<u>Hollowborn</u>

<u>Dev Report</u>

By Oliver Scott

1.

1 CONTENTS

2	Over	view	2
	2.1	Project Summary	2
	2.2	Goals	2
	2.3	Planning	3
	2.3.1	Deliverables	3
	2.3.2	Gantt Chart	4
3	Rese	arch	5
	3.1	Design Research	5
	3.2	Asset Research	7
	3.3	Methodologies	7
4	Prod	uction	8
5	Evalu	uation1	5
6	Biblio	ography1	7

2 OVERVIEW

2.1 PROJECT SUMMARY

This project focussed on a problem many games suffer from - a mediocre melee combat system. If the core of a game is not its combat, often some combat will still exist within the game but will have been added as more of an afterthought. This leads to the combat feeling unsatisfying and neglected.

Hollowborn aimed to prove that engaging melee combat can be developed with minimal development resources, and more importantly, with no prior experience. Hollowborn is primarily inspired by Dark Souls/Elden Ring, with secondary inspirations from Monster Hunter and Dragons Dogma.

The project acts as a foundation that a bigger game could be built on top of, as well as a demonstration of combat techniques that various games can implement to improve their combat systems.

2.2 GOALS

Systems:

The systems in the game were designed to be robust and modular, making them easily expandable.

Designer "Easy":

The project was structured in a way that makes the designers life easy. Data tables for weapons are an example of how I achieved this.

Satisfying:

The combat focussed on being engaging and satisfying through a variety of combat design and game feel techniques. The controls should feel highly responsive.

2.3 PLANNING

2.3.1 Deliverables

- Executable Game
 - \circ Built from Unreal Engine 5 project.
 - Playable on Windows.
 - Playable with keyboard & mouse and controller.
 - \circ 3D Game.
 - Third-person character controller akin to that of Elden Ring.
 - Single level with enemies to use the combat controller against.
- Satisfying Combat
 - o Screen shake.
 - o Hit-Stop.
 - o Particles.
 - o Sound Effects.
 - o Visual Effects.
 - Smooth animation and blending.
 - o Combos.
- Complex Damage System
 - Resistances component will allow all entities to have their own resistance profile against a variety of damage types and status types.
 - o Damage types
 - Physical
 - Magic
 - Fire
 - Lightning
 - Holy
 - o Status types
 - Bleed
 - Frostbite
- RPG Stats System
 - RPG Stats that scale various player stats like health, stamina and damage.
 - Weapon Infusion System akin to Elden Ring (Different infusion types have different stat scaling).
- Debug Menu
 - Allow advanced testing actions such as teleporting, changing stats, weapon infusion, etc.
 - \circ Available in demo.
 - Allows for rapid testing/iteration.
- Enemy Al

- Realistic Enemy AI that has a variety of attack patterns.
- Uses AI perception in Unreal Engine.
- Chase player, attack or block when within range.

2.3.2 Gantt Chart

These deliverables were broken down into milestones and turned into a Gantt Chart. The tasks were ordered by priority to ensure that if I had over-scoped, the project would likely still lead to a playable demo.

