

Notes:
Mono: Classes which are tagged with 'Mono' inherit from MonoBehaviour (and thus all classes which inherit from that class do as well). These scripts are attached to an instance of a gameObject in the scene. Thus, these scripts have access to the gameObject they're attached to (including its other scripts, world position, colliders, etc.). Common mono methods we use: Awake(), OnEnable(), and Start() are called at initialization (in that order). FixedUpdate(), Update(), and LateUpdate() are called each frame (in that order). OnCollisionEnter() and OnTriggerEnter() are called when collisions are detected. OnDisable() called when the scripts gameObject object is disabled.

Naming Conventions: Data members beginning with 'm_' are generally available to be set in the editor. Members beginning with '_' are generally internal to the class, and updating each frame.

Patterns: Some patterns are described in detail below. Others (for which the explanation seems unnecessary) are in the diagram, near the classes which implement it.

MVC Pattern

Model - Player, PlayerController
View - GameObject in Scene
Controller - PlayerController
User interacts with PlayerController (controller) through PlayerInput, which updates Player/PlayerController/Animator etc. values (model), which then get reflected in the game scene (view)

Interface Pattern

Implemented by several classes, including Interactable, Attack, and Character. They each provide range of methods & data designed to be final, extended, or overridden.

