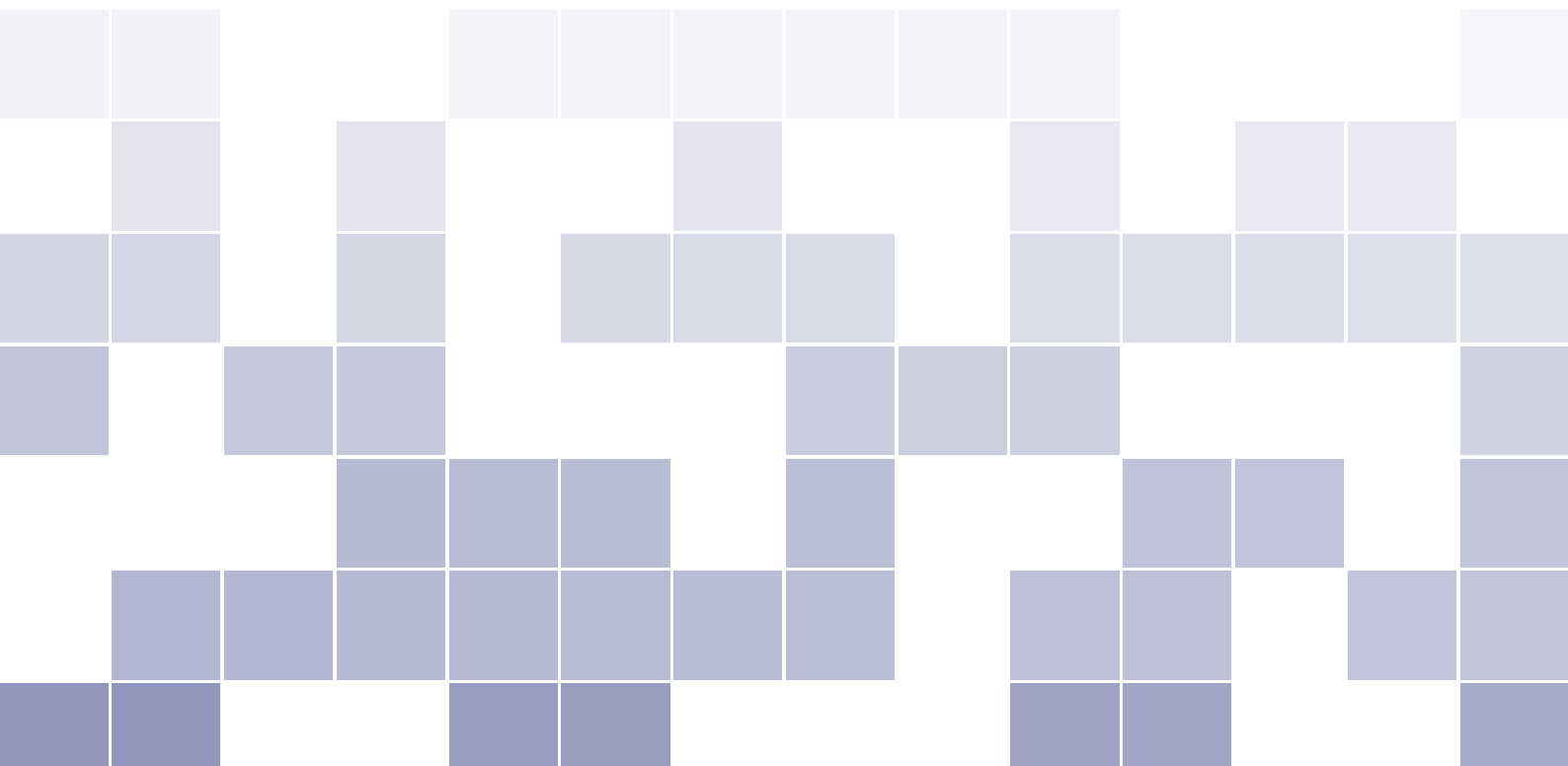


# PARALLAX