1	Hollowborn		Course: BSc (Hons) Computer Games Desi							
2	Oliver Scott			L						
3		Pro	ject Start:	Mon, 30-S	ep-2024	7				
4		Disp	lay Week:	9						
5		-	- 6							
6	TASK AS	SIGNED TO	ROGRESS	START	END	WORK DAYS				
7	1. Planning and Pre-Production				-					
8	Task 1.1 Finalise Pitch Document		100%	Mon 30/9/24	Mon 30/9/24	1				
9	Task 1.2 Make Gantt Chart		100%	Mon 30/9/24	Tue 1/10/24	2				
10	Task 1.3 Asset Sourcing Research		100%	Thu 3/10/24	Fri 4/10/24	2				
11	Task 1.4 Proposal Document		100%	Mon 7/10/24	Thu 10/10/24	4				
12	Task 1.5 Ethics Form		100%	Mon 14/10/24	Thu 17/10/24	4				
13	2. Basic Combat Controller				-	-				
14	Task 2.1 Locomotion (Motion Matching)		100%	Mon 21/10/24	Mon 21/10/24	1				
15	Task 2.2 Initial HUD		100%	Wed 23/10/24	Wed 23/10/24	1				
16	Task 2.3 Action Resources		100%	Wed 23/10/24	Wed 23/10/24	1				
17	Task 2.4 Dodging		100%	Wed 23/10/24	Wed 23/10/24	1				
18	Task 2.5 Attacking (Light & Heavy)		100%	Thu 24/10/24	Thu 24/10/24	1				
19	Task 2.6 Test Dummy Creation		Teeko	2 Hit Impulse	E-: 25/40/04	- 1	10004	Thu 5/10/04	Thu 5/10/04	1
20	Task 2.7 Hit Detection	55	Task9	A Hit Popetions (11)	t Knocked Deve	<u>ا</u>	100%	Thu 5/12/24	Fri 6/12/24	1
21	Task 2.8 Damage Pop-Ups	50	10 Part	A HICKEACTIONS (HI	I, KHOCKEG-DOWN	1	100%	1 nu 3/12/24	FII 0/12/24	2
22	Task 2.8 Lock On System	50	Tack 1	0.1 LOS Detection			100%	Mon 9/12/24	- Mon 9/12/24	1
23	Task 2.8 Input Buffering	50	Task1	0.1 LOS Delection	h Detection		100%	Mon 9/12/24	Mon 9/12/24	1
24	3. Debug Menu	59	Task 1	0.2 Damage & rouc	aver		100%	Mon 9/12/24	Mon 9/12/24	1
25	Task 3.1 Create Debug Window	61	Task 1	0.4 Enemy Attack Pl	aver & Plaver Blo	cking	100%	Wed 11/12/24	Thu 12/12/24	2
20	A Resistance/Damage Calculation	62	Task1	0.5 Animations & P	olish	citing	100%	Thu 12/12/24	Thu 12/12/24	1
27	4. Resistance/Damage Calculation	63	11. Play	test	51511		10070	110 12/ 12/24	-	-
20	Task 4.2 Make Resistance Profile Struct	64	Task 1	1.1 Make Build			100%	Wed 18/12/24	Wed 18/12/24	1
30	Task 4.3 Make Weapon Profile Struct	65	Task 1	1.2 Make Online Fo	rm		100%	Wed 18/12/24	Wed 18/12/24	1
30	Task 4.4 Implement Damage Calculations	66	Task 1	1.3 Run playtest			100%	Wed 18/12/24	Mon 6/1/25	4
32	Task4.5 Modularize Weapons&Attacks	67	Task 1	1.4 Respond to feed	lback		100%	Mon 6/1/25	Fri 10/1/25	5
33	Task 4.6 Test System on Dummy	68	12. Adva	nced Polish					-	
34	5. Basic Polish	69	Task 1	2.1 Add Hit-Stop to	Attacks		100%	Mon 13/1/25	Mon 13/1/25	1
35	Task 5.1 Initial Combat SFX	70	Task 1	2.2 Crunchier & Exp	anded SFX		100%	Mon 13/1/25	Tue 14/1/25	2
36	Task 5.2 Initial Screenshake	71	Task 1	2.3 Animation Refir	ements		100%	Tue 14/1/25	Thu 16/1/25	3
37	Task 5.3 Initial Particles	72	14. Pois	e System					-	
38	Task 5.4 Animation Clean Up	73	Task 1	4.1 Research Poise			100%	Thu 16/1/25	Thu 16/1/25	1
39	6. Weapon Infusion	74	Task 1	4.2 Plan Poise Impl	ementation		100%	Fri 17/1/25	Fri 17/1/25	1
40	Task 7.1 Infuse via Debug Menu	75	Task 1	4.3 Implement Pois	e		100%	Mon 20/1/25	Wed 22/1/25	3
41	Task 7.2 Change Damage Type	76	15. Con	troller Support					-	
42	7. RPG Scaling Stats	77	Task 1	5.1 Design controls	scheme		100%	Wed 22/1/25	Wed 22/1/25	1
43	Task 7.1 Add in all stats	78	Task 1	o.∠ Port controls to	gamepad		100%	Wed 22/1/25	Thu 23/1/25	2
44	lask 7.2 Adjust via Debug Menu	79	Took 1	6 1 Design new Fran	my h/h0		10004	Mon 27/1/25	- Mon 27/1/25	1
45	Task 7.3 Make stats scale variables	80	Task 1	6.2 Implement pow	Enemytype		100%	Tup 28/1/25	Thu 30/1/25	3
46	Task 7.4 Scaling for weapon infusion	01	Tack 1	6.3 Test and Refine	new Energy type		100%	Fri 31/1/25	Mon 3/2/25	2
4/	A Compatible Contraction Dars based on stats	92	Task 1	6.4 Polish Enemies	F.g. Strafing		100%	Mon 3/2/25	Tue 4/2/25	2
48	o. Compat Design Research	84	17. Fina	l Plavtest	a.g. ou uning		20070	110110/2/20	-	-
49	Task 8.2 Design Combat	85	Task 1	7.1 Plan Final Plavt	est		100%	Tue 4/2/25	Tue 4/2/25	1
51	Task 8.3 Undate Forum	86	Task 1	7.2 Make Build			100%	Tue 4/2/25	Tue 4/2/25	1
52	9. Advanced Combat Controller	87	Task 1	7.2 Make Online Fo	rm		100%	Tue 4/2/25	Tue 4/2/25	1
53	Task 9.1 Motion Warping	88	Task 1	7.3 Run playtest			100%	Tue 4/2/25	Mon 10/2/25	5
54	Task 9.2 Combos	89	Task 1	7.4 Respond to Fee	dback		100%	10/02/2025	Wed 12/2/25	3
		90	18. Arte	fact Submission					-	
		91	Task 1	8.1 Final Refinemer	nts		100%	Thu 13/2/25	Thu 13/2/25	1
		92	Task 1	8.2 Get Demo Read	/		100%	Fri 14/2/25	Fri 14/2/25	1
		93	Task 1	8.3 Upload and Sub	mit		0%	Fri 14/2/25	Fri 14/2/25	1
		94	19. Prep	are for Artefact Pre	esentation				-	
		95	Task 1	9.1 Making of Docu	ment		0%	Mon 17/2/25	Mon 17/2/25	1
		96	Task 1	9.2 Prepare Present	ation		0%	Tue 18/2/25	Tue 18/2/25	1
		97	Task 1	9.3 Practice			0%	Wed 19/2/25	Wed 19/2/25	1

3 RESEARCH

3.1 DESIGN RESEARCH

Coming from minimal combat experience, I decided to first research combat techniques that professional developers often use.

