



Location	Test Case	Input	Expected Output	Actual Output	
Start Menu displayStartMenu(); menuChoice(A);	1	1	create Vampire object	create Vampire object	
Start Menu displayStartMenu(); menuChoice(A);	2	2	create Barbarian object	create Barbarian object	
Start Menu displayStartMenu(); menuChoice(A);	3	3	create Blue Men object	create Blue Men object	
Start Menu displayStartMenu(); menuChoice(A);	4	4	create Medusa object	create Medusa object	
Start Menu displayStartMenu(); menuChoice(A);	5	5	create Harry Potter object	create Harry Potter object	
Start Menu displayStartMenu(); menuChoice(A);	Integer not 1 or 2	8	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid	
Start Menu displayStartMenu(); menuChoice(A);	Character	t	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid	
Start Menu displayStartMenu(); menuChoice(A);	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid	
Start Menu displayStartMenu(); menuChoice(A);	String(char digit)	a7	Prompt user to enter new choice until valid Prompt user to enter new valid		
Start Menu displayStartMenu(); menuChoice(A);	String (digit char)	5d	Prompt user to enter new choice until valid	ntil Prompt user to enter new choice until valid	
Start Menu displayStartMenu(); menuChoice(A);	String(char digit char)	s5s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid	

Start Menu displayStartMenu();	float	3.3	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
menuChoice(A);				

Location	Test Case	Input	Expected Output	Actual Output
Start Menu displayStartMenu(); menuChoice(B);	1	1	create Vampire object	create Vampire object
Start Menu displayStartMenu(); menuChoice(B);	2	2	create Barbarian object	create Barbarian object
Start Menu displayStartMenu(); menuChoice(B);	3	3	create Blue Men object	create Blue Men object
Start Menu displayStartMenu(); menuChoice(B);	4	4	create Medusa object	create Medusa object
Start Menu displayStartMenu(); menuChoice(B);	5	5	create Harry Potter object	create Harry Potter object
Start Menu displayStartMenu(); menuChoice(B);	Integer not 1 or 2	8	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Start Menu displayStartMenu(); menuChoice(B);	Character	t	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Start Menu displayStartMenu(); menuChoice(B);	String (only char)	abc	Prompt user to enter new choice until Prompt user to enter new valid valid	
Start Menu displayStartMenu(); menuChoice(B);	String(char digit)	a7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Start Menu displayStartMenu(); menuChoice(B);	String (digit char)	5d	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Start Menu displayStartMenu(); menuChoice(B);	String(char digit char)	s5s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Start Menu displayStartMenu(); menuChoice(B);	float	3.3	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Location	Test Case	Input	Expected Output	Actual Output
End Menu displayEndMenu(); menuChoice();	1	1	go back to character choice	go back to character choice
End Menu displayEndMenu(); 2 2 exit program menuChoice();		exit program		
Fnd Menu		Prompt user to enter new choice until valid	Prompt user to enter new choice until valid	
End Menu displayEndMenu(); menuChoice();	Character t Prompt user to enter new choice until Prompt user n		Prompt user to enter new choice until valid	
End Menu displayEndMenu(); menuChoice();	String (only char)	abc	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
End Menu displayEndMenu(); menuChoice();	String(char digit)	a7	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
Fnd Menu		Prompt user to enter new choice until valid		

End Menu displayEndMenu(); menuChoice();	String(char digit char)	s5s	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid
End Menu displayEndMenu(); menuChoice();	float	3.3	Prompt user to enter new choice until valid	Prompt user to enter new choice until valid

Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	В		attacker type	barbarian	correct
	a		defender type	vampire	correct
	r		defender armor	1	correct
	b		defend strength	correct value from last round	correct
	а	Round 1	attack	2 die rolls betwee 1-6	correct
	r	NOUTIU 1	defense	1 die roll between 1-6 or charm	correct
Ĵ.	b		points of damage	correct value of attack - defense or 0 if charm used	correct
Ove	i		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
ne(	a		new strength	current strength - damage	correct
gameOver()	n		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	vampire	correct
ult	٧		defender type	barbarian	correct
roundResults()			defender armor	0	correct
pur	V a	Round 2	defend strength	correct value from last round	correct
rot			attack	1 random die roll between 1 -12	correct
			defense	2 die rolls betwee 1-6	correct
	m		points of damage	correct value of attack -defense	correct
	р		applied damage after armor	same as points of damage	correct
	i		new strength	current strength - damage	correct
	r		gameOver check	if strength <= 0 then game ends	correct
	е	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	_		Test Case	Expected Output	<b>Actual Output</b>
	2		attacker type	barbarian	correct
	a		defender type	barbarian	correct
	b		defender armor	0	correct
	-		defend strength	correct value from last round	correct
	a	Round 1	attack	2 die rolls betwee 1-6	correct
	;	Nouriu 1	defense	2 die rolls betwee 1-6	correct
r()	1		points of damage	correct value of attack -defense	correct
gameOver()	a		applied damage after armor	same as points of damage	correct
ne(	n		new strength	current strength - damage	correct
gar	.,		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()	V		attacker type	barbarian	correct

불			defender type	barbarian	correct
roundResult	R		defender armor	0	correct
ndF	В		defend strength	correct value from last round	correct
rou	а		attack	2 die rolls betwee 1-6	correct
	r	Round 2	defense	2 die rolls betwee 1-6	correct
	b		points of damage	correct value of attack -defense	correct
	a		applied damage after armor	same as points of damage	correct
	r :		new strength	current strength - damage	correct
	1		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
	a	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	В		attacker type	barbarian	correct
	а		defender type	Blue Men	correct
	r		defender armor	3	correct
	b		defend strength	correct value from last round	correct
	а	Round 1	attack	2 die rolls betwee 1-6	correct
	r	Nouriu 1	defense	correct die rolls based on strenght(1-3) between 1-6	correct
r()	i		points of damage	correct value of attack -defense	correct
OVE	а		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
gameOver()	n		new strength	current strength - damage	correct
gal			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
	V		attacker type	Blue Men	correct
sult			defender type	barbarian	correct
oundResults()			defender armor	0	correct
pun	В		defend strength	correct value from last round	correct
S	1	Round 2	attack	2 die rolls betwee 1-10	correct
	u	Nouna 2	defense	2 die rolls betwee 1-6	correct
	е		points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
	M		new strength	current strength - damage	correct
	е		gameOver check	if strength <= 0 then game ends	correct
	n	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	barbarian	correct

	В		defender type	Medusa	correct
	В				
			defender armor	3	correct
	а		defend strength	correct value from last round	correct
	r	Round 1	attack	2 die rolls betwee 1-10	correct
	b	Nouriu 1	defense	1 die roll between 1-6	correct
r()	i		points of damage	correct value of attack -defense	correct
Ş	а		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
ne(	n		new strength	current strength - damage	correct
gar			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
roundResults() gameOver()	٧		attacker type	medusa	correct
L L			defender type	barbarian	correct
Res			defender armor	0	correct
pu	М		defend strength	correct value from last round	correct
ron	е		attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	d	Round 2	defense	2 die rolls betwee 1-6	correct
	u		points of damage	correct value of attack -defense	correct
	S		applied damage after armor	same as points of damage	correct
	а		new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location I	Input		Test Case	Expected Output	<b>Actual Output</b>
	а		attacker type	barbarian	correct
	r		defender type	Harry Potter	correct
	b		defender armor	0	correct
	а		defend strength	correct value from last round	correct
	r	Round 1	attack	2 die rolls betwee 1-10	correct
	i	NOUTIU 1	defense	2 die rolls between 1-6	correct
5	а		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
Ove	n		applied damage after armor	same as points of damage	correct
ne(			new strength	current strength(strength 20 if Hogwarts used) - damage	correct
Results() gameOver()	V		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
9()			attacker type	Harry Potter	correct
<b>4</b>	Н		defender type	barbarian	correct
	а		defender armor	0	correct

pur	r		defend strength	correct value from last round	correct
Į.	r	Round 2	attack	2 die rolls between 1-6	correct
	У	Kouliu Z	defense	2 die rolls betwee 1-6	correct
	Р		points of damage	correct value of attack -defense	correct
	0		applied damage after armor	same as points of damage	correct
	t		new strength	current strength - damage	correct
	t		gameOver check	if strength <= 0 then game ends	correct
	е	game over	loop contunies until a character dies	proccend to end menu when character dies	correct

Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Blue Men	correct
	В		defender type	vampire	correct
	- 1		defender armor	1	correct
	u		defend strength	correct value from last round	correct
	е	Round 1	attack	2 die rolls betwee 1-10	correct
		Nound 1	defense	1 die roll between 1-6 or charm	correct
Ć.	М		points of damage	correct value of attack - defense or 0 if charm used	correct
Ove	е		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
ne(	n		new strength	current strength - damage	correct
roundResults() gameOver()			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
9()	٧		attacker type	vampire	correct
u it	•		defender type	Blue Men	correct
Res			defender armor	3	correct
pur	V		defend strength	correct value from last round	correct
ror	а	Round 2	attack	1 random die roll between 1 -12	correct
	m	Rouna 2	defense	1 die roll between 1-6	correct
	р		points of damage	correct value of attack -defense	correct
	i		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	r	r e	new strength	current strength - damage	correct
	е		gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	В		attacker type	Blue Men	correct
	- 1		defender type	barbarian	correct
	u		defender armor	0	correct
	е		defend strength	correct value from last round	correct
		Round 1	attack	2 die rolls betwee 1-10	correct
	М	Mound 1	defense	2 die rolls betwee 1-6	correct
:r()	е		points of damage	correct value of attack -defense	correct
gameOver()	n		applied damage after armor	same as points of damage	correct
me(			new strength	current strength - damage	correct
gai	٧		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()			attacker type	barbarian	correct

불			defender type	Blue Men	correct
roundResult	В		defender armor	3	correct
ndi	а		defend strength	correct value from last round	correct
rou	r b a		attack	2 die rolls betwee 1-6	correct
		Round 2	defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
	r		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	i		new strength	current strength - damage	correct
	а		gameOver check	if strength <= 0 then game ends	correct
	n	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	В		attacker type	Blue Men	correct
	D I		defender type	Blue Men	correct
			defender armor	3	correct
	u e	Round 1	defend strength	correct value from last round	correct
	e M		attack	2 die rolls betwee 1-10	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
() 	e		points of damage	correct value of attack -defense	correct
Ove Ove	n		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
me	"		new strength	current strength - damage	correct
roundResults() gameOver()	V		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
()s:	V		attacker type	Blue Men	correct
sult	•		defender type	Blue Men	correct
<u>Re</u>	В		defender armor	3	correct
pun	ı		defend strength	correct value from last round	correct
Š	u	Round 2	attack	2 die rolls betwee 1-10	correct
	e	Nouna 2	defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
	М		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	e		new strength	current strength - damage	correct
	n		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Blue Men	correct

			defender type	Madusa	correct
	В		defender type defender armor	Medusa	correct
	1			3	correct
	u		defend strength	correct value from last round	correct
	e	Round 1	attack	2 die rolls betwee 1-10	correct
_			defense	1 die roll between 1-6	correct
er()	М		points of damage	correct value of attack -defense	correct
Õ	е		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
me	n		new strength	current strength - damage	correct
roundResults() gameOver()			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
()s:	V		attacker type	medusa	correct
sult	v		defender type	Blue Men	correct
Re	•		defender armor	3	correct
pur	М		defend strength	correct value from last round	correct
ro	e	Round 2	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	d	Nouriu 2	defense	1 die roll between 1-6	correct
	u		points of damage	correct value of attack -defense	correct
	•		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	S		new strength	current strength - damage	correct
	а		gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	В		attacker type	Blue Men	correct
	- 1		defender type	Harry Potter	correct
	u		defender armor	0	correct
	е		defend strength	correct value from last round	correct
	М	Round 1	attack	2 die rolls betwee 1-10	correct
	е	Noullu 1	defense	2 die rolls between 1-6	correct
r()	n		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
Ove			applied damage after armor	same as points of damage	correct
ne(	٧		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
gar			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
0			attacker type	Harry Potter	correct
Š	Н		4.004.01.01.07.00		
Results() gameOver()	H a		defender type	Blue Men	correct