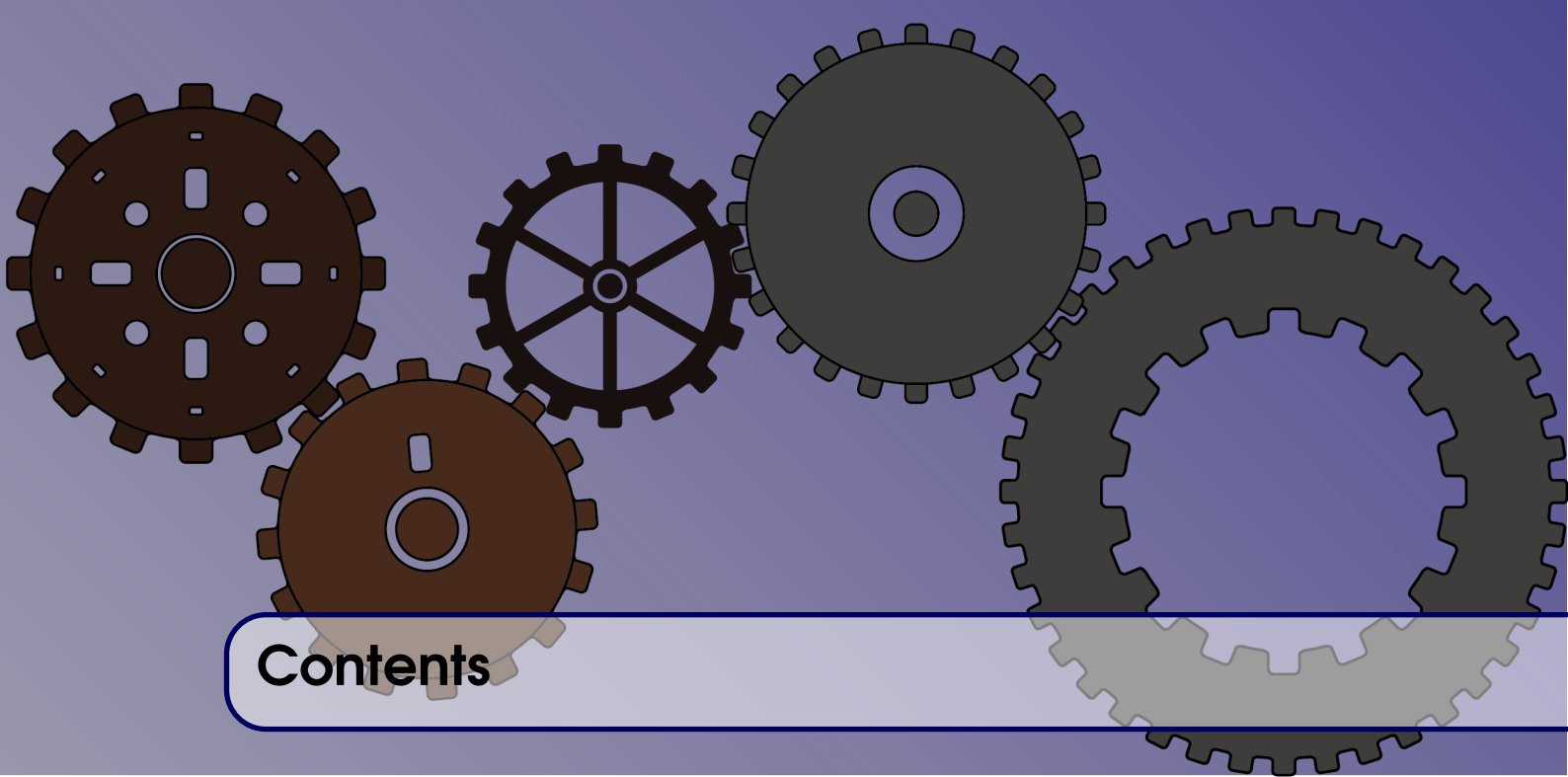
(Working Title)

