**Enemies**

Shield Skeleton:  
**Medium speed walk** straight towards the player. (Maybe in between the player and ranged enemies synergy\*)

At a certain **range from the player** or if it **gets damaged** it puts its **shield up** and takes greatly reduced damage, in the second case the triggering hit is not reduced but subsequent hits are.

When attacking it winds up by putting its shield up and doing a **very short charge attack** towards the player. The direction is taken when the attack is triggered.

The attack deals damage and ideally either **interrupts the player’s action** or **knocks him back** slightly. (This can be added after)

After attacking, during its attack cooldown the **enemy waits in place** then resumes walking towards the player to attack.

**Defend ranged enemies mechanic**: The shield enemy will try to get **between the player and a selected ranged enemy** while maintaining a certain distance from the enemy. (Could be distance between ranged enemy and player / 2 aka somewhat halfway in between) If the player is closer to the defendee then the shield enemy put his shield away and start running towards the defense location. Putting his shield back down should take a certain amount of time.   
This could create a **player tactic** where the player runs past the shield enemy towards the defendee making the shield skeleton start running only for the player to turn on it and attack. Different shield enemies can have different reactions to this but the base tactic of triggering an action by running past it towards the defendee could stay the same across all shield enemies that defend another enemy.   
An example of a **different reaction** could be a shield enemy that takes out a weapon, possibly dropping its shield, and tries to do a dash attack with a long wind-down at the player, running past this enemy could have a higher risk but a large reward because of that attack’s long wind-down, so the player would run past the enemy, avoid the dash attack, then turn to dispose of the defenseless shield enemy.

Ranged Skeleton:  
This can be a bow or a simple caster.   
**Slow walk** and tries to **stay away from the player**. (How is to be determined, random raycast away from the player to check for walls? And to get the a\* grid node)

While walking towards the player to get in range to attack, it **checks the distance every few frames** (or in seconds) and **if it has line of sight**.

When attacking its wind up needs to be **excessively clear** in **when the attack is about to be unleashed** and **show its attack direction**.

After its wind down, when the enemy’s attack is still on cooldown it **walks away from the player**. Once its attack is ready it walks back towards the player to attack.

Enemy Movement: Script that holds the different types of movement the enemy can do. Or possibly just keep track of the current movement state. Run Away – Chase – Defend – etc.  
Only one movement type can be active at once.  
The movement types can be: Moving directly to location – Following path.

Enemy States: for each action script enemy specific action script, example:   
Shield Skeleton Bash (attack) states; windUp, Swing, windDown, cooldown, checkingDist.   
Shield Skeleton ShieldUp (ability) states; shieldUp, shieldDown, shieldUpCooldown, forcedDown.

Logic:  
Each stage of the attack has an array of sprites that it cycles through to create that animation. It also has an array of floats designating when events should be played during the stage.

The number of stages could be variable by having an array of stage **classes** that each contain an array of animation sprites, array of event timings (float) and an array of events.  
  
  
On the scriptable object enemy base there is an array of ability classes.  
  
Ability Class -> Everything they do, attacks, guard, spells, etc. except A\* walking and aggro.  
  
 float totalDuration: The total duration of the enemy doing the ability.  
  
 Sprite sprites Array: An array of sprites to play during the specific ability.  
  
 float spriteChange Array: An array of floats denoting when to change to the next sprite.  
  
 UnityEvents/float methods Array: Either actual UnityEvents or just floats and the attack script takes care of calling the right method.  
  
Each enemy has his own “action” script derived from Enemy\_Action for the sake of references. In their own action script there are references to different enemy specific ability scripts as well as generic methods shared by all the enemy action scripts such as Stop, Death, Pause, Checks, etc.   
  
Once an enemy is aggroed, it calls to its a generic method on the action script to start doing checks. Checks that are specific to each enemy, for the ShieldSkeleton this is ShieldBashCheck and ShieldUpCheck which are his two abilities. These checks can be done in their own scripts referenced by the action script; they can then speak and interact with each other.

General Enemy Behavioral Types;

**Area**: Stays within a certain radius of its spawn point (or area point), can chase the player if it is within the radius, will go back towards its area after a chase, chases do not necessarily end with an attack as they have a maximum timer.

**Chase:** Enemy that once aggroed will start chasing the player anywhere. The chases will still be timer based so the enemies will still stop and take breaks but will always resume chase.

**Wander:** Walks around randomly and attacks the player only when it is within range, could have a small aggro range where it will initiate a short chase to try and attack but after each attack or chase it resumes wandering if the player is out of range.

**Poke:** Once aggroed will chase the player to attack, but after every attack it will run away from the player for a certain time then try to attack again.

**Defender:** Stands between the player and a point or another enemy that it is defending, attacks when the player within range or closer to what they are defending then they are.