This general combat design was covered by a variety of videos linked in the <u>Bibliography</u> section of this report. I gathered the following information from these:

- \circ Think about how the player should feel. Like a god, an assassin?
- Hit states sell impact, e.g. knockdown, fly-back, etc.
- Consider stun locking.
- o Transition directly into a hit state rather than blending into it.
- \circ $\;$ Each enemy should challenge the player in a variety of different way.
- If the player's attacks have root motion, make sure the enemies are translated backwards to improve combat flow.
- Use hit stop and camera shake.
 - Ensure the hit stop only affects the hit actor and the instigator rather than the whole game's time dilation.
 - \circ $\;$ Add a hit stop blend in and out so it is smoother and less abrupt.
- Ensure the camera is framing the combat correctly.
- Make the most effective way to play your game the most fun.
- Encourage a varied experience.
 - \circ $\;$ Attacks should not be effective against everything.
 - Do not restrict or discourage an action, rather encourage other actions.
- \circ $\;$ $\;$ Increase player reaction time by adding or removing the number of enemy attacks.
- Use input buffering.
- o Use motion warping and generous hit boxes.
- Use motion values for combos.
- \circ $\;$ Sound design is critical to the project.

I applied most of these techniques to the game's combat system, and it is evident that the combat's feel and flow would be much worse without this research. On top of this I covered a lot of research into Elden Ring and how the systems are constructed and interact with one another. This covered the more technical side of the research and was primarily done via the <u>Wiki</u> (*Elden Ring Wiki*, 2025) and by playing the game.

Technical research focussed on areas specific to Elden Ring's design and systems such as stat scaling, damage types, status types, weapon infusion, poise, and damage calculations.

This technical research was then collated onto a draw.io page where I made design diagrams and structural planning, here are some examples:

Stats Damage Scaling: Source: https://www.youtube.com/watch?v=Dt8K1DL1HDo Two handing gives 50% more strength, so always two hand strength weapons Stats cannot scale above 99 Weapons have a scaling value for each stat that will fit into a category S > 1.75 S > 1.75 A 1.4 - 1.75 A 1.4 - 1.75 B0.9 - 1.3999B0.9 - 1.3999C0.6 - 0.8999C0.6 - 0.8999 D 0.25 - 0.5999 D 0.25 - 0.5999 E < 0.25 E < 0.25

Scaling Value * Base Damage (Saturation/100) Note: Saturation is the scaling you get from your stats (Cumulative Value on a graph)



S_WeaponProfile	E_DamageType	S_DamageType	E_AilmentType	S_AilmentType
Physical [Physical, Value] : S_DamageType	Physical	Damage Type : E_DamageType	Bleed	Damage Type : E_AilmentType
Physical Type : E_DamageType	Strike	Value : float	Frostbite	Value : float
Strike [Strike, Value] : S_DamageType	Slash		Poison	
Slash [Slash, Value] : S_DamageType	Pierce		DeathBlight	
Pierce [Pierce, Value] : S_DamageType	Magic			
Magic [Magic, Value] : S_DamageType	Fire			
Fire [Fire, Value] : S_DamageType	Lightning			
Lightning [Lightning, Value] : S_DamageType	Holy			
Holy [Holy, Value] : S_DamageType				
HasAilment? : Boolean				
AilmentType : S_AilmentType				

3.2 ASSET RESEARCH

Asset research was important for Hollowborn because the project heavily relied on sourcing assets particularly for animations, sounds, and visual effects. I would not have been able to make all of these by myself within the project's timeframe, hence I carried out research on where to find assets. My primary locations for assets were Humble Bundle (*Software*, no date), FAB (*Fab*, no date), and Mixamo (*Mixamo*, no date). An asset list will be covered in the <u>Bibliography</u> section.

I did implement the new motion matching animation techniques from Epic's sample project (Game Animation Sample Project in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community, no date). This was something I discovered when researching what new features Unreal Engine 5 was offering. The animation is so smooth and was a good starting point for my project.

On the other hand, I did research Epic's Gameplay and Ability System (Gameplay Ability System for Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community, no date). This would have been an interesting approach; however, I decided against it quite early on as I initially wanted to tailor this project towards technical design. The RPG stats and resistances were something I wanted to tackle myself and test my capabilities to design great systems. This was a decision made to improve my portfolio.

3.3 METHODOLOGIES

During the development cycle, whether I was researching or responding to feedback, I had a methodology that I followed. This ensured that my systems were constantly adapting to new research and feedback:

- Identify topic.
- Search for a design video to explain the theory behind it, why it works, when it should be used, and what alternatives there are.
- Identify and Analyse how this topic is explored in games.
- Design how it will be used in Hollowborn.
- Research how it can/should be implemented.
- Implement it.
- Iterate if testing data indicates that it may need this.