Technical Design Document

Kyle Salitrik



Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



# Contents

I	Classes	
<b>1</b>	<b>Parex Controller</b>	<b>6</b>
1.1	Variables	6
1.2	Functions	7
<b>2</b>	<b>Parex Character</b>	<b>9</b>
2.1	Variables	9
2.2	Functions	10
<b>3</b>	<b>Base Enemy</b>	<b>11</b>
3.1	Variables	11
3.2	Functions	12
3.3	Enemy AI Tree	13
<b>4</b>	<b>Base Weapon</b>	<b>15</b>
4.1	Variables	15
4.2	Functions	16
<b>5</b>	<b>Base Projectile</b>	<b>17</b>
5.1	Variables	17
5.2	Functions	17

Index .....	19
-------------	----



# Classes

<b>1</b>	<b>Parex Controller</b> .....	<b>6</b>
1.1	Variables	
1.2	Functions	
<b>2</b>	<b>Parex Character</b> .....	<b>9</b>
2.1	Variables	
2.2	Functions	
<b>3</b>	<b>Base Enemy</b> .....	<b>11</b>
3.1	Variables	
3.2	Functions	
3.3	Enemy AI Tree	
<b>4</b>	<b>Base Weapon</b> .....	<b>15</b>
4.1	Variables	
4.2	Functions	
<b>5</b>	<b>Base Projectile</b> .....	<b>17</b>
5.1	Variables	
5.2	Functions	



## 1. Parex Controller

The Parex Controller Class handles all input from the player and delegates it to actions within the world through the Player Character Class and animations.

### 1.1 Variables

Variable	Type	Description
AxisModifier	Int	The axis modifier is +1 if facing right and -1 if facing left. It determines the lateral movement direction.
MoveHorizontal	Float	The horizontal input axis value from the keyboard (1) or thumbstick (-1 to 1) after being modified by the axis modifier.
MoveDepth	Float	The depth input axis value from the keyboard (1) or thumbstick (-1 to 1).
MouseSensitivity	Float	Multiplier for the mouse input.
InvertMouselook	Bool	Determines whether or not to have an inverted mouselook.
CameraTurning	Bool	Set to true during camera movement to prevent restarting the camera movement if button is rapidly pressed.
SprintMove	Float	Value of movement after being modified by sprint modifier.
Sprinting	Bool	Determines whether or not the character is sprinting.
Jumping	Bool	Determines whether or not the character is jumping.
Turning	Bool	Determines whether or not the character is turning.
FacingRight	Bool	Determines whether or not the character is facing right.
IsAiming	Bool	Determines whether or not the character is aiming.
AimDownSights	Bool	Determines whether or not the character is aiming down sights.
Reloading	Bool	Determines whether or not the character is sprinting.

Sprinting	Bool	Determines whether or not the character is reloading.
UseController	Bool	Determines whether to use the controller or mouse look inputs.
PlayerCharacter	Class	Reference to player character.
FrontWall	Class	Used to reference all front wall elements and toggle visibility.
ClampAngle	Float	Max/Min angle that player can look.
GameInstance	Class	Reference to the game instance.

## 1.2 Functions

Function	Description
Initialize	Create variable references to player character, movement, and game instance.
SetMovementAxes	Takes in the horizontal and depth input axis values and assigns them according to the direction the player is facing and the camera mode.
Move	Gets the actor's rotation, the Axis Modifier (+1 for facing right, -1 for left) and whether or not the player is sprinting, and adds movement inputs accordingly.
ControllerLook	Takes the input axes from the right thumbstick and creates a parabolic output that is clamped to the allowed viewing angle. It then sets the aim-offset angle.
PitchLook	Takes a mouse input value and gets the current aim offset, runs the CheckAxisBounds function and then adds the output of that function to the aim offset.
YawLook	Does the same thing as PitchLook, except for the yaw axis.
CheckAxisBounds	Takes in the current axis value and input values for pitch or yaw and checks whether or not they are within an allowable clamp angle. If they are out of bounds, it sets the angle to the maximum axis value.
Sprint	If the character is moving forward, increase the max runspeed by 1.5x and play sprinting animation.
Jump	Checks if the character is not in cover or moving backwards, if that is true, allow the character to jump.
TakeCoverPressed	Call the TakeCover function in the Parex Character class.
SetAiming	Sets the boolean for whether or not the character's laser sight is on and then calls the ChangeViewMode function in the Parex Character class.
AimDownSights	If the character is currently aiming, sets the boolean of AimDownSights to true and calls the ChangeViewMode function in the Parex Character class.
Fire	If the character is not sprinting or reloading, it calls the FireWeapon event in the Parex Character class.
Reload	Checks if any magazines are left, and that the character is not sprinting or currently reloading and then calls the ReloadWeapon function in the Parex Character Class.
Turn180	As long as the character is not aiming, in cover, sprinting, currently turning, or jumping, the function gets the starting rotation of the actor, and calls the rotate actor function.