pur	r		defend strength	correct value from last round	correct
Į į	У	Round 2	attack	2 die rolls between 1-6	correct
	Р	Kouliu 2	defense	1 die roll between 1-6	correct
	О		points of damage	correct value of attack -defense	correct
	t		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	t		new strength	current strength - damage	correct
	е		gameOver check	if strength <= 0 then game ends	correct
	r	game over	loop contunies until a character dies	proccend to end menu when character dies	correct

Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	Н		attacker type	Harry Potter	correct
	а		defender type	vampire	correct
	r		defender armor	1	correct
	r		defend strength	correct value from last round	correct
	у	Round 1	attack	2 die rolls between 1-6	correct
	Р	Kouliu 1	defense	1 die roll between 1-6 or charm	correct
Ć.	0		points of damage	correct value of attack - defense or 0 if charm used	correct
Ove	t		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
ne(	t		new strength	current strength - damage	correct
gar	е		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
roundResults() gameOver()	r		attacker type	vampire	correct
in the			defender type	Harry Potter	correct
Res	٧		defender armor	0	correct
pur			defend strength	correct value from last round	correct
ro	V	Round 2	attack	1 random die roll between 1 -12	correct
	a	Roullu 2	defense	2 die rolls between 1-6	correct
	m		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
	р		applied damage after armor	same as points of damage	correct
	i		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
	r		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
	е	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	r		attacker type	Harry Potter	correct
	r		defender type	barbarian	correct
	у		defender armor	0	correct
	Р		defend strength	correct value from last round	correct
	0	Round 1	attack	2 die rolls between 1-6	correct
	t	Nouna 1	defense	2 die rolls betwee 1-6	correct
).i	t		points of damage	correct value of attack -defense	correct
gameOver()	е		applied damage after armor	same as points of damage	correct
me(	r		new strength	current strength - damage	correct
gal			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()	٧		attacker type	barbarian	correct

븕			defender type	Harry Potter	correct
roundResult	В		defender armor	0	correct
pu	а	Da vila	defend strength	correct value from last round	correct
ron	r		attack	2 die rolls betwee 1-6	correct
	b	Round 2	defense	2 die rolls between 1-6	correct
	a		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
	r		applied damage after armor	same as points of damage	correct
	i		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
	а		gameOver check	if strength <= 0 then game ends	correct
	n	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	a		attacker type	Harry Potter	correct
	r		defender type	Blue Men	correct
	r		defender armor	3	correct
	У		defend strength	correct value from last round	correct
	Р	Round 1	attack	2 die rolls between 1-6	correct
	0	Nound 1	defense	correct die rolls based on strenght(1-3) between 1-6	correct
) i	t		points of damage	correct value of attack -defense	correct
Ove	t		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
me	е		new strength	current strength - damage	correct
roundResults() gameOver()	r		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
() s:			attacker type	Blue Men	correct
sult	٧		defender type	Harry Potter	correct
Re	•		defender armor	0	correct
pun	В		defend strength	correct value from last round	correct
ō	- 1	Round 2	attack	2 die rolls betwee 1-10	correct
	u	Rouna 2	defense	2 die rolls between 1-6	correct
	е		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
	М		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
	е		gameOver check	if strength <= 0 then game ends	correct
	n	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Harry Potter	correct

