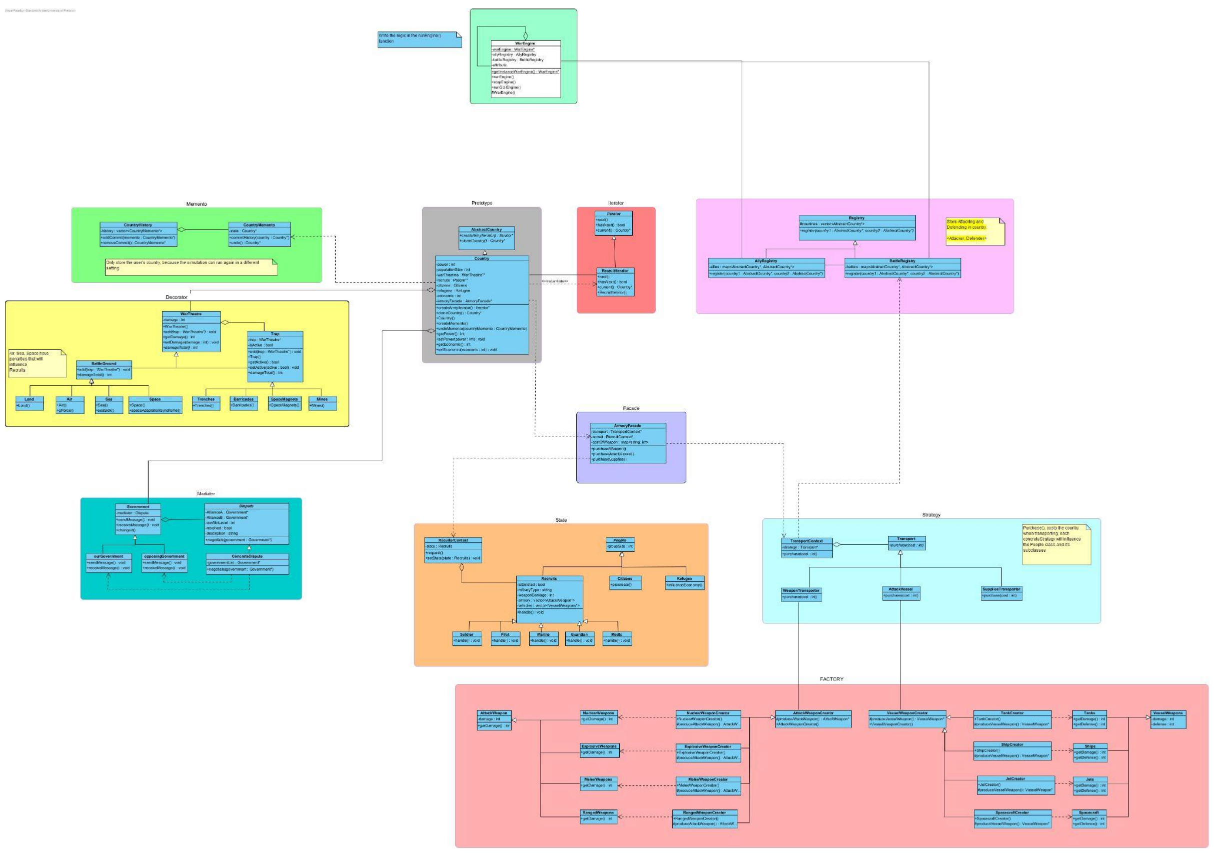
Concrete Abstracts COS 214 Project

Index

ConcretAbstracts

Task 1: Practical Assignment 5

Pre-initial UML class diagram design



Task 2: Design

2.1 Functional Requirements

War Theatres

There will be 4 types of war theatres where battle can take place. These 4 war theatres decided for the purpose of this project will be on Land, Air, Sea and Space.

Weapons

Every war requires some mode of weapons, for the purpose of the project it will be assumed that the weapons are of the physical type. These weapons will be broken down into 2 categories: Attack weapons and Vessel weapons. Attack weapons will be simulating firearms while Vessel weapons will simulate artillery type weapons (Vehicles).