RotateActor	Activates the turning animation and uses a LERP function to turn the character 180 degrees. At the end of the timeline, it flips the axis modifier and clamps the character to 90 or -90 degrees in order to keep the character moving straight.
TurnCamera	If the dynamic camera is activated, it LERP rotates the camera at the same time as RotateActor between 15 and -15 degrees by calling the ChangeViewMode function in the Parex Character class.
Pause	Pauses the game and displays the Pause Menu widget.
Jumped	Utility function to change the Jumping boolean.
CoverBackwards	Utility function for the Parex Character class to set the axis modifier and rotate the mesh if cover is located behind the character.
Reloaded	Utility function to change the Reloading boolean.
StopSprinting	Utility function to change the Sprinting boolean
UpdateSettings	This function is called when the game is unpaused to update any variables that may have been changed in the settings menu: Invert Mouselook, Use Controller, Mouse Sensitivity, Static Camera.





## 2. Parex Character

The Parex Character class handles all interaction between the world objects and the character mesh.

### 2.1 Variables

Variable	Type	Description
AxisModifier	Int	The axis modifier is +1 if facing right and -1 if facing left. It influences the player movement.
CoverAvailable	Bool	True when player is near cover.
CoverRotation	Rotator	Difference between the character and cover's normal vector.
InCover	Bool	True when player is in cover.
EndOfCoverLeft	Bool	Shows when player is at the left edge of a cover box.
EndOfCoverRight	Bool	Shows when player is at the right edge of a cover box.
FacingCover	Bool	True when the character normal and cover normal vectors are the same.
SpringArmStart	Transform	Set to the current spring arm location before moving.
SpringArmEnd	Transform	Set to the final spring arm location before moving.
SpringArmLengthStart	Float	Starting spring arm arm length.
SpringArmLengthEnd	Float	Ending spring arm arm length.
WeaponToSpawn	Actor	Used to set the initial weapon to spawn the character with.
EquippedWeapon	Class	Reference to the equipped weapon.
PlayerController	Class	Reference to the owning player controller (Parex Controller).
Height	Float	Distance between actor and ground.
Speed	Float	Max walk speed of character.
MaxHealth	Float	Maximum health player can have.
CurrentHealth	Float	Player's current health.

PreviousHealth	Float	Set when taking damage and used to LERP health bar.
Dead	Bool	True when health $\leq 0$ .
StaticCamera	Bool	True if camera is set to static in settings.
PreviousAngle	Float	Used to move lightsource angle smoothly. A global lightsource is used in Platformer mode to increase visibility.

## 2.2 Functions

Function	Description
Get Height	Gets the character's current height while jumping in order to transition animation states.
InitializeCharacter	Sets all instance variables and spawns the characters weapon.
SetAimingMechanic	Turns on and off the gun's laser and changes a boolean that determines whether or not to aim down sights.
ChangeViewMode	Sweeps the camera between the Platformer and Over-the-Shoulder views.
DeterminSpringArmPath	Calculates the path that the camera spring-arm must travel to change views.
SweepSpringArm	Moves the spring arm over a period of time using liner interpolation. If dynamic camera is on, it also angles the camera to bias the direction the character is facing.
SetFrontWallVisibility	Sets the visibility of the front wall to invisible while in the platformer mode.
CheckForCover	Determines whether or not the character is standing within a cover collision box and returns a boolean value.
TakeCover	If cover is available, run the set in cover function.
GetEndOfCover	Checks whether the character is at the edge of cover and returns a boolean.
SetInCover	If the player is facing away from cover, first turn them to face the cover and then play the cover animation. If dynamic camera is enabled it also turns the camera to bias the direction of movement.
CoverBackwards	Turns the character mesh to face cover and changes axis modifiers accordingly.
FireWeapon	If the character is able to fire, trigger the fire weapon function of the weapon class.
Reload	Trigger the reload function of the weapon class and play reload animation.
TakeDamage	When the take damage event message is received, run the calculate health and update health functions.
Calculate Health	Takes the max and min damage caused by the incoming projectile, critical chance and skeletal mesh bone and determine the amount of damage caused.
UpdateHealth	Updates the healthbar using a LERP over time for smooth transitioning.
Die	Checks to see if the character's health is at or below 0. If it is, the character dies, ragdolls, and the pause menu is shown.