	Н		defender type	Medusa	correct
	а		defender armor	3	correct
	r		defend strength	correct value from last round	correct
	r	Round 1	attack	2 die rolls between 1-6	correct
	У		defense	1 die roll between 1-6	correct
er()	Р		points of damage	correct value of attack -defense	correct
ŏ	0		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
me(	t		new strength	current strength - damage	correct
roundResults() gameOver()	t		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()	е		attacker type	medusa	correct
ult	r		defender type	Harry Potter	correct
Res			defender armor	0	correct
pu	V		defend strength	correct value from last round	correct
rou		Round 2	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	М	Rouna 2	defense	2 die rolls between 1-6	correct
	е		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
	d		applied damage after armor	same as points of damage	correct
	u		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
	S		gameOver check	if strength <= 0 then game ends	correct
	а	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	у		attacker type	Harry Potter	correct
	Р		defender type	Harry Potter	correct
	О		defender armor	0	correct
	t		defend strength	correct value from last round	correct
	t	Round 1	attack	2 die rolls between 1-6	correct
	е	Kouliu 1	defense	2 die rolls between 1-6	correct
-C	r		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
Ove			applied damage after armor	same as points of damage	correct
neC	٧		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
gan			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
0;	Н		attacker type	Harry Potter	correct
ults	а		defender type	Harry Potter	correct
Results() gameOver()	r		defender armor	0	correct

pur	r		defend strength	correct value from last round	correct
ror	У	Round 2	attack	2 die rolls between 1-6	correct
	Р	Kouria 2	defense	2 die rolls between 1-6	correct
	0		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
	t		applied damage after armor	same as points of damage	correct
	t		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
	е		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
	r	game over	loop contunies until a character dies	proccend to end menu when character dies	correct

Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Medusa	correct
			defender type	vampire	correct
	М		defender armor	1	correct
	е		defend strength	correct value from last round	correct
	d	Round 1	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	u	Roullu 1	defense	1 die roll between 1-6 or charm	correct
r.O	S		points of damage	correct value of attack - defense or 0 if charm used	correct
roundResults() gameOver()	a		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
ne(			new strength	current strength - damage	correct
gar	٧		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()			attacker type	vampire	correct
ult			defender type	Medusa	correct
Res	V		defender armor	3	correct
pur	a		defend strength	correct value from last round	correct
ro	m	Round 2	attack	1 random die roll between 1 -12	correct
	р	Rouna 2	defense	1 die roll between 1-6	correct
	i		points of damage	correct value of attack -defense	correct
	r		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	е		new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Medusa	correct
	М		defender type	barbarian	correct
	е		defender armor	0	correct
	d		defend strength	correct value from last round	correct
	u	Round 1	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	S	Round	defense	2 die rolls betwee 1-6	correct
3r()	а		points of damage	correct value of attack -defense	correct
gameOver()			applied damage after armor	same as points of damage	correct
me	٧		new strength	current strength - damage	correct
	•		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()			attacker type	barbarian	correct

븕	В		defender type	Medusa	correct
Res	a		defender armor	3	correct
roundResult	r		defend strength	correct value from last round	correct
rou	b	Round 2	attack	2 die rolls betwee 1-6	correct
	а	Rouna 2	defense	1 die roll between 1-6	correct
	r		points of damage	correct value of attack -defense	correct
	i		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	a		new strength	current strength - damage	correct
	n		gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Medusa	correct
	М		defender type	Blue Men	correct
			defender armor	3	correct
	e d		defend strength	correct value from last round	correct
	u	Round 1	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	u S		defense	correct die rolls based on strenght(1-3) between 1-6	correct
er()			points of damage	correct value of attack -defense	correct
Ŏ	а		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
me	v		new strength	current strength - damage	correct
roundResults() gameOver()	V		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
()s:	•		attacker type	Blue Men	correct
sult	В		defender type	Medusa	correct
Re	ı		defender armor	3	correct
oun o	u		defend strength	correct value from last round	correct
5	e	Round 2	attack	2 die rolls betwee 1-10	correct
	ŭ	nouna 2	defense	1 die roll between 1-6	correct
	М		points of damage	correct value of attack -defense	correct
	e		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	n		new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	Medusa	correct