Playtesting was my primary method to evaluate the success of the project. There were two playtesting sessions. The first covered the game's controls and responsiveness, and the second was the whole demo itself to see if I had achieved my goals. Play tester's feedback allowed me to keep the project focussed and enabled progressive iteration. Satisfying combat is a subjective topic, so to account for this, I ensured that any play testers fitted my target audience of 'Souls-like enjoyers'.

Following this playtesting, they had to fill out a <u>research questionnaire</u> which gave me feedback into the various areas I was testing for. From these scores and feedback, I was then able to evaluate whether my project was currently successful or not, and what changes needed to be made to make it better. The final scores demonstrated whether I applied my research well enough to achieve my goal of a satisfying melee combat system; an ideal score was 8 or above.

4 PRODUCTION

Hollowborn is a large project with many systems within, so this section will focus on some of the key systems developed, why they were added, and how they compare to industry.

Damage Resistance System:

This system was inspired by Elden Ring as it allows for each enemy to be strong or weak against varying damage types. This makes each enemy encounter more dynamic and allows the player to adjust their build if they are struggling against a certain enemy or boss.

This means that both the player and all enemies need the ability to have this damage resistance. Understanding this, I made this damage resistance system into a component so it can easily be added to any entity in the game.

I carried out research into these <u>Damage Types</u> (*Damage Types*, 2025) and also <u>Status Types</u> (*Status Effects*, 2024), and I added the following types:

Damage Types:

- Physical (Standard, Strike, Slash, Pierce)
- Magic
- Fire
- Lightning
- Holy

Status Types:

- Bleed (Victim suffers instant damage based on their max HP)
- Frostbite (Victim suffers instant damage based on their max HP, has slowed stamina recovery, and has reduced damage negation for 30 seconds)

Here I knew that I would need a resistance profile structure that allows designers to input the corresponding resistances to each damage type. See Figure 1.

🔻 De	fault Value				
🖝 Re	sistance Profile				
•	Defence	8 Map elements	Ð	Û	
	Holy	~	100.0		
	Lightning	~	100.0		
	Fire	×	100.0		
	Magic	×	100.0		
	PPierce	~	100.0		
	PStrike	~	100.0		
	PSlash	~	100.0		
	PStandard	~	100.0		
►	Negation		8 Map elements	Ð	Û
	Robustness		225.0		
	KnockdownNegation		0.0		
	KnockbackNegation		0.0		
	Poise		50.0		

Figure 1 Resistance Profile

Designers can now easily set the defence and negation against each damage type for each enemy as well as other variables seen at the bottom. But how is damage then calculated from this?

The damage also comes in as a structure with a motion value and then the damage is calculated separately for each damage type and then added together at the end.

▶ DamageProfile E Damage Ty; ∨ ≣ ∨ ∎ Float ∨	
	ū
StatusProfile EStatus Effec EFloat	Ū

Figure 2 Damage Profile

The damage calculation was a lot more complex that I had initially anticipated. I expected some basic formula, but Elden Ring uses damage brackets to concatenate formulas to create a damage graph.



I implemented this exact system as the curve ensures that damage never is zero and has fall off. This information was found <u>here</u> (*Calculating Damage*, 2024).

Elden Ring uses this system better than Hollowborn due to the breadth of the game. Elden Ring has hundreds of enemies, whereas Hollowborn has two. It is much harder for this system to be showcased in Hollowborn as I am only using undead skeleton enemies (weak to holy damage), whereas if I had a new nature enemy, I could for example make them weak to fire damage. This would distinctly show this system in action and players would learn what is effective against each enemy.

Furthermore, Elden Ring implements additional features that Hollowborn does not, such as "HitBox Damage" where enemies may have weak spots and take more/less damage. They also implement critical damage where enemies can take lots of damage from a special animation like a riposte or a backstab. Certain weapons like daggers deal additional damage here. Elden Ring also has additional status effects such as poison, scarlet rot, madness, and death blight.

RPG Stats & Weapon Infusion System:

Many RPG's including Elden Ring implement a stat system as a form of progression for the player. I have used <u>Elden Ring's implementation</u> (*Stats*, 2024) of this once again. The stats in Hollowborn are:

- Vigor (Scales Health)
- Mind (Scales Mana)
- Endurance (Scales Stamina)
- Strength (Scales Physical and Fire Damage)
- Dexterity (Scales Physical and Lightening Damage)
- Intelligence (Scales Magic Damage)
- Faith (Scales Lightning and Holy Damage)
- Arcane (Scales Status Build Up)

Creating dynamic health bars was more complex than I initially thought as I hadn't considered that the widgets would need to be directly on a canvas panel so that their size can be adjusted. I had to rework the resource bars to be placed onto the canvas panel and then adjust the size via this code:



The RPG system for Hollowborn is like that of Elden Ring, allowing progression and stat scaling. However, it is a lot more simplified than Elden Ring. In Elden Ring each stat affects more things, e.g. Vigor affects fire resistance, immunity, and HP, but more importantly each stat has a falloff graph.



*Figure 3 Vigor Scaling Graph (*A Comprehensive Study of Elden Ring Stats: Why the PvP Meta Should NOT be 125 : r/Eldenring, *no date)*

Elden Ring uses stat curves to ensure there is fall off meaning that as you get towards 60 of a stat, the return on investment gets less. Hollowborn does use fall off, for example I use this formula to calculate health based on the Vigor stat:

 $y = 520 \log (x+5)$, where log is of base 100.