## 3. Base Enemy

The base enemy character includes all of the functions that the AI controller references and handles the actions that the enemy character performs.

### 3.1 Variables

Variable	Type	Description
Waypoint1	Location	Location of first position waypoint for static patrolling enemies.
Waypoint2	Location	Location of second position waypoint for static patrolling enemies.
EnemyOrigin	Location	Spawn location of enemy, used for determining distance from spawn.
CombatStatus	Enum	Determines whether the enemy is guarding, patrolling or attacking.
CombatType	Enum	Determines the type of enemy: HeavyCover, Assault, or Cover.
CurrentHealth	Float	Current health.
MaxHealth	Float	Max health.
Dead	Bool	True if CurrentHealth $\leq$ 0.
PlayerCharacter	Class	Reference to player character spotted.
DamageMin	Float	Minimum damage that the current weapon causes.
DamageMax	Float	Maximum damage that the current weapon causes.
CritChance	Float	Chance to cause a critical hit.
IsFiring	Bool	True when enemy is firing weapon.
AimAt	Name	Randomly selected bone of character to aim at.
AimLocation	Vector	Vector between enemy's right hand and player bone aimed at.
ShotgunFireCount	Int	Randomly chosen amount of shots to fire.

ShotgunBurstCount	Int	Number of pellets to shoot for each shotgun burst. Also used to divide damage per pellet.
ShotgunSpreadRadius	Float	Radius of cone for spread of pellets at ShotgunSpreadDistance.
ShotgunSpreadDistance	Float	Height of cone for bullet spread.
ShotgunMinShots	Int	Minimum number to assign to fire count while using a shotgun.
ShotgunMaxShots	Int	Maximum number to assign to fire count while using a shotgun.
RifleFireCount	Int	Randomly chosen amount of shots to fire.
RifleSpreadRadius	Float	Radius of cone for spread of pellets at RifleSpreadDistance.
RifleSpreadDistance	Float	Height of cone for bullet spread.
RifleMinShots	Int	Minimum number to assign to fire count while using an assault rifle.
RifleMaxShots	Int	Maximum number to assign to fire count while using an assault rifle.
SniperFireCount	Int	Randomly chosen amount of shots to fire.
SniperSpreadRadius	Float	Radius of cone for spread of pellets at SniperSpreadDistance.
SniperSpreadDistance	Float	Height of cone for bullet spread.
SniperMinShots	Int	Minimum number to assign to fire count while using a sniper rifle.
SniperMaxShots	Int	Maximum number to assign to fire count while using a sniper rifle.
ClampAngle	Float	Maximum angle the enemy can offset their aim before turning to face the player.
CoverAvailable	Bool	True when enemy is near cover.
CoverRotation	Rotator	Rotation difference between enemy normal vector and cover normal vector.
TakingCover	Bool	True when in cover.
IsAiming	Bool	True when enemy is aiming at player.

### 3.2 Functions

Function	Description
InitializeEnemy	Initializes instance variables on enemy spawn.
CheckForCover	Returns true if cover is available and sets cover rotation.
SeePlayer	Calls SetCombatStatus if player is seen by enemy.
HearPlayer	Calls SetCombatStatus if player is heard by enemy.
SetCombatStatus	If player is seen or heard, set enemy to combat and recursively call the function for any other enemies within 3 meters.
TakeDamage	If the enemy is hit by the player and not in combat, flag the enemy for combat. Afterwards, calls CalculateHealth
CalculateHealth	Takes the MinDamage, MaxDamage, and CritChance of the weapon as well as the hit bone and calculates the damage to deal to the enemy.
Die	If health <= 0, set health = 0, Dead boolean to true, stop enemy from performing any other actions and die with a ragdoll effect..