			defender type	Medusa	correct
			defender armor	3	correct
	М		defend strength	correct value from last round	correct
	е		attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	d	Round 1	defense	1 die roll between 1-6	correct
Ç	u		points of damage	correct value of attack -defense	correct
ver	S		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
Dec	а		new strength	current strength - damage	correct
roundResults() gameOver()			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
9	V		attacker type	Medusa	correct
ults	•		defender type	Medusa	correct
Res			defender armor	3	correct
l pu	М		defend strength	correct value from last round	correct
rou	е	Daywad 2	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	d	Round 2	defense	1 die roll between 1-6	correct
	u		points of damage	correct value of attack -defense	correct
	S		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	а		new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	а		attacker type	Medusa	correct
	m		defender type	Harry Potter	correct
	р		defender armor	0	correct
	i		defend strength	correct value from last round	correct
	r	Round 1	attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	е	Round 1	defense	2 die rolls between 1-6	correct
er()			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
Results() gameOver()	V		applied damage after armor	same as points of damage	correct
me	•		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
gal			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
()s:	Н		attacker type	Harry Potter	correct
sult	а		defender type	Medusa	correct
			defender armor	3	correct

I	pur	r		defend strength	correct value from last round	correct
	ror	У	Round 2	attack	2 die rolls between 1-6	correct
		Р	Kouria 2	defense	1 die roll between 1-6	correct
		0		points of damage	correct value of attack -defense	correct
		t		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
		t		new strength	current strength - damage	correct
		е		gameOver check	if strength <= 0 then game ends	correct
		r	game over	loop contunies until a character dies	proccend to end menu when character dies	correct

Location	Input		Test Case	Expected Output	Actual Output
			attacker type	vampire	correct
	V		defender type	vampire	correct
	_		defender armor	1	correct
	a		defend strength	correct value from last round	correct
	m	Round 1	attack	1 random die roll between 1 -12	correct
	р :	Kouna 1	defense	1 die roll between 1-6 or charm	correct
Or	I		points of damage	correct value of attack - defense or 0 if charm used	correct
Ove	r		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
ne(	е		new strength	current strength - damage	correct
roundResults() gameOver()			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
9()	V		attacker type	vampire	correct
ult.	•		defender type	vampire	correct
Res	.,		defender armor	1	correct
pu	V	Round 2	defend strength	correct value from last round	correct
rot	a m p i r		attack	1 random die roll between 1 -12	correct
			defense	1 die roll between 1-6 or charm	correct
			points of damage	correct value of attack - defense or 0 if charm used	correct
			applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
	е		gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	V		attacker type	vampire	correct
	a		defender type	barbarian	correct
	a m		defender armor	0	correct
	g		defend strength	correct value from last round	correct
	i i	Round 1	attack	1 random die roll between 1 -12	correct
	r	Nound 1	defense	2 die rolls betwee 1-6	correct
) ir()	e		points of damage	correct value of attack -defense	correct
Ove	e		applied damage after armor	same as points of damage	correct
me(	V		new strength	current strength - damage	correct
s() gameOver()	V		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()	•		attacker type	barbarian	correct

복			defender type	vampire	correct
roundResult	В		defender armor	1	correct
n di	а		defend strength	correct value from last round	correct
ro	r	Round 2	attack	2 die rolls betwee 1-6	correct
	b		defense	1 die roll between 1-6 or charm	correct
	а		points of damage	correct value of attack - defense or 0 if charm used	correct
	r		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
	1		new strength	current strength - damage	correct
	a		gameOver check	if strength <= 0 then game ends	correct
	n	game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	vampire	correct
	V		defender type	Blue Men	correct
	а		defender armor	3	correct
	m		defend strength	correct value from last round	correct
	р	Round 1	attack	1 random die roll between 1 -12	correct
	i	Kouliu 1	defense	correct die rolls based on strenght(1-3) between 1-6	correct
ő	r		points of damage	correct value of attack -defense	correct
Öve	е		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
ne(			new strength	current strength - damage	correct
roundResults() gameOver()	V		gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
s()			attacker type	Blue Men	correct
sult T			defender type	vampire	correct
Re	В		defender armor	1	correct
Pun	- 1		defend strength	correct value from last round	correct
<u>ō</u>	u	Round 2	attack	2 die rolls betwee 1-10	correct
	е	Nouna 2	defense	1 die roll between 1-6 or charm	correct
			points of damage	correct value of attack - defense or 0 if charm used	correct
	М		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
	е		new strength	current strength - damage	correct
	n		gameOver check	if strength <= 0 then game ends	correct
		game over	loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
			attacker type	vampire	correct