This can be seen in Figure 4 below. This graph makes sure that at 0 Vigor, the player's health is at 180 and then at 100 Vigor it is approximately 525. But it is not quite as customisable as that in Elden Ring where they seem to use the brackets technique again.



INFUSE							
STANDAR	D	HEAVY					
KEEN		MAGIC					
FIRE		LIGHTNING					
SACRED		BLOOD					
FROST		STATS					
Fire Iron_Longsword							
ATTACK POWI	ER	Phys: Standard					
PHYSICAL	125 + 2	23					
Magic	0						
Fire	125 + 1	8					
Lightning	0						
Holy	0						
Stat Scaling							
Str A	Dex	D					
Int -	Fai						
ARC -							
STATUS BUILE	DUP						
Str:10 Dex:1	0 Int:	10 Fai:10 Arc:10					
	BAC	CK					

Figure 4 Vigor Scaling Fall Off Graph

Figure 5 Weapon Infusion UI

As mentioned at the start, stats like strength affect the attack power of physical weapons. This leads onto the weapon scaling and the weapon infusion system. The player can use the Debug Menu to change their weapon's infusion type. See Figure 5. Fire damage scales with strength and so the stat scaling for this weapon I have set to A and dexterity to D. These are not flat values as I had anticipated. The weapon information was gathered <u>here.</u> (*Weapon Scaling* | *Elden Ring Wiki* | *Fandom*, no date)

• E: < 0.25
• D: 0.25 - 0.5999
• C: 0.6 - 0.8999
• B: 0.9 - 1.3999
• A: 1.4 - 1.75
• S: > 1.75
The formula for calculating a weapon's scaling damage is:
$Scalingvalue * Basedamage\left(rac{statscaling}{100} ight)$

The letters represent the scaling value brackets, and the stat scaling represents what the stats value is like strength. This is how a weapon's scaling damage is calculated; hence it is represented as base damage + scaling damage. The scaling damage is low in Figure 5 because the player only has 10 of each stat, where there is a maximum of 100.

Each weapon has its own row in the weapons data table. This ensures that weapons are easily expandable in the game and so designers can easily adjust them.

		Row Name	WeaponName	BaseSocket	TipSocket	WeaponModel	TrailParticles
	1	Iron_Longsword	Iron_Longsword	Base	Тір	/Script/Engine.StaticMesh'/Game/Characters/GKnight/Meshes/Weap	on/ /Script/Niagara.NiagaraSystem'/Game/VFX/
1	2	Enemy_Longswore	Enemy_Longsword	Base	Тір	/Script/Engine.StaticMesh'/Game/Animations/Imported/A_Longswor	Ar /Script/Niagara.NiagaraSystem'/Game/VFX/
		Enemy_Axe	Enemy_Axe	Base	Тір	/Script/Engine.StaticMesh'/Game/SKnight_modular/AxeMesh.AxeMes	h' /Script/Niagara.NiagaraSystem'/Game/VFX/
		Enemy_Boss_Axe	Enemy_Boss_Axe	Base	Тір	/Script/Engine.StaticMesh'/Game/SKnight_modular/AxeMesh.AxeMesh	ih' /Script/Niagara.NiagaraSystem'/Game/VFX/
		- 12					
	R	ow Editor ×					
Ľ	ron	_Longsword	ب ب				
-	r Iro	n_Longsword					
	We	aponName					Iron_Longsword
	Ba	seSocket					Base
	Tij	oSocket					Тір
	We	eaponModel					SM_WP_GothicKnight_Sword V
		ailParticles					NS_Trail_Longsword ✓ € ₯
	Ble	ockNegation					0.8
	Gu	ardBoost					0.25
▶	Lig	ghtAttackCombo					4 Array elements 🕣 🛱
▶		ghtSprintAttack					
▶	Не	avyAttackCombo					2 Array elements 🕣 🛱
▶	Не	avySprintAttack					

Each weapon has its own model, trail, names, sockets, attacks, values, etc. The relevant part to the weapons damages though is here, the infusion profile:

InfusionType	Standard V
✓ InfusionProfile	
▼ InfusionScalingProfiles	9 Array elements \ominus 🛱
▶ Index [0]	4 members 🗸
▶ Index [1]	4 members 🗸 🗸
▶ Index [2]	4 members 🗸 🗸
▶ Index [3]	4 members 🗸 🗸
▶ Index [4]	4 members 🗸 🗸
▼ Index [5]	4 members 🗸 🗸
InfusionType	Lightning V
BaseDamage	140
	2 Map elements \ominus 🛱
Dexterity V	1.0 ~
Faith ×	1.0 ~
StatusValues	0 Map elements 🕀 🛱
▶ Index [6]	4 members 🗸 🗸
▶ Index [7]	4 members 🗸 🗸
▶ Index [8]	4 members 🗸
CanBeInfused?	

In here the weapon's default infusion can be set (Infusion Type), and then the infusion profile can be filled out. For example, the lightning infusion on this longsword gives it a base damage of 140, and a B scaling in both faith and dexterity and then the formula mentioned above is applied to calculate the scaling damage for it based on the player's stats.

