

COM326 Assignment 2

Title: Role Playing Game (RPG) core classes

Issue date: Friday November 24th

Submission date: Friday December 22th 2017 before 23:59

% Contribution to module mark: 25%

Type: Group assignment, allocations listed in excel spreadsheet in BBL accompanying this assignment specification.

Assignment brief:

This assignment evaluates your understanding of key principles of class inheritance, composition, polymorphism, pointers and references.

You are required to implement the UML class hierarchy on the following page. The following must be adhered to:

- All class, function and data member names must be implemented as specified.
- The GameCharacter class is abstract, it should not be possible to create a standalone instance of this class.
- Derived classes inherit from GameCharacter and override all listed virtual functions providing custom implementations.
- Complete description for key classes/functions is provided in the table following the UML diagrams.
- Custom constructors, getters and setters should be provided for **all** data members.
- Virtual destructors must be implemented if memory is dynamically allocated!
- Data encapsulation should be enforced with access to all data members provided through getters and setters. No protected or public access specifiers.
- Getters should be Get<DataMemberName>(). For example
 - GetPunchDamage()
 - GetWeigthLimit()
 - GetHealth
- Derived classes should be able to get and set the data members of its immediate base class.
- Ensure you use pass by reference and pass by value where appropriate. Avoid making unnecessary copies and pass by value for simple types (non objects).
- Slack (<https://slack.com/>) and trello (<https://trello.com/>) are two tools commonly used across the tech industry to support group communication and project coordination/management. You should use them for this project.

