File Diffs

AbilityActionChain.h

* Function modified
  + PushAction
  + ChargeAction
* Functions added
  + PullAction
  + BackupAction
  + PushSpeedAction
  + MovementSpeedAction
  + RotationAction

AbilityActionChain.cpp

* OnAbilityEnd
  + Bug fix for when block is never cleared
* PushAction
  + Added float speed
  + Adjusted function calls to account for new variable
* PullAction
  + Input float distance and float speed
  + Return bool
  + Takes ability owners forward vector and pulls the porf hit toward ability owner
* BackupAction
  + Input float distance and float speed
  + Return bool
  + Gets porf hit forward vector and pushes character away from that vector
* PushSpeedAction
  + Input float pushSpeed
  + Return bool
  + Sets the porf hit push speed
  + Consider removing
* MovementSpeedAction
  + Input float movementSpeed
  + Return bool
  + Sets the porf hit movement speed
* RotationAction
  + Float rotSpeed and float lerpRatio
  + Return bool
  + Sets the ability owners rot speed and lerpRatio
* ChargeAction
  + Added float speed
  + Adjusted function calls to accept speed variable

PorfCharacterBase.h

* Push
  + Added float speed parameter
* Functions added
  + SetPushSpeed
  + GetPushSpeed
  + SetMovementSpeed
  + SetRotationVariables
  + IncreaseStunMeter
  + RegenerateStunMeter
  + CheckStatusEffect
  + SnapTurn
* Properties added
  + Float m\_manaMeterDelay
  + Float m\_maxStunMeter
  + Float m\_stunMeter
  + Float m\_stunMeterRegen
  + Float m\_stunMeterDelay
  + Bool m\_abilityHeldDown
* Private variables added
  + Float m\_delayManaregen
  + Float m\_delayStunMeterRegen

PorfCharacterBase.cpp

* Tick
  + Added Function call for RegenerateStunMeter
* RestoreMana
  + Added a delay mana regen logic
* CastAbility
  + Added Error checking to avoid ability overlap between abilities
* DrainMana
  + Adjusted to account for mana drain
* IncreaseStunMeter
  + Input float amount
  + Return bool
  + Logic to increase the stun meter by the amount inputted
* RegenerateStunMeter
  + Input float deltaTime
  + Regenerates stun meter over time
* Move
  + Added error check to make sure character is able to move
* Push
  + Added float speed to input
  + Error check to make sure character can move
* SetPushSpeed
  + Input float pushSpeed
  + Sets character push speed
* GetPushSpeed
  + Gets characters push speed
* SetMovementSpeed
  + Input float movementSpeed
  + Sets corresponding variable in the character’s movement component
* SetRotationVariables
  + Input float rotSpeed and float lerpRatio
  + Sets corresponding variables in the character’s movement component
* SnapTurn
  + Input FVector direction
  + Calls character’s movement component function Snap
* ReceiveDamage
  + Logic for when player is invulnerable will receive no damage
* CheckStatusEffect
  + Input StatusEffects effect
  + Return bool
  + Searches the status effects on a character and returns true if found
* RemoveFinishedStatusEffects
  + Destroys the status effects actors

PorfCharacterMovementComponent.h

* Function modifies
  + Push
* Functions added
  + SetPushSpeed
  + GetPushSpeed
  + SetMovementSpeed
  + SetRotationSpeed
  + Snap
* Properties
  + Float m\_maxMovementSpeed
  + Float m\_ranpingSpeed
  + Float m\_lerpRatio
  + Float m\_pushSpeedDropRatio
* Private Variables
  + Loat m\_currentPushSpeed
  + Float m\_pointToLowerSpeed
  + Flaot m\_quadraticA
  + Bool m\_calculatePush

PorfCharacterMovementComponent.cpp

* TickComponent
  + Added error checking so movement speed does not exceed the maximum
  + Adjusted CalculateRotation to take in deltaTime as a parameter
  + Adjusted CalculateForces to take in DeltaTime as a parameter
* CalculateRotation
  + Input float DeltaTime
  + Logic for the mechanic that handles the character rotation based on the current speed of the character
  + Used RInterpTo function for rotation interpolation
* Push
  + Input FVector direction, float distance, float speed
  + Logic for the push mechanic
* CalculateForces
  + Input float DeltaTime
  + The logic for the push mechanic to decrease speed from a certain point along the push distance
* SetRotationSpeedAndLerpRatio
  + Input float rotSpeed and float lerpRatio
  + Set the corresponding variables

Effect.cpp

* Added invulnerable to enum StatusEffects