

## Index

- [Description](#)
- [Errors](#)
- [Adjustments](#)
- [Fixes](#)

## Description

This file contains all the bugs, errors and mistakes along with the fixes done in the Auto Battle RPG, provided by Kokku. This file is an attempt to document all the changes, why they were done and how they affect the project as a whole.

## Errors

This section contains all the bugs, errors and mistakes of the code, things that stop the code from executing along with mistakes that make the game loop behave in undesirable ways.

1. **Vector: No such file or directory(in Grid.h):** simple typo, it should be vector without the capital 'v'.
2. **Vector has not been declared AND expected ',' or '...' before '<' token(in Character.h):** an error that occurs by not declaring 'using namespace std' before the class and after the libraries.
3. **'Shared\_ptr' does not name a type(in BattleField.h):** the code didn't include the '<memory>' library.
4. **Switch quantity not an integer AND expected unqualified-id(in BattleField.cpp):** the variable choice should be an integer so the switch may work and after changing it to an integer it doesn't need the 'std::' in front of the variable.
5. **No Matching function for call to 'getline()(in BattleField.cpp):** this function to get the line doesn't exist in the code and furthermore it isn't needed, as for the 'cin >> choice' already does that.
6. **Conversion from 'const char\*' to 'int' in a converted constant expression(in BattleField.cpp):** caused because the switch was treating the player choices (1-4) as string not as integers.
7. **Cast to pointer from integer of different size(in BattleField.cpp):** happened because classIndex, that was declared as an int was casted to an int \*, so we should pass &classIndex instead.
8. **Cannot convert 'Character\*' to 'std::shared\_ptr<Character>\*' in assignment(in BattleField.cpp):** it happens because the class Character doesn't have the type of a shared pointer.
9. **Cannot convert 'std::shared\_ptr<Character>' to 'std::shared\_ptr<Character>\*' in assignment(in BattleField.cpp and**

**BattleField.h):** simply a faulty use of a pointer, only needing to remove the '\*' in the 'std::shared\_ptr<Character>\*'.

- 10. Cannot convert 'std::shared\_ptr<Character>' to 'Character\*' (in BattleField.cpp and Character.h):** the target from the character class needs to be a shared pointer and not a Character\*, as in the BattleField.cpp code EnemyCharacter and PlayerCharacter receive a shared pointer as targets.
- 11. No matching function for call to 'std::\_\_cxx11::list<Character>::push\_back(std::shared\_ptr<Character>&)' (in BattleField.cpp):** just needed to pass the memory address of the PlayerCharacter and EnemyCharacter through the function.
- 12. No matching function for call to 'Character::Character(Types::CharacterClass\*&)' (in BattleField.cpp):** removed the '\*' and the change done on item 7. of this list (the & from '&classIndex') in 'Types::CharacterClass\* characterClass = (Types::CharacterClass\*)&classIndex'.
- 13. 'shared\_ptr' does not name a type (in CharacterClass.h):** imported the memory library.
- 14. No matching function for call to 'Types::GridBox::GridBox()' (in Character.cpp and Character.h):** had to add a '\*' in 'Types::GridBox currentBox;'.
- 15. Cannot convert 'std::shared\_ptr<Character>' to 'Character\*' (in Character.cpp and Character.h):** simply change the type that Attack() receives in Character.h to a shared pointer.
- 16. No declaration matches 'void Character::Attack(Character\*)' (in Character.cpp):** fixed by changing the declaration of the method Attack(Character\* target) to Attack(shared\_ptr<Character> target).
- 17. Request for member 'xIndex' in '((Character\*)this)->Character::currentBox', which is of pointer type 'Types::GridBox\*' (maybe you meant to use '->' ?) AND request for member 'xIndex' in '((std::\_\_shared\_ptr\_access<Character, \_\_gnu\_cxx::\_\_S\_atomic, false, false>\*) (in Character.cpp):** simply change the '.' to '->' in the line 'currentBox.xIndex > target->currentBox.xIndex'.
- 18. Cannot convert 'Types::GridBox' to 'Types::GridBox\*' in assignment (in BattleField.cpp):** added & identifier in front of '\*l\_front' on two lines ('PlayerCharacter->currentBox = &\*l\_front;' and 'EnemyCharacter->currentBox = &\*l\_front;').
- 19. Changing various address identifiers from '.' to '->' (in Character.cpp):** due to the fact that there are a various number of address identifiers done wrong on Character.cpp, I simply put them all in this error just to simplify the documentation.
- 20. No match for 'operator=' (operand types are '\_\_gnu\_cxx::\_\_alloc\_traits<std::allocator<Types::GridBox>, Types::GridBox>::value\_type' {aka 'Types::GridBox'} and 'Types::GridBox\*') (in Character.cpp and Grid.h):** changed the variable type

in Grid.h of the grids variable from 'std::vector<Types::GridBox> grids' to std::vector<Types::GridBox\*> grids.