COM326 Assignment 2

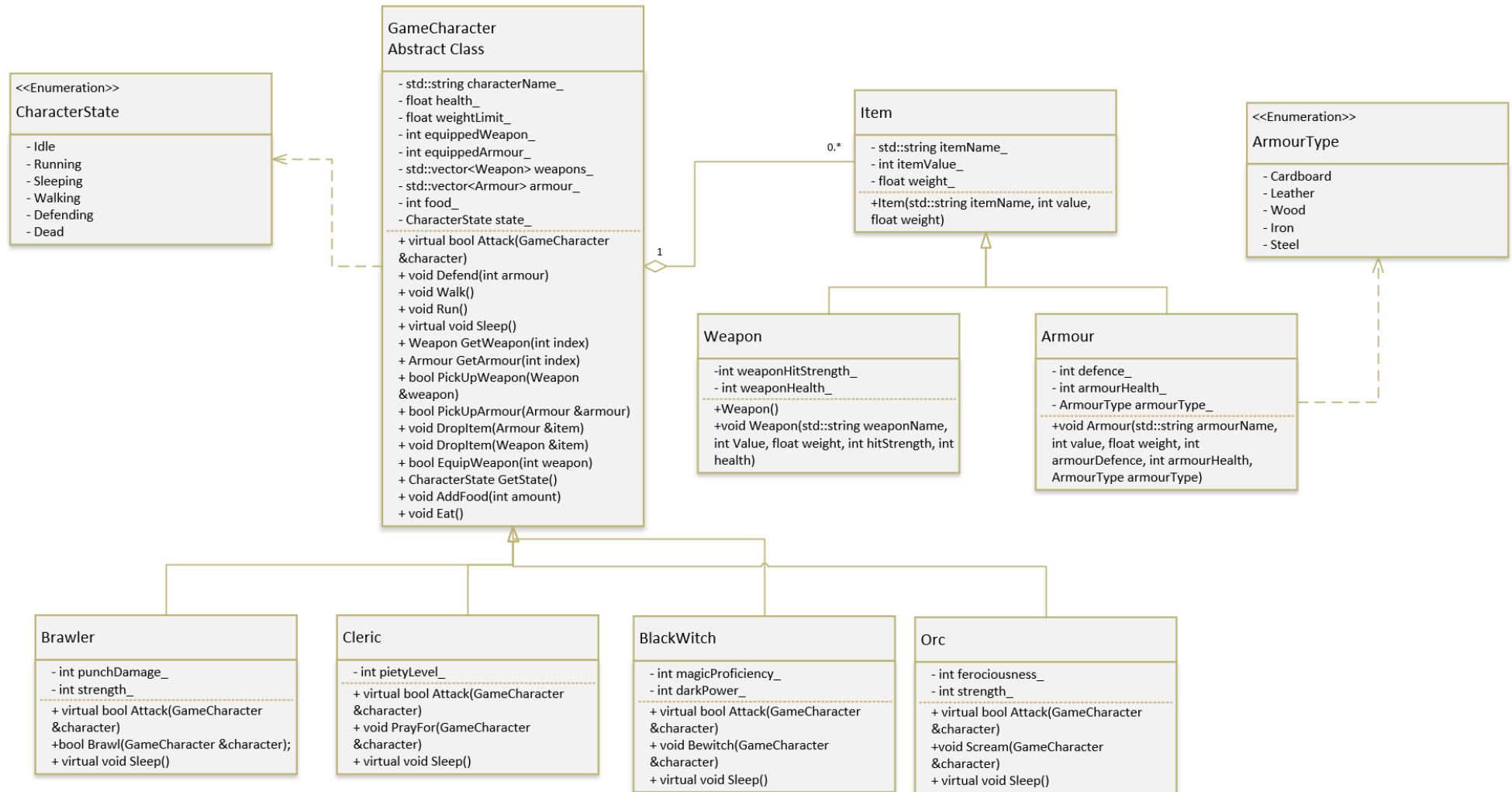


Figure 1: UML class diagram to complete

COM326 Assignment 2

Notes:

- For brevity and clarity, the following are not shown in the UML diagram, but you should be implemented:
 - Default and custom constructors
 - Data member getters and setters
 - Virtual destructors if necessary
- With regards to constructors – Parameters should be passed in matching the order the data attributes are specified. From base through to derived object. For example:
 - `Armour(std::string name, int value, float weight, int defence, int health, ArmourType type)`
 - `Brawler(std::string name, float health, float weightLimit, int food, CharacterState state, int punchDamage, int strength)`
- Assume game characters start:
 - Without weapons and armour (no need to pass vectors containing items) – so empty vectors
 - Without a weapon or armour equipped (equippedWeapon and equippedArmour both -1 denoting no item equipped)
- GameCharacter is an abstract class. It should not be possible to create instances of this class. Derived classes may or may not override base functionality/logic. So read the descriptions carefully.
- You are free to choose appropriate data container(s) to represent inventory. Options include but are not limited to arrays and vectors. The get inventory methods (armour and weapon) should function as specified regardless of the internal implementation of the class.

Class	Member function	Description
GameCharacter	Custom constructor	Custom constructors for the game characters should initialise core values. Constructors should not pass vectors of items. Just leave the armour and weapons vectors empty at initialisation. Equipped weapon and armour should be '-1' at initialisation to signify that no item has been equipped. Initial character state should be 'idle'
GameCharacter	Attack(GameCharacter &character)	Should attack the game character with the weapon. See section character attack logic for implementation details. Should also change the calling character state to Idle. Returns true if the attack was successful (damage inflicted), otherwise false. The character parameter refers to the character being attacked. EG

