designed to process these quickly, but they do have some challenges in terms of compositing them to create bigger objects, accurately representing curvature, etc.
   - Implicit modelling: uses implicit functions of the form $f(x, y) = x^2 + y^2$. The main advantage is that composition of implicit surfaces is trivial, along with intersections. They also give an immense amount of control for modelling things like skin. The downside is that they do not have a direct representation that the hardware can process, so presenting them is expensive.
   - Parametric modelling: uses parametric functions to model surfaces, which allow for a high degree of intuitive control for creating complex smooth shapes. Unlike implicit surfaces, they can be quickly converted to polygons for presentation, which means that increasing the resolution of the generated model is very easy. The downside is that they do still require a conversion, and that they mostly rely on quads, which makes modelling certain kinds of models more difficult.
   - Procedural modelling: uses a given set of *rules* to generate the model. This can be very useful for generating things like terrains or trees, which follow specific patterns. The output of procedural modelling can be polygons which makes it fast to present. The downside is that the functions used to control them can be difficult to model and evaluate.
2. **Computer Animation:**
   - Interpolation-based Animation: the idea is to have the artist specify two positions at two different times t_0, t_1. The goal is then to produce the intermediate states that the object needs to be in to move from t_0 to t_1. The points are called *key frames*, while the process of generating

the intermediate states is called *interpolation*.

- Kinematics: objects are represented as *chains* composed of *bones* (rigid portions that do not move) and *joints* which connect the bones together. The movement can then be specified in one of two ways: the artist can provide the individual angles of rotation for each joint, in which case the system then needs to compute the final position for the chain. This is called *forward kinematics*. Alternatively, the artist specifies the final position, in which case the system needs to compute the rotation angles to get there. This is called *inverse kinematics*.
- Motion Capture: the idea is to use a live model that is outfitted with tracking points. Special cameras are then used to capture the motion and translate the images of the tracking points into 3D coordinates. These can then be mapped onto a 3D model and transfer the animation.
- Physically-based Animation: relies on using the laws of physics to model the way objects behave and move in reality.
- Fluids: simulates the way fluids such as liquids and gases behave and move.

3. **Computer Rendering:**
   - Offline rendering: the rendering process is allowed to take as long as it needs. Generally this is paired with more realistic rendering techniques in order to produce the *best* possible final image. The duration can be anywhere from minutes to days or even weeks. Nowadays these are performed in massive clusters of graphics hardware or servers to accelerate the rendering.
   - Real-time rendering: while offline rendering has no bounds on time, real-time rendering is defined as any render where frames are produced fast enough for humans to interact with in real-time. For most applications, this is defined to be 60FPS, but it can be higher than this.
   - Physically-based rendering: similar to physically-based animation, PBR focuses on simulating and modelling the way light behaves in the real world. This involves *very* expensive calculations that model the way light collides and is reflected/scattered by surfaces.
   - Photorealistic rendering: attempts to produce an image that would match as much as possible a photograph taken under equivalent conditions. It is important to note that while PBR falls under the umbrella of photorealistic rendering, not all photorealistic techniques are physically based. We will discuss what this means later on.
   - Non-photorealistic rendering: sometimes called *artistic* rendering. These are techniques were the objective isn't to produce something realistic, but rather something that fits a specific aesthetic.

What Will We be Focusing On?

We will mostly be sticking to computer rendering, and cover some topics of computer modelling. Due to the inherent complexity of the topic, we will not discuss animation aside from some brief references later on.