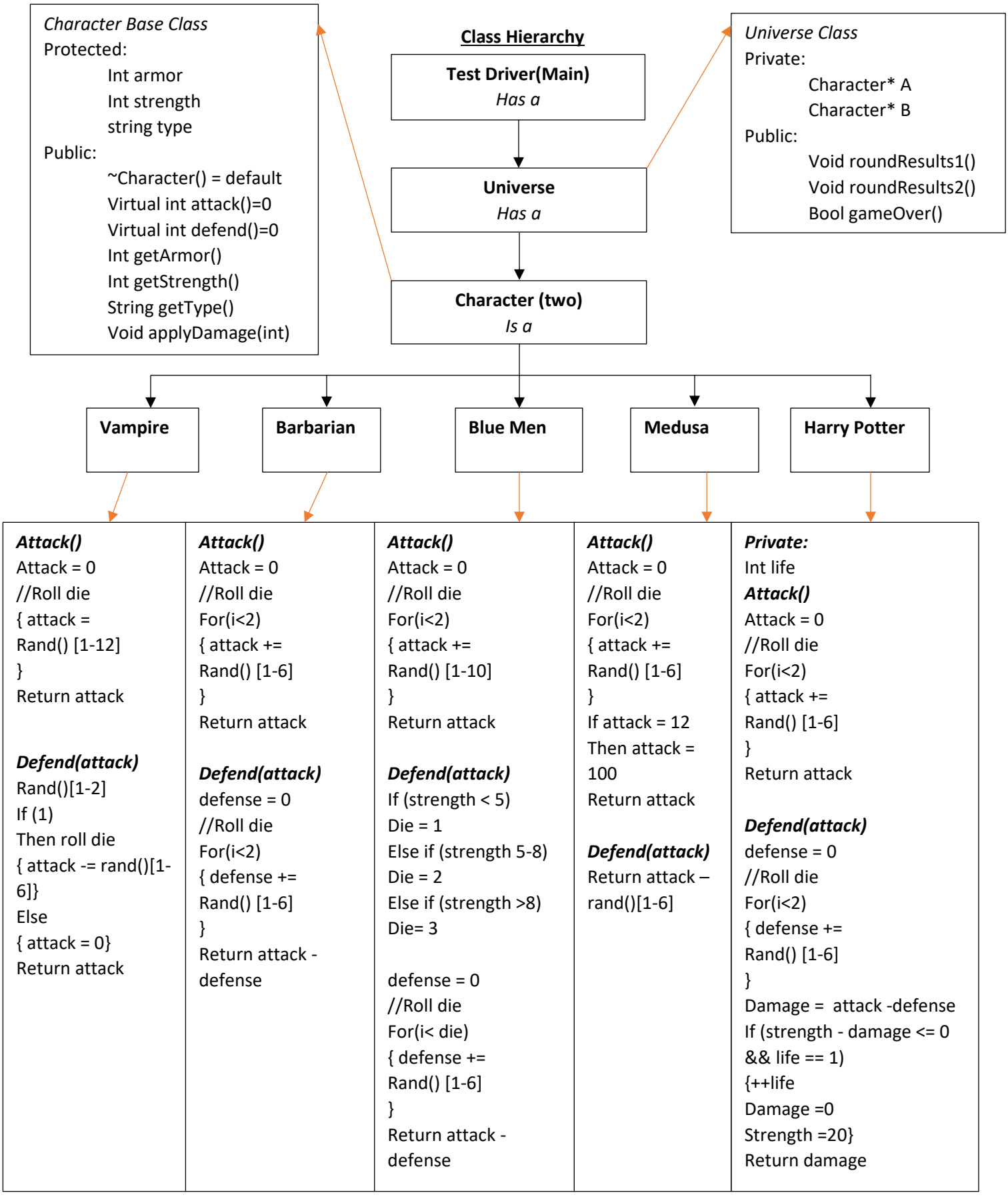
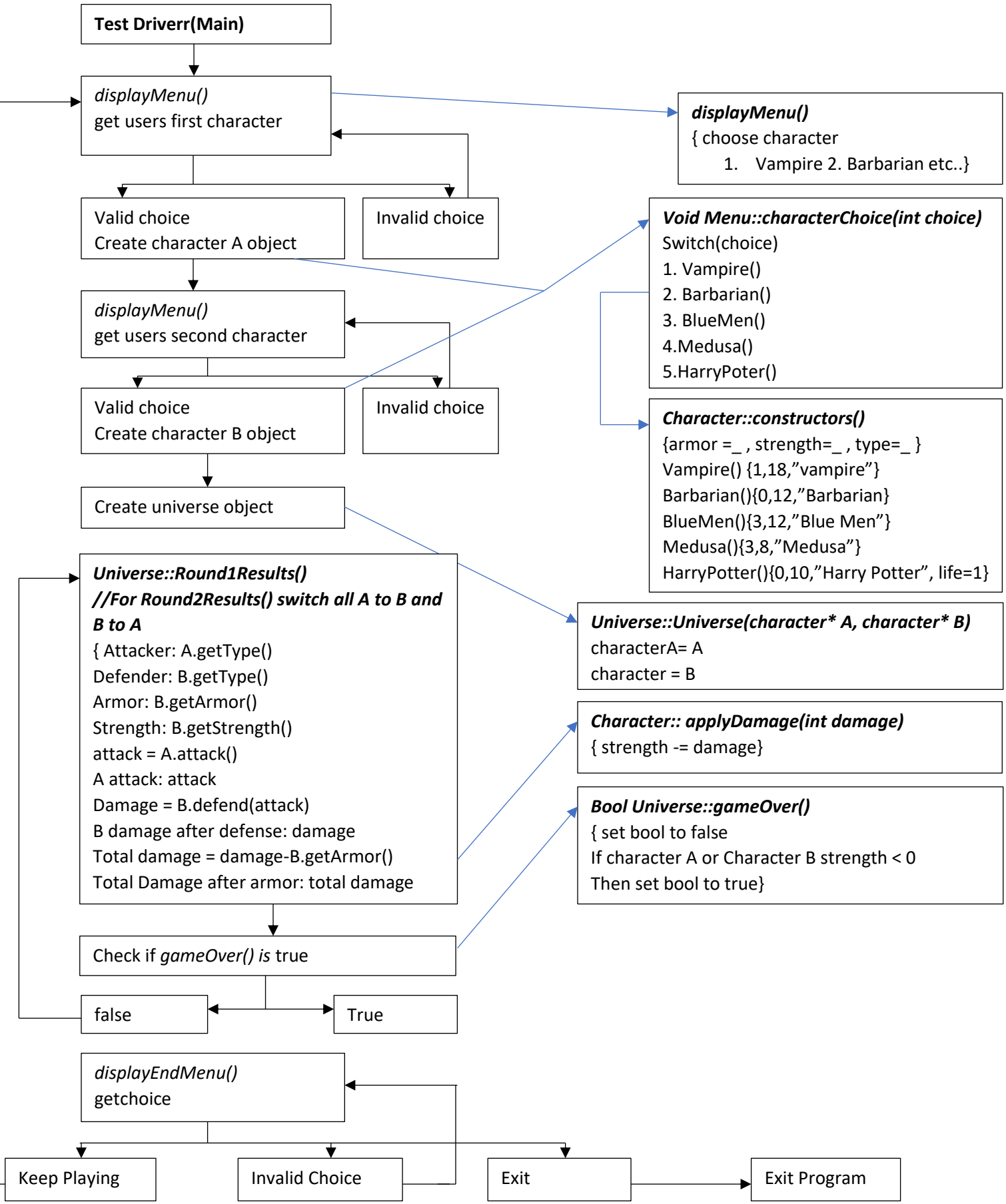