FireWeapon	Based on the combat type, call either FireShotgun, FireRifle, or FireSniperRifle.
PrepareToFire	Used by all weapons, it takes the minimum and maximum number of shots to fire, waits between 1 and 3 seconds and then proceeds to the next function in the chain based on the weapon after returning the FireCount.
FireWeaponBurst	Takes the weapon animation to play, projectile count, projectile spread, spread distance, projectile type and damage multiplier as arguments. It then sets the damage of the projectiles to spawn, plays the firing animation for the equipped weapon, creates the number of projectiles given and fires at a random direction within the spread cone.
FireShotgun	Calls PrepareToFire, sets ShotgunFireCount and then calls ShotgunFiring on a timer for the number of shots set by PrepareToFire.
ShotgunFiring	Set FireCount to 0. While FireCount<ShotgunFireCount, calls FireWeaponBurst. Once FireCount == ShotgunFirecount, reset FireCount to 0 and randomly assign a new interger to ShotgunFireCount within the given range.
FireRifle	Calls PrepareToFire, sets RifleFireCount and then calls RifleFiring on a timer for the number of shots set by PrepareToFire.
RifleFiring	Set FireCount to 0. While FireCount<RifleFireCount, calls FireWeaponBurst. Once FireCount == RifleFireCount, reset FireCount to 0 and randomly assign a new interger to RifleFireCount within the given range.
FireSniper	Calls PrepareToFire, sets SniperFireCount and then calls SniperFiring on a timer for the number of shots set by PrepareToFire.
SniperFiring	Set FireCount to 0. While FireCount<SniperFireCount, calls FireWeaponBurst. Once FireCount == SniperFireCount, reset FireCount to 0 and randomly assign a new interger to SniperFireCount within the given range.

### 3.3 Enemy AI Tree

The AI Tree controls the behavior of the enemy based on it's combat type and status.

## Out Of Combat

- CombatStatus: Static Patrol
  - Sets the waypoints assigned when creating the character (First loop only)
  - Moves to Waypoint 1
  - Waits 3 seconds +- 1 second
  - Moves to Waypoint 2
  - Waits 3 seconds +- 1 second
  - Repeat
- CombatStatus: Dynamic Patrol
  - Gets a random waypoint within a specified radius
  - Moves to that wapoint
  - Wait for 3 +- 1 second

- Repeat
- CombatStatus: Guard
  - Stays stationary

## In Combat

- CombatType: Assault
  - Player out of firing range:
    - \* Stop attacking
    - \* Set move speed to 500
    - \* Move to Player within a given radius
  - Player is within firing range:
    - \* If enemy is facing player, set move speed to 250 and move toward player if they move out of a given distance.
    - \* If enemy is not facing player, face the player.
    - \* Attack the player.
- CombatType: Cover/HeavyCover
  - If attacking, stop.
  - Find nearest cover. HeavyCover enemies try to find cover further from the player than normal Cover enemies.
    - \* If cover is available, go to the nearest cover.
    - \* If cover is not available move toward the player and revert to assault attack pattern.
  - If enemy is in cover, check if player is in firing arc.
    - \* If true, stand up and aim at player, then attack them.
    - \* If false, move to another cover that is further from the player.
  - If character is too close to the enemy, move to a cover that is further away if possible.



## 4. Base Weapon

The Base Weapon class is the parent class from which specific weapons are created.