21. **Warning: no return statement in function returning non-void(in Character.cpp):** added a temporary return statement at the CheckCloseTargets method.
22. **'printf' was not declared in this scope(in Grid.cpp):** added stdio.h so I can have access to the printf function.
23. **No matching function for call to 'Types::GridBox::GridBox()':** changed the line 'Types::GridBox\* currentgrid = new Types::GridBox();' to 'Types::GridBox\* currentgrid = grids[Columns \* i + j];'
24. **Cannot convert 'Types::GridBox\*\*' to 'Types::GridBox\*' in initialization(in BatteField.cpp):** reverted the changes of item 18. and removed the & identifier and added the \* for all the '= \*\_front' lines.
25. **Request for member 'occupied' in '\* l\_front.\_\_gnu\_cxx::\_\_normal\_iterator<Types::GridBox\*\*, std::vector<Types::GridBox\*> >::operator->()', which is of pointer type 'Types::GridBox\*' (maybe you meant to use -> ?)(in Battlefield.cpp):** added \* in front of the 'l\_front->occupied = true;' lines.
26. **ISO C++ forbids comparison between pointer and integer [-fpermissive](in BattleField.c):** removed the '->Index - 1' of the line 'if(find(battlefield->grids.begin(), battlefield->grids.end(), currentBox) != battlefield->grids.end())'.

## Adjustments

This section contains all the things that I adjusted, viewed that I thought needed adjustments simply for good practices or for optimizing the code.

1. **Non-existence of a makefile:** the code doesn't have a makefile to execute all the '.c' and '.h' files, with the existence of the makefile it makes easier for the user and the developer to execute everything without needing to know the code (which part should execute to get the full code(encapsulation)), and they don't need to type all the bash code every time.
2. **Order of the includes:** there isn't a fixed "good way" to include all the libraries of C++, but I prefer using all the system ones at the top followed by the personal '.h'.
3. **Ident inside the classes:** inside some classes the indent isn't done properly, so I had to fix it following good practices.
4. **Useless spaces and spacing:** through the code are some spacing and line breaks that fill the code with useless spaces and spacing, I removed them to make the code more easy to read.
5. **Missing spacing of problems with line breaks:** had to include some spacing and line breaks where needed to make the code easier to read.

6. **Inconsistency of the opening and closing of the brackets:** the brackets' various instances are opened in different ways(along the method or under it), so I put them under the method to keep consistency.
7. **Removing unnecessary parentheses and other punctuation:** in some lines of code there are unnecessary parentheses and other punctuations that are not needed.
8. **Fixing typos:** fixing typos on the code, such as 'charcater class.'

## Fixes

This section contains all the corrections done in the code, bugs and mistakes to optimizations done on it.

1. Fixed a type done to the vector library.
2. Added 'using namespace std' to fix the vector error.
3. Added the memory library, so the code could recognize the shared\_ptr.
4. Changed the variable type of choice to an integer.
5. Removed the 'std::' that was in front of the choice variable.
6. Removed the 'std::getline(std::cin, choice);'
7. Fixed the switch by removing the quotes from the numbers, so it used integers and not string.
8. Added the address of the variable classIndex instead of only the variable.
9. Changed the declaration of the EnemyCharacter of 'new Character(enemyClass)' to 'std::make\_shared<Character>(enemyClass)', to create a shared pointer.
10. Removing the '\*' in 'shared\_ptr<Character>\* EnemyCharacter;' .
11. Changing the target variable type from Character\* to shared\_ptr<Character>.'
12. Changed the AllPlayers->push\_back to AllPlayers.push\_back.
13. Added the memory address of the PlayerCharacter and EnemyCharacter through the push\_back function into the AllPlayers list.
14. Removed the change done on item 8. and all the '\*' from 'Types::CharacterClass\* characterClass = (Types::CharacterClass\*)&classIndex'
15. Added the memory library in CharacterClass.h.
16. Added '\*' to the line 'Types::GridBox currentBox;'.
17. Change the target type from Character\* to shared\_ptr.
18. Fixed the declaration of the Attack() method.
19. Changed the '.' to '->' in the line 'currentBox.xIndex > target->currentBox.xIndex'.
20. Added & identifier before the \*\_front.
21. Added a bunch of '->' identifiers in change of the '.' identifiers.
22. Added a temporary return statement to make the code work.
23. Added the stdio.h library.
24. Reverted the item 18 and removed &, adding \* instead.

25. Added \* in front of the 'l\_front->occupied = true;' lines.
26. Removed '-> Index - 1' of the 'if(find(battlefield->grids.begin(), battlefield->grids.end(), currentBox) != battlefield->grids.end())' line.