Character Base Class
Protected:
Int armor
Int strength
string type
Public:
~Character() = default
Virtual int attack()=0
Virtual int defend()=0
Int getArmor()
Int getStrength()
String getType()
Void applyDamage(int)
Class Hierarchy
Test Driver(Main)
Has a
Universe
Has a
Character (two)
Is a
Universe Class
Private:
Character* A
Character* B
Public:
Void roundResults1()
Void roundResults2()
Bool gameOver()
Vampire
Barbarian
Blue Men
Medusa
Harry Potter
Attack()
Attack = 0
//Roll die
{ attack =
Rand() [1-12]
}
Return attack
Defend(attack)
Rand()[1-2]
If (1)
Then roll die
{ attack -= rand()[1-6]}
Else
{ attack = 0}
Return attack
Attack()
Attack = 0
//Roll die
For(i<2)
{ attack +=
Rand() [1-6]
}
Return attack
Defend(attack)
defense = 0
//Roll die
For(i<2)
{ defense +=
Rand() [1-6]
}
Return attack - defense
Attack()
Attack = 0
//Roll die
For(i<2)
{ attack +=
Rand() [1-10]
}
Return attack
Defend(attack)
If (strength < 5)
Die = 1
Else if (strength 5-8)
Die = 2
Else if (strength >8)
Die= 3
defense = 0
//Roll die
For(i< die)
{ defense +=
Rand() [1-6]
}
Return attack - defense
Attack()
Attack = 0
//Roll die
For(i<2)
{ attack +=
Rand() [1-6]
}
If attack = 12
Then attack = 100
Return attack
Defend(attack)
Return attack – rand()[1-6]
Private:
Int life
Attack()
Attack = 0
//Roll die
For(i<2)
{ attack +=
Rand() [1-6]
}
Return attack
Defend(attack)
defense = 0
//Roll die
For(i<2)
{ defense +=
Rand() [1-6]
}
Damage = attack -defense
If (strength - damage <= 0
&& life == 1)
{++life
Damage =0
Strength =20}
Return damage