COM326 Assignment 2

		<code>Brawler.Attack(orc);</code> //The brawler attacks the orc GameCharacter object
GameCharacter	Defend(int armour)	Equips the armour in the vector <code>armour_</code> at index <code>int armour</code> . Also changes the character state to <code>Defending</code> . If the function parameter <code>armour</code> does not correspond to a valid armour item in the vector <code>armour_</code> then the value of <code>equippedarmour</code> item remains -1 (no item equipped).
GameCharacter	Walk	Changes the character state to <code>Walking</code>
GameCharacter	Run	Changes the character state to <code>Running</code>
GameCharacter	Sleep	Changes the character state to <code>Sleeping</code> The character <code>health</code> should also increase by: <ul style="list-style-type: none"> • 20% if brawler or cleric • 15% if BlackWitch or Orc <u>Character health should not exceed 100</u>
GameCharacter	Weapon GetWeapon(int index)	Returns a copy of the Weapon item that index.
GameCharacter	Armour GetArmour(int index)	Returns a copy of the Armour item that index.
GameCharacter	bool PickUpWeapon(Weapon &weapon)	Attempts to add the weapon to the character's weapon vector. Returns <code>true</code> if the item is added otherwise <code>false</code> . The item can only be added to the inventory if the item weight plus the total weight of all the other items in the inventory does not exceed the character's <code>weightLimit</code> data member.
GameCharacter	bool PickUpArmour(Armour &armour)	Attempts to add the armour to the character's armour vector. Returns <code>true</code> if the item is added otherwise <code>false</code> . The item can only be added to the inventory if the item weight plus the total weight of all the other items in the inventory (both weapons and armour) does not exceed the character's <code>weightLimit</code> data member.

COM326 Assignment 2

GameCharacter	Dropltem(Armour &item)	Removes the item from the character's armour inventory. Don't just check item names, check all attributes to ensure a match
GameCharacter	Dropltem(Weapon &item)	Removes the item from the character's weapon inventory
GameCharacter	bool EquipWeapon(int weapon)	Sets the <code>equippedWeapon_</code> data member to the index of the inventory that points at the weapon to equip. If the value of the weapon parameter is -1 the function sets <code>equippedWeapon_</code> to -1 signifying that the character does not have a weapon equipped. The function should return true if the value of int weapon is valid. Valid values are -1 (no weapon equipped) to <code>weapons_.size() - 1</code> .
GameCharacter	GetState	Returns the current state of the character
GameCharacter	AddFood(int amount)	Adds the <code>amount</code> of food to the <code>food_</code> value
GameCharacter	Eat()	Consumes 20% of the available food. Each unit (1) of food consumed will add 0.25 units of health to the character.
Brawler	Attack(GameCharacter character)	<p>This function largely follows the attack game logic as prescribed. However, the amount of damage inflicted increases by 5% for every 10 strength.</p> <p>Example: Target is defending so base damage is -10% of target's health Brawler strength is 35 so $(35/10 = 3) 3 * 5\% = 15$.</p> <p>$10\% + 15\% = -25\%$ damage</p>
Brawler	Brawl(GameCharacter character)	Attempts to attack the game character by punching. Follows the attack logic replacing the weapon hitstrength with punchDamage. If the Brawl attack is successful it will inflict only 50% of the damage normally associated with attacking with a weapon. Returns true if attack is successful and damage inflicted. UML updated to show return type bool.
Cleric	PrayFor(GameCharacter character)	The function has a 50% change of success. If successful it will increase the health of the target game character by $5\% * pietyLevel$. No effect if unsuccessful. The cleric can pray for themselves.

COM326 Assignment 2

BlackWitch	Bewitch(GameCharacter character)	<p>The function attempts to place the target character in a sleeping state. There is a 10% chance of the Bewitch function being successful. This increases by 5% with each level of magicProficiency. So a magic proficiency level of 5 would mean</p> <p>$10\% + (5\% * 5) = 35\%$ chance of success</p>
BlackWitch	Attack(GameCharacter character)	<p>This function largely follows the attack game logic as prescribed. However, if the darkPower is at 100% the damage inflicted is increased by 20%.</p>
Orc	Scream(GameCharacter character)	<p>This function has a 20% chance of making the target game character flee (change character state to running). The chance of success increases by 5% with each level of ferociousness.</p>
Orc	Attack(GameCharacter character)	<p>This function largely follows the attack game logic as prescribed. However, the amount of damage inflicted increases by 5% for every 10 strength.</p> <p>Example: Target is defending so base damage is -10% of target's health Brawler strength is 35 so $(35/10 = 3) 3 * 5\% = 15\%$.</p> <p>$10\% + 15\% = -25\%$ damage</p>