			defender type	Medusa	correct
roundResults() gameOver()	V a m	Round 1	defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
	р		defense	1 die roll between 1-6	correct
	i		points of damage	correct value of attack -defense	correct
	r		applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
	е		new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
	v		attacker type	medusa	correct
			defender type	vampire	correct
Res	M e		defender armor	1	correct
ם			defend strength	correct value from last round	correct
rou			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
	d	Round 2	defense	1 die roll between 1-6 or charm	correct
	u	game over	points of damage	correct value of attack - defense or 0 if charm used	correct
	S		applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
	а		new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends	correct
			loop contunies until a character dies	proccend to end menu when character dies	correct
Location	Input		Test Case	Expected Output	<b>Actual Output</b>
	V	Round 1	attacker type	vampire	correct
	а		defender type	Harry Potter	correct
	m		defender armor	0	correct
	р		defend strength	correct value from last round	correct
	i		attack	1 random die roll between 1 -12	correct
	r		defense	2 die rolls between 1-6	correct
C	е		points of damage	if Hogwarts used damage = 0 else attack - defense	correct
Results() gameOver()			applied damage after armor	same as points of damage	correct
me(	V		new strength	current strength(strength 20 if Hogwarts used) - damage	correct
gai			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
()s:	Н		attacker type	Harry Potter	correct
sult	а		defender type	vampire	correct
ns			defender armor	1	

pur	r	r y P Round 2 o t	defend strength	correct value from last round	correct
rou	У		attack	2 die rolls between 1-6	correct
	Р		defense	1 die roll between 1-6 or charm	correct
	o t		points of damage	correct value of attack - defense or 0 if charm used	correct
			applied damage after armor	damage - 1(armor) if negative then 0 damage applied	correct
	t		new strength	current strength - damage	correct
	е		gameOver check	if strength <= 0 then game ends	correct
	r	game over	loop contunies until a character dies	proccend to end menu when character dies	correct

## CS 162\_400

## **Project 3 Reflection**

## **Brittany Dunn**

Originally when I began writing the design plan for this project I had some struggles with how to have the attack and defend functions work correctly without having to specific the character so the correct damage would be able to be calculated after all the special abilities. However, after I reread the assignment instructions again a few more times, I noticed that the defense function was supposed to take the attack as a parameter. Knowing this detail made it a lot easier to figure out the damage while taking into account the characters special abilities. This made writing my design plan a lot easier. Also, when writing the design plan, I had considered using die objects for each character, however after some thought I decided to directly add the rolling die into the attack and defend functions without using die objects. It seemed using die objects would make it more complicated and I would run into more issues that way when I began coding.

When I began coding I had some issues with creating the character objects. Originally, I had the characters being created in a menu function which accepted int as a parameter and used the parameter in a switch case to decide which character type object to create. This worked fine the issue was being able to use the character objects to create a universe object with two characters. Originally, I had the menu choice function as void I realized to be able to use the character objects I needed to return them. So, I made the return of the function be shared\_ptr instead of void. This allowed me to access the character objects created using the menu. I choose to use smart pointers instead of pointers because they were easier to declare since character is an abstract class.

When calculating the damage, I noticed the damage could increase strength if the amount of damage after defense and armor was negative. The damage should not add to strength so I added in an if statement to account for negative damage. If the damage were to be negative it would be set to 0 damage. I could have added this to the defense function or after the armor was applied. I decided it was best to do it at the point of the program where the armor was applied. I did this because then it would not interfere with the defense function or Harry's Hogwarts ability and the user would be able to see every step of the attack, defense, abilities, damage, armor, and applied damage without interference.

The main issue I had when coding was with my random function not being random. Originally, I had been using <ctime> to help make the numbers random. But each time I ran the program the numbers were not random enough so the rounds were never ending. I removed <ctime> and this made the numbers appear to be more random and the game was completing in a reasonable number of rounds.