### 4.1 Variables

Variable	Type	Description
AmmoCurrent	Int	Current weapon ammo.
AmmoMax	Int	Maximum weapon ammo.
AmmoMagazines	Int	Current weapon magazines.
HeatCurrent	Int	Current heat buildup.
HeatMax	Int	Maximum heat tolerance.
HeatPrevious	Int	Previous heat, used to LERP heat bar on UI.
HeatGenerationRate	Int	Rate at which the weapon generates heat.
HeatDissipation	Int	Rate at which the weapon dissipates heat.
Overheated	Bool	True when current heat == max heat.
DamageMin	Float	Minimum damage done by weapon.
DamageMax	Float	Maximum damage done by weapon.
CritChance	Float	Critical chance of weapon.
ImpactForce	Float	Determines the impact force of the projectiles fired by the weapon on the environment.
LaserOn	Bool	True when aiming laser is on.
LaserStart	Vector	Starting location of the laser sight.
LaserEnd	Vector	Ending location of the laser sight. Used to make laser not penetrate objects.
Laser	Particle System	Reference to the laser sight particle system.
FiringAnimation	Animation	Firing animation to use for the weapon mesh.
ProjectileClass	Class	Type of projectile the weapon fires.



RecoilVertical	Float	Amount of vertical recoil caused by the gun.
RecoilHorizontal	Float	Amount of horizontal recoil caused by the gun.
FireRate	Float	How fast the weapon fires.
IsFiring	Bool	True when weapon is firing.

## 4.2 Functions

Function	Description
InitializeWeapon	Initialize heat dissipation timer and timer play rate.
UpdateTick	Updates whether the overheal alarm should go off or whether or not the laser sight is active.
ReloadGun	Checks whether or not any magazines are available, and if there are any remaining, discards the current ammo, plays the reload animation and updates the UI ammo bar.
FireWeapon	Checks if the gun is not overheated and still has ammo, then calls the WeaponFire, GenerateHeat and Recoil functions if the conditions are true. If the conditions are false, it calls the StopFiring function.
WeaponFire	Plays the firing animation for the mesh, calls CalculateShootInfo and then spawns a projectile with a trajectory created by that function. Finally, it sets the damage information of the projectiles that are fired.
CalculateShootInfo	Gets the trace between the muzzle of the weapon and the first object that blocks the trace and returns a trajectory from the muzzle to the intersected object.
GenerateHeat	Sets the previous heat to the current heat level, add the heat generated for a single shot to that value and then calls the UpdateHeat function. If the current heat $\geq$ max heat, it sets the weapon to be overheated and calls the overheal alarm.
UpdateHeat	LERPs the heat bar between the previous heat and current heat up to the maximum heat.
DissipateHeat	Takes the current heat, subtracts the heat dissipation rate from it and then calls UpdateHeat.
OverheatAlarm	If the gun overheats, set the overheated text visible on the UI and prevent the weapon from firing until the heat reaches 0 again.
Recoil	Adds a pitch and yaw input to the player's current aim over a short timespan based on the recoil magnitudes.





## 5. Base Projectile

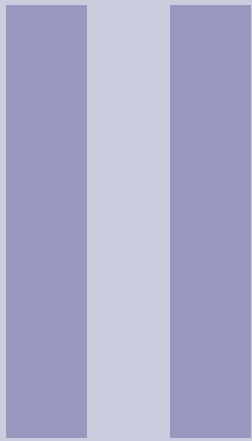
The base projectile contains the particle effects and damage information to transmit to a hit object.