COM326 Assignment 2

Character Attack logic

The character attack logic should work as follows:

- The character cannot attack another character if:
 - They do not have a weapon equipped (exception for brawler)
 - Brawl function called if no weapon equipped
 - Their health is ≤ 20 ;
 - The character being attacked is dead
- If the hit strength of the weapon used by the attacking character is less than the defence value of the armour of the defending character, then there is only a 20% chance of the attack being successful. Otherwise there is a 60% chance of a successful attack.
- If the defending character is not wearing any armour, then the chance of a successful attack increases to 80%.
- An unsuccessful attack will result in no damage on the character being attacked. However, the health of the weapon used by the attacking character will be reduced by a random value between 10% and 20% if the defender is wearing armour of any type.
- Weapons with health ≤ 0 should be removed from the inventory.
- If the attack is deemed **successful** it will inflict damage on the defending character as follows:
 - – 10% health to the character being attacked if their state is `Defending`
 - – 100% health to the character being attacked if their state is `Sleeping`
 - No effect if the character being attacked state is `Dead`
 - – 20% health to the character being attacked for all other states.
 - The health of the defending characters armour will also be reduced by 10%
 - Armour with health ≤ 0 should be removed from the inventory.

COM326 Assignment 2

Additional notes:

- Your assignment submissions will be automatically tested using unit tests. The unit tests will expect you to have implemented the classes as specified in the UML diagram. If you provide alternatively named functions:
 - You are failing to meet the specification
 - The unit test will fail and you will not receive marks for that feature.
- You should use the supplied visual studio solution and projects for your assignment. Rename the solution/project to *GroupX*, replacing x with your group number.
- If should use GitHub to enable group delivery of your assignment.
 - **Add me as a collaborator to your GitHub project.** s.wilson@ulster.ac.uk
 - Ensure you make the repository private. Public repositories should not be used.
- You must document the design, planning and implementation of your assignment solution using slack/trello. There should be clear evidence of planning, task coordination, team collaboration and communication over the period of the module.

Marking scheme:

You will be awarded specific [marks] for each Unit test your solution passes. The unit tests will explicitly test the following:

- Functions associated with correctly creating and managing hero and game character objects [20]
- Functions associated with Character inventory management [30]
- Functions associated with Character conflict (examples include: attack, brawl, Bewitch) [40]
- Good code commentary and compliance with UU Magee coding style guide [10]

Submission details:

- You should submit your assignment (one submission per group) to blackboard on or before the deadline.
- You should remove temporary build files from visual studio and zip your visual studio project & solution (see video from assignment 1). If your zipped Visual Studio solution is more than 2MB then you have done it wrong!
- If you do not submit your code your submission will receive **a mark of zero**. Triple check that your .zip file contains:
 - Visual Studio solution .sln file
 - Visual Studio project files (.vcxproj and .vcxproj.filters).
 - All source files (fully commented with author info)
- Accompanying the VS .zip containing your solution you should include:
 - Word document with weekly screenshots from trello & slack (in sequence) documenting the planning, coordination and implementation process.
 - Agreed level of contribution sheets. One for each group. (See last page)
 - Screenshot of your github repository contributions/commits

COM326 Assignment 2

Group Assessment forms

Group assessment

Use the following table to assess the contribution made by other members of your group. Did everyone contribute equally? If, so each member of the group should be given 20% (assuming a group of five). If two members of the group contributed significantly more than their colleagues, the distribution of percentage weighting should reflect this, e.g. 10+15+15+30+30.

In the following table allocate a mark for each member of the group and list the major contributions of the group members to the completion of the assignment.

Group member (Print student ID)	Description of the work undertaken by the student in completion of the assignment	% Contribution to task completion

Students who do not contribute to the assignment will receive a mark of zero. Individual marks will be scaled accordingly if contribution < 33.3%. No additional marks given for individual contributions that exceed 33.3%.