Program Design



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 choice until valid

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

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 valid	Prompt user to enter new choice until 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();
menuChoice();
	2	2	exit program	exit program

End Menu
displayEndMenu();
menuChoice();
	Integer not 1 or 2	8	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 valid	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

End Menu
displayEndMenu();
menuChoice();
	String (digit char)	5d	Prompt user to enter new choice until valid	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	Actual Output
roundResults() gameOver()	Barbarian v. Vampire	Round 1	attacker type	barbarian	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	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
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	vampire	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			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	Actual Output
s() gameOver()	Barbarian v	Round 1	attacker type	barbarian	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	barbarian	correct

roundResult	. B a r b a r i a n	Round 2	defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			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**	**Actual Output**
roundResults() gameOver()	B a r b a r i a n v . B l u e M e n	Round 1	attacker type	barbarian	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	Blue Men	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			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**	**Actual Output**
			attacker type	barbarian	correct

roundResults() gameOver()	Barbian v. Medusa	Round 1	defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	medusa	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			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**	**Actual Output**
Results() gameOver()	arbarian v. Ha	Round 1	attacker type	barbarian	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	Harry Potter	correct
			defender type	barbarian	correct
			defender armor	0	correct

round	r r y P o t t e	Round 2	defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			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	Actual Output
roundResults() gameOver()	Blue Men v. Vampire	Round 1	attacker type	Blue Men	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	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
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	vampire	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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	Actual Output
s() gameOver()	Blue Men v.	Round 1	attacker type	Blue Men	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	barbarian	correct

roundResult	Barbarian	Round 2	defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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**	**Actual Output**
roundResults() gameOver()	Blue Men v. Blue Men	Round 1	attacker type	Blue Men	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	Blue Men	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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**	**Actual Output**
			attacker type	Blue Men	correct

roundResults() gameOver()	Blue Men v. Medusa	Round 1	defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	medusa	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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**	**Actual Output**
Results() gameOver()	Blue Men v. Har	Round 1	attacker type	Blue Men	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	Harry Potter	correct
			defender type	Blue Men	correct
			defender armor	3	correct