### 5.1 Variables

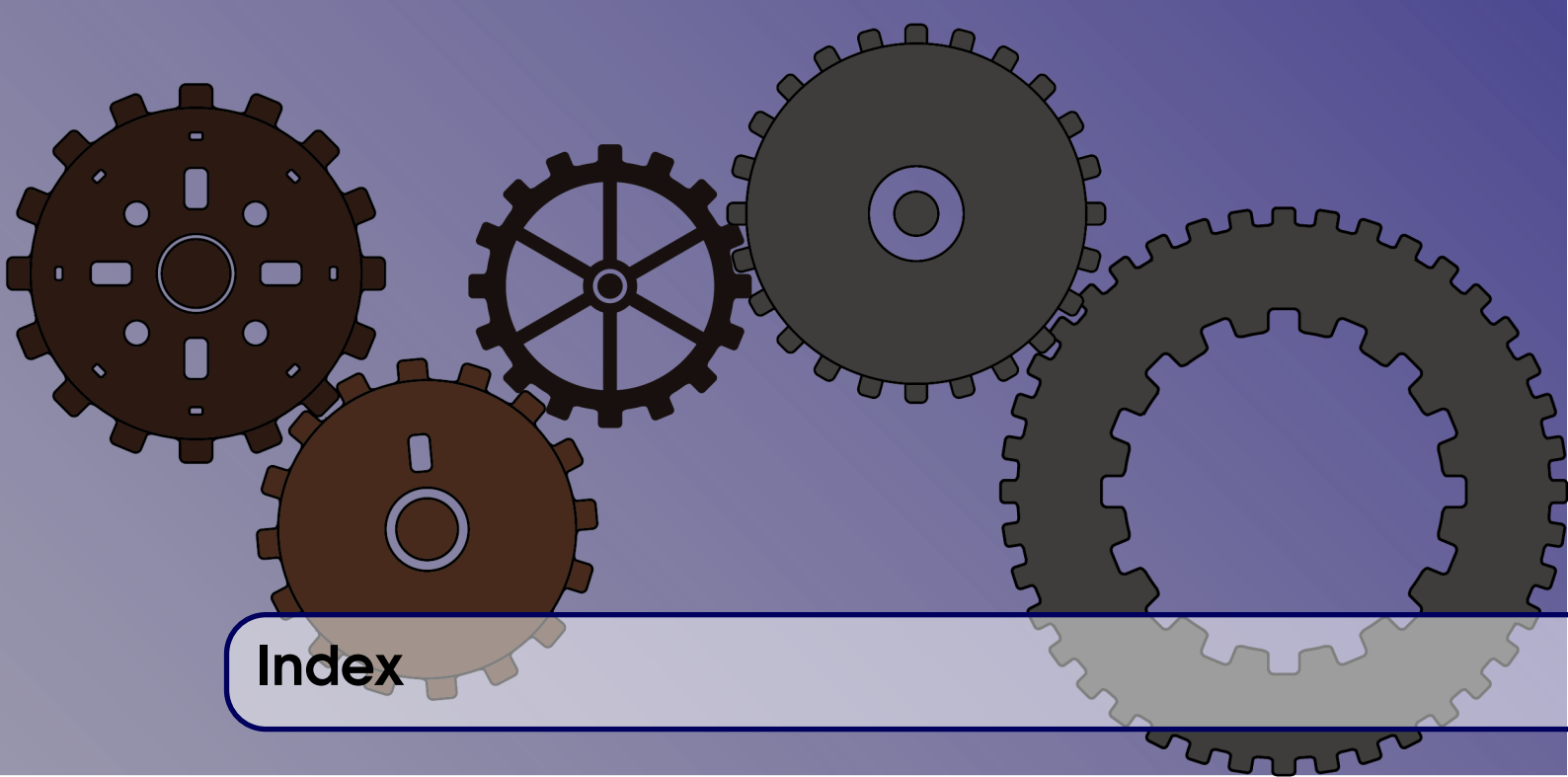
Variable	Type	Description
DamageMin	Int	Minimum damage of the projectile that was passed on from the weapon class.
DamageMax	Int	Maximum damage of the projectile that was passed on from the weapon class.
CritChance	Int	Critical hit chance of the projectile that was passed on from the weapon class.
ImpactForce	Int	Impact force of the projectile that was passed on from the weapon class.
ImpactSound	SoundCue	Sound to play when projectile impacts an object.
ImpactEffect	ParticleSystem	Particle system to spawn on impact.

### 5.2 Functions

Function	Description
InitializeProjectile	Sets the maximum projectile lifespan to 10 seconds when the particle is spawned.
Impact	When the projectile impacts an object, it transfers the damage information to that object, spawns the impact particle system, plays the impact sound and then destroys the projectile.



# Closing Materials



# Index

## B

Base Enemy .....	11
Base Enemy: AI Tree .....	13
Base Enemy: Functions .....	12
Base Enemy: Variables .....	11
Base Projectile .....	17
Base Projectile: Functions .....	17
Base Projectile: Variables .....	17
Base Weapon .....	15
Base Weapon: Functions .....	16
Base Weapon: Variables .....	15

## C

Classes .....	6
---------------	---

## P

Parex Character .....	9
Parex Character: Functions .....	10
Parex Character: Variables .....	9
Parex Controller .....	6
Parex Controller: Functions .....	7
Parex Controller: Variables .....	6