- Attack Weapons
    - Ranged Weapons
    - Melee Weapons
    - Explosive Weapons
    - Nuclear Weapons
- Vessel Weapons
    - Tanks
    - Jets
    - Ships
    - SpaceCrafts

Transport

During war there is always a mode of transport. This can be to transport people or even goods. For the purposes of this project 3 types of transport and each type will cost the country. These transport types will consist of transporting weapons, vessels and finally supplies (Food, medical and so on).

People

This is the actual populations of countries. There will be 3 main types of people firstly the Citizen population group (Belongs to a country), secondly the Refugee group which is sent from allied countries and has an influence on our economy and finally the Recruit group. The recruit group is broken down even further such that specific recruit group types will go hand in hand to a specific war theatre, below is the listed recruit groups with their respective war theatres.

- Soldier(Land)
- Pilot(Air)
- Marine(Sea)
- Guardian(Space)
- Medic(All War Theatres)

Countries

Country posses a power, this is a determining factor in the warfare of the simulation in the project

- Calculated by the Economy and Group Size (Population)

Economy attribute to allow for a global currency.

- Based on GDP (Gross Domestic Product)

War Engine

Pre-Phase 1

This is the description of the war simulation:

The engine has 8 built-in countries which the user will interact with all or a certain amount depending on the choices the user makes and by chance.

The engine will start with the user designing their country. Some options are up to the user such as country name but others are up to chance such as population size.

What is up to the user

- name
- Priority: Capitalism(spend more on weapons) or Socialism(spend more on feeding)

What is up to chance

- Population size

{The object country is made}

A built-in country is picked randomly to have a dispute with your country. The dispute will be chosen randomly

{dispute object created}

Phase 1 (beginning):

Here the dispute is presented to the user. This simulates the beginnings of wars which involves social interactions, negotiations, and a lot of emotions.

This is a short story mode simulating the progressions to war.

We have an array of messages that go alongside a certain user input (yes/no). eg. "The enemy wants you to force your citizens to converge" -Yes or No?

As the story unfolds, the dispute levels rise which will lead to Phase 2

Here, in Phase 1, the user needs to input the reason for the dispute. Depending on the reason, the user's country will either be the attacker or the defender.

Reasons for dispute:

- Land (Economic gain, fight for resources) [Attacker] {get to choose war theatres in phase 2}
- Revenge [Attacker] {get to choose war theatres in phase 2}
- Religion [Defender] {random war theatres : Engine chooses}
- A country wants its neighbours to turn vegan [Defender] {random war theatres : Engine chooses}
- Nationalism(Countries trying to prove their worth) [Attacker] {get to choose war theatres in phase 2}

Phase 2 (negotiations and preparation for war)

1. Alliances

- Alliances are made in this phase.
  (the alliances of the enemy are built-in depending on the dispute chosen)
    - Have 'country' be your ally but it will cost 15% of your economy
    - Have 'country' be your ally but it will cost 15% of your Population
    - Have 'country' be your ally but it will cost 5% of your Power

Towards the end of Phase 2 (after the alliances are made, if any. 'More alliances can be gained as the war progresses')

2. War Theatres

(the attacker has more power of choice)

(if a country only has soldiers than they can only fight on land, not space simply because they have not guardians)

- The war theatres are decided from either user input or by the engine.
- Max Number of war theatres = (number of countries)*4.
  (we can decide when war theatres are initialised during the coding phase.

i.e. the war theatres might not all be created at the start of the application but as needed, more theatres can be created as the simulation goes on)

3. Users distributes recruite into the different military regiments.

4. Allow user to buy weapons or supply.

sample phase 2 interface:

Government action:
if(user is attacker in dispute) the user gets to choose which war theatres to attack)
a)Military actions (organise military, buy weapons, set traps if defending)
b)Manage supplies (buy supplies, feed population, use of transportation)
c)Review country (view county strength, view population size, death toll (0 from the start), view military(view all divisions, weapons, and levels))