round	r y P o t t e r	Round 2	defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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	Actual Output
roundResults() gameOver()	Harry Potter v. Vampire	Round 1	attacker type	Harry Potter	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	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
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	vampire	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - 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	Actual Output
s() gameOver()	Harry Potter v	Round 1	attacker type	Harry Potter	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	barbarian	correct

roundResult	.Barbarian	Round 2	defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - 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**	**Actual Output**
roundResults() gameOver()	arryPotter v. Blue Men	Round 1	attacker type	Harry Potter	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	Blue Men	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - 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**	**Actual Output**
			attacker type	Harry Potter	correct

roundResults() gameOver()	Harry Potter v. Medusa	Round 1	defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	medusa	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - 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**	**Actual Output**
Results() gameOver()	Potter v. Har	Round 1	attacker type	Harry Potter	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	Harry Potter	correct
			defender type	Harry Potter	correct
			defender armor	0	correct

round	r y P o t t e r	Round 2	defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - 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	Actual Output
roundResults() gameOver()	Medusa v. Vampire	Round 1	attacker type	Medusa	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	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
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	vampire	correct
			defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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	Actual Output
s() gameOver()	Medusa v.	Round 1	attacker type	Medusa	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	barbarian	correct

roundResult	Barbarian	Round 2	defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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**	**Actual Output**
roundResults() gameOver()	Medusa v. Blue Men	Round 1	attacker type	Medusa	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	Blue Men	correct
			defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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**	**Actual Output**
			attacker type	Medusa	correct

roundResults() gameOver()	Medusa v. Medusa	Round 1	defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	Medusa	correct
			defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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**	**Actual Output**
Results() gameOver()	ampire v. Har	Round 1	attacker type	Medusa	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	Harry Potter	correct
			defender type	Medusa	correct
			defender armor	3	correct

round	r y P o t t e r	Round 2	defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			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	Actual Output
roundResults() gameOver()	Vampire v. Vampire	Round 1	attacker type	vampire	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			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
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	vampire	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			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
			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	Actual Output
s() gameOver()	Vampire v.	Round 1	attacker type	vampire	correct
			defender type	barbarian	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	2 die rolls betwee 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	barbarian	correct

roundResult	Barbarian	Round 2	defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-6	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
			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**	**Actual Output**
roundResults() gameOver()	Vampire v. Blue Men	Round 1	attacker type	vampire	correct
			defender type	Blue Men	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	correct die rolls based on strenght(1-3) between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	Blue Men	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls betwee 1-10	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
			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**	**Actual Output**
			attacker type	vampire	correct

roundResults() gameOver()	Vampire v. Medusa	Round 1	defender type	Medusa	correct
			defender armor	3	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	1 die roll between 1-6	correct
			points of damage	correct value of attack -defense	correct
			applied damage after armor	damage - 3(armor) if negative then 0 damage applied	correct
			new strength	current strength - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
		Round 2	attacker type	medusa	correct
			defender type	vampire	correct
			defender armor	1	correct
			defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6 if 12 then attack is 100	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
			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**	**Actual Output**
Results() gameOver()	Vampire v. Harr	Round 1	attacker type	vampire	correct
			defender type	Harry Potter	correct
			defender armor	0	correct
			defend strength	correct value from last round	correct
			attack	1 random die roll between 1 -12	correct
			defense	2 die rolls between 1-6	correct
			points of damage	if Hogwarts used damage = 0 else attack - defense	correct
			applied damage after armor	same as points of damage	correct
			new strength	current strength(strength 20 if Hogwarts used) - damage	correct
			gameOver check	if strength <= 0 then game ends; proceed to end menu	correct
			attacker type	Harry Potter	correct
			defender type	vampire	correct
			defender armor	1	correct

round	r y P o t t e r	Round 2	defend strength	correct value from last round	correct
			attack	2 die rolls between 1-6	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
			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

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.