My weapon infusion system is almost identical to that of Elden Ring, allowing the player to change the damage type of their weapon and what stats it uses to scale. This is a fantastic system as it allows players to customise weapons to fit their current build. The only thing missing are the weapon arts, but that was not within the project's scope.

Combat, Animation & Game Feel:

Elden Ring's combat is intense and highly skilful. Trying to replicate this was the biggest challenge of the project, coming from no prior experience. My <u>research</u> played a big part in my ability to create this.

Playing Elden Ring and analysing it myself was crucial here as well as there were many techniques I discovered that I was not aware of before. One example of this is the ability to rotate mid-attack. I achieved this via an animation notify state:

ANS_RotateCharacter

During this window, if the player is not locked on, they are able to change their rotation towards their input direction. This improves game feel and responsiveness. Elden Ring does this slightly better due to the longer telegraphing on attacks. The longsword animation pack I used has attacks with minimal wind up which makes this window a bit shorter and less noticeable.

Custom animation notifies and notify states were used a lot in Hollowborn as they improve modularity.



For example, this animation notify state opens and closes the input window for input buffering. Input buffering is something many combat focussed games use to improve game feel and responsiveness. I have implemented this in a previous project of mine, and it was used in Hollowborn for all combat actions like dodging, attacking, healing, etc.

The primary challenge with Hollowborn's combat was getting the responsiveness to feel correct. Players reported during testing that they felt that the attacks were too "sticky" and unforgiving. One argument is that the combat design is meant to be more deliberate and punishing, however after paying closer attention to how Elden Ring deals with this situation I discovered the tricks that they use.

Elden Ring allows players to quickly blend from one action to another, cancelling the blend out animation of another with a new one. For example, attacking then dodging doesn't require the whole recovery section of the attack animation to finish before allowing the player to dodge. Adjusting the blend out times and durations, as well as input windows were the adjustments I made to fix this issue. These issues were prominent even in the final playtest but have now been fixed.

An advanced animation technique that I implemented into Hollowborn is Motion Matching. This gave my project instant, professional locomotion via Epic's sample project (*Game Animation Sample Project in Unreal Engine* | *Unreal Engine* 5.5 *Documentation* | *Epic Developer Community*, no date). The issue with this though was that it did not look realistic while holding a two-handed sword in one hand. To get around this, I made my own custom locomotion when the sword is drawn via a blend space. This looks more like Elden Ring as it uses a similar system, however the motion matching would've put Hollowborn above Elden Ring in terms of realistic locomotion. This locomotion is still available when the weapon is sheathed though.

Lastly, a key part of attacking in Elden Ring is how it translates the player forward via root motion. All attacks used had root motion, and if I wanted any further root motion to the animation, I used motion warping. Motion warping artificially increased the displacement of each attack allowing the combat to flow better with enemies also being translated backwards on hit (knockback/hit impulse). Enemies use these techniques too to track/target the player with their attacks. I added motion warping via a component and animation notifies. Without motion warping and root motion, the combat would have felt static and unnatural.

<u>Enemy Al:</u>

The enemy AI took a lot of inspiration from Dark Souls with long telegraphing on attacks giving the player time to react and dodge. I carefully sourced animations for this reason ensuring that each attack was designed to test the player.

One important technique I have used that may have gone unnoticed is making the enemies track the players position via their head and hip rotations.



These were the nodes I used to achieve this in the enemy's animation blueprints. There is a clamp to this, so the enemies don't rotate their heads all the way around behind them. This greatly improves how realistic/natural the enemies look and move.

My AI was quite basic overall compared to Elden Ring. It involved all the basic actions like moving, attacking, blocking, hit reactions, etc. However, Elden Ring's enemy AI involves strafing movement, a token system (for group AI), and advanced pathing. Ideally, I would like to use Unreal Engine's Environmental Query System to improve the AI's pathing as well. These additional features were out of scope for the project though.

5 EVALUATION

As mentioned in the <u>Research Methodologies</u> section, the primary evaluation of the success of Hollowborn comes down to the playtesting feedback. For the final playtest, Hollowborn achieved an 8.2/10 for satisfaction, and a 7.8/10 overall.

8. On a scale of 1-10, 1 being not at all, 10 being very satisfying, did combat impactful and satisfying?	More details	
Level 10 Level 9 1 Level 8 Average Rating Level 5 Level 5	4	
7.80 Average Rating ★★★★★★★★★☆☆☆☆	Level 10 Level 9 Level 8 Level 7 Level 6 Level 5 Level 4 Level 3 Level 2 Level 1	2 2

An ideal score was 8+ so this highlights to me that I have successfully created a satisfying melee combat system and a good demo for the project. Many things have been improved since the final playtest, so to extrapolate I would expect a rating of 8.8+ for satisfaction and 8.5+ for the demo with the latest changes.

This data indicates that Hollowborn successfully meets the deliverables I had set out for and even reached some of the expanded deliverables added during the mid-point review. It's important to note that I managed to get two industry game designers who are souls-like fans to participate in this feedback, so these aren't just results from my friends. These results are credible and give me confidence that Hollowborn is my best work yet.

