**GAM531/DPS931**

**Material reviewed in class for final exam**

* **Rendering pipeline**
  + **Concept**
    - **vertex shader -> fragment shader**
  + **Shaders: concept and usage in pipeline**
    - **fragment shader, vertex shader (necessary shaders)**
    - **geometry shader, tesselation shader (optional shaders)**
  + **Allocating buffers in VRAM to load vertex coordinates / vertex colors / texture coordinates, etc.**
* **Linear transformations**
  + **Rotation**
  + **Translation**
  + **Scale**
  + **Model-View matrix**
* **Camera**
  + **Concept**
  + **Camera Matrix**
  + **Camera properties:**
    - **Near/Far plane, field of view (FOV) in horizontal/vertical direction**
    - **Culling frustum**
  + **Camera extrinsic parameters**
    - **Position**
    - **Forward vector**
    - **Up vector**
* **Model loading**
  + **Concept**
  + **OBJ standard**
* **Texture Mapping**
  + **Concept**
  + **Loading images into VRAM**
  + **Texture coordinates**
  + **Applying properties to texture image in VRAM**
  + **Applying texture to polygons using texture coordinates**
* **Collision**
  + **Detection**
    - **Box-Box**
    - **Circle-Circle**
    - **Triangle-Triangle**
  + **Collider or wrapper simplifies collision detection**
  + **Handling**
* **AI in games**
  + **Concept**
  + **Application in games**
  + **Algorithms:** 
    - **Simple AI (squash game, )**
    - **Path finding**
    - **A\* search**
    - **Minimax (board games, turned based, small number of game states)**
  + **Advanced AI: Machine Learning, OpenAI**
* **Game Engines**
  + **Definition**
  + **Main components** 
    - **Physics**
    - **Scene Management**
    - **Game Objects**
      * **Particle Systems**
      * **Animated objects**
      * **Static objects**
      * **Lights**
  + **Used game engines in industry**
  + **Unity**
  + **Unreal**
  + **CryEngine**
  + **GameMaker**
  + **Godot**