Part 2 : Game Engine

* Agreements
  + Coding Styles
    - Pascal Case
    - Braces on new line
    - Variable prefixes such as “bIsAThing” on Booleans only
  + Project Structure
    - Same structure as shown in class
  + Minimum PC requirements
    - Any old potato
* Classes
  + Input class (cross platform input interface)
  + Resource/Asset Manager
  + Audio Manager
  + Graphics Interface
    - OpenGL, DirectX
  + Main/Game Loop
  + Initialization (System Specs gathering)
  + Utilities (Math Libraries)
    - Vectors
    - Functions
      * Normalize
      * Look At
      * Etc.
  + Object (Logic only, don’t have transform, cannot be placed, or spawned in the world)
  + Actor (Anything that can be placed or spawned in the world)
    - Transform
  + Log class
  + World
    - Has function that goes through objects in the scene and updates them, checks for actors and draws them.
* Actor Components
  + AI
  + Collision
  + Renderer
  + Input (listens for inputs from input class)
  + Pickup
* Actors/Classes
  + Game class
    - Handles game time/fps
    - Init resources?
  + Character class (Virtual)
    - Base character properties/methods
      * Collider/bounding box for collision detection
  + Main character class (player controlled – inherits from Character class)
    - Takes input and sets appropriate vector direction
  + AI character class (inherits from Character class)
    - State machine (wander or follow)
      * Collectible characters use wander until they are collected and become part of the main character
      * Enemy characters either wander or follow
  + Non-moving enemy class (inherits from Character class)
    - Static enemies
  + In game background class
    - Sets the background resource and scrolls it according to the player’s movement
  + HUD class
    - Shows the current score and size of the player’s group