The key takeaways from this project are:

- Plan rigorously and create a Timeline/Gantt Chart.
- Research in depth and use multiple sources.
- Be ruthless with scope, MoSCoW.
- Playtest with your target audience early.
- Implementing animation techniques, e.g. Montages, Blendspaces, Motion Matching, Motion Warping, Bone Look At, etc.
- Modular and designer friendly code is a MUST in big projects, even if it's an independent project.
- Create a Debug Menu for efficient testing.

The project will continue development until my submission of it to GradEx on June 6th 2025. The features I aim to add during this time are as follows:

- Lock-on system rework
 - Tester feedback pointed out that it should always lock on to a target if there is one within range. The current lock-on system does a long sphere trace in the direction of the camera, so it can be hard to line up sometimes. A rework to this was out of scope, hence has been left until after submission. This would improve the quality of life for Hollowborn.
- Enemy token system
 - Currently if the player is fighting more than one enemy at once, it feels unfair because both enemies can attack simultaneously. The token system is an AI technique that prevents this from happening, meaning my level will be able to include multiple enemies per encounter hence providing a new challenge for the player.
- Chaos destruction
 - The benches in the boss room can block the player's movement. Ideally the player should be able to break them when hitting them or rolling into them like in Elden Ring. The enemies should be able to do the same, which would add to the power of the enemies, making the game more satisfying.
- Add an Official Boss
 - The boss right now is just a scaled-up enemy that testers did not find interesting. I want to add a new boss with different attacks and animations that I specifically design for. There should be boss music, a boss health bar, and potentially 2 phases.
- Improve the camera

- Testers reported that the camera often would get too close to the player in tight spaces making it impossible to see. I will need to research some common industry techniques to rework Hollowborn's camera so the player is not fighting with the camera and can just enjoy the combat.
- Expanded combat.
 - If within scope, Hollowborn would benefit from some satisfying executions/critical hits like backstabs or ripostes. Parrying is something that testers suggested could be added too.
- Add a main menu.

Hollowborn was initially meant to be a portfolio piece for a technical design role, however I now wish to pursue combat design which is the reasoning behind the plans I have for the project.

6 **BIBLIOGRAPHY**

Combat Design Research:

- o Combat Design Blog (says, 2024)
- GMTK: What makes a great combat system? (*What Makes a Good Combat System*?, 2018)
- Combat Design Devlogs (*Creating a satisfying combat system for my game* | *Withersworn Devlog*, 2024)
- Killer Combat on an Indie Budget (*Dreamscaper: Killer Combat on an Indie Budget*, no date)
- Combat In God of War (Evolving Combat in 'God of War' for a New Perspective -YouTube, no date)
- Combat In Ghost of Tsushima (*Master of the Katana: Melee Combat in 'Ghost of Tsushima'*, 2022)
- Motion Warping (How to Use Motion Warping in your Animations Easily in Unreal Engine 5 - YouTube, no date)
- Motion Values (Motion Values and how to find them (Monster Hunter), 2016)

Game Feel Research:

- Juice it or Lose it (Juice it or lose it a talk by Martin Jonasson & Petri Purho, 2012)
- The Art of Screenshake (Jan Willem Nijman Vlambeer 'The art of screenshake' at INDIGO Classes 2013, 2013)
- Sound Design
 - (MetaSounds for Beginners | GameSoundCon 2022 | Unreal Engine, 2022)
 - (Metasound Basics: Part 1 A Guide to Next Generation Audio in the Unreal Engine 5, 2023)

Souls-Like Playlist: (Souls-Like Melee Combat Tutorial Showcase [UE5], 2023)

Souls-Like Udemy Course: (Unreal Engine 5: Soulslike Melee Combat System, no date)

Assets:

 (Modular Dungeon Kit (Medieval Dungeon, Dungeon, Dungeons, Crypt, Tomb), no date) (Gothic Knight, no date) (Slash and Hit VFX (with SFX), no date) (RPG FX Starter Pack, no date) (Longsword Animset Pro, no date) (Skeleton Knight Modular, no date) (Skeleton Knight Modular, no date) (Sword Combat Powerful Whooshes SFX, no date) (ANIMATION | Sword Combat Asset Pack | Synty[™] Official Store, no date) (Cinzel - comments | dafont.com, no date)

Official Bibliography:

A Comprehensive Study of Elden Ring Stats: Why the PvP Meta Should NOT be 125 : r/Eldenring (no date). Available at:

https://www.reddit.com/r/Eldenring/comments/tin3ds/a_comprehensive_study_of_elden_ring _stats_why_the/?rdt=37227 (Accessed: 20 February 2025).

ANIMATION | Sword Combat Asset Pack | SyntyTM Official Store (no date) Synty Store. Available at: https://syntystore.com/en-gb/products/animation-sword-combat (Accessed: 20 February 2025).

Calculating Damage (2024) *Elden Ring Wiki*. Available at: https://eldenring.wiki.fextralife.com/Calculating+Damage (Accessed: 20 February 2025).

Cinzel - comments | *dafont.com* (no date). Available at: https://www.dafont.com/font-comment.php?file=cinzel (Accessed: 20 February 2025).

Creating a satisfying combat system for my game | *Withersworn Devlog* (2024). Available at: https://www.youtube.com/watch?v=-Ts9xF9Owe0 (Accessed: 20 February 2025).

Damage Types (2025) *Elden Ring Wiki*. Available at: https://eldenring.wiki.fextralife.com/Damage+Types (Accessed: 20 February 2025).

Dreamscaper: Killer Combat on an Indie Budget (no date). Available at: https://www.youtube.com/watch?v=30mb5exWpd4 (Accessed: 20 February 2025).

Elden Ring Wiki (2025) *Elden Ring Wiki*. Available at: https://eldenring.wiki.fextralife.com/Elden+Ring+Wiki (Accessed: 20 February 2025).

Evolving Combat in 'God of War' for a New Perspective - YouTube (no date). Available at: https://www.youtube.com/watch?v=hE5tWF-Ou2k (Accessed: 20 February 2025).

Fab (no date). Available at: https://www.fab.com/ (Accessed: 20 February 2025).

Game Animation Sample Project in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community (no date) Epic Games Developer. Available at: https://dev.epicgames.com/documentation/en-us/unreal-engine/game-animation-sample-project-in-unreal-engine (Accessed: 20 February 2025).

Gameplay Ability System for Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community (no date) Epic Games Developer. Available at:

https://dev.epicgames.com/documentation/en-us/unreal-engine/gameplay-ability-system-forunreal-engine (Accessed: 20 February 2025).

Gothic Knight (no date) Fab.com. Available at: https://www.fab.com/listings/0641220a-cff6-4833-aaab-52eb1e4b3ec5 (Accessed: 20 February 2025).

How to Use Motion Warping in your Animations Easily in Unreal Engine 5 - YouTube (no date). Available at: https://www.youtube.com/watch?v=kLgMogzw_uE (Accessed: 20 February 2025).

Jan Willem Nijman - Vlambeer - 'The art of screenshake' at INDIGO Classes 2013 (2013). Available at: https://www.youtube.com/watch?v=AJdEqssNZ-U (Accessed: 20 February 2025).

Juice it or lose it - a talk by Martin Jonasson & Petri Purho (2012). Available at: https://www.youtube.com/watch?v=Fy0aCDmgnxg (Accessed: 20 February 2025).

Longsword Animset Pro (no date) *Fab.com*. Available at: https://www.fab.com/listings/ca8abf87-b0ee-49aa-a477-1e467626b5ce (Accessed: 20 February 2025).

Master of the Katana: Melee Combat in 'Ghost of Tsushima' (2022). Available at: https://www.youtube.com/watch?v=1ih5BxnJu2l (Accessed: 20 February 2025).

Metasound Basics: Part 1 - A Guide to Next Generation Audio in the Unreal Engine 5 (2023). Available at: https://www.youtube.com/watch?v=uckpxH42jQ0 (Accessed: 20 February 2025).

MetaSounds for Beginners | *GameSoundCon 2022* | *Unreal Engine* (2022). Available at: https://www.youtube.com/watch?v=l6ew3hTDa8g (Accessed: 20 February 2025).

Mixamo (no date). Available at: https://www.mixamo.com/#/ (Accessed: 20 February 2025).

Modular Dungeon Kit (Medieval Dungeon, Dungeon, Dungeons, Crypt, Tomb) (no date) Fab.com. Available at: https://www.fab.com/listings/4cc9a46e-6870-4e44-8aeb-4ffb61e76b4c (Accessed: 20 February 2025).

Motion Values and how to find them (Monster Hunter) (2016). Available at: https://www.youtube.com/watch?v=126eFmejJko (Accessed: 20 February 2025).

says, W.G. (2024) 'Combat Design, Mechanics and Systems for Satisfying Game Combat', 5 July. Available at: https://gamedesignskills.com/game-design/combat-design/ (Accessed: 20 February 2025).

Skeleton Knight Modular (no date) *Fab.com*. Available at: https://www.fab.com/listings/fc3a309a-a3eb-46de-bebe-dcb40dc31e48 (Accessed: 20 February 2025).

Slash and Hit VFX (with SFX) (no date) Fab.com. Available at: https://www.fab.com/listings/f6bf529f-6db4-43a5-a5f1-2ff5b8d9a0f4 (Accessed: 20 February 2025).

Software (no date) *Humble Bundle*. Available at: https://www.humblebundle.com/software (Accessed: 20 February 2025).

Souls-Like Melee Combat Tutorial Showcase [UE5] (2023). Available at: https://www.youtube.com/watch?v=9J0NrZJV2H8 (Accessed: 20 February 2025).

Stats (2024) *Elden Ring Wiki*. Available at: https://eldenring.wiki.fextralife.com/Stats (Accessed: 20 February 2025).

Status Effects (2024) *Elden Ring Wiki*. Available at: https://eldenring.wiki.fextralife.com/Status+Effects (Accessed: 20 February 2025).

Sword Combat Powerful Whooshes SFX (no date) Fab.com. Available at: https://www.fab.com/listings/b1516025-50a1-43e0-98b0-380c1903ba37 (Accessed: 20 February 2025).

Unreal Engine 5: Soulslike Melee Combat System (no date) *Udemy*. Available at: https://www.udemy.com/course/unreal-engine-5-soulslike-combat/ (Accessed: 20 February 2025).

Weapon Scaling | *Elden Ring Wiki* | *Fandom* (no date). Available at: https://eldenring.fandom.com/wiki/Weapon_Scaling (Accessed: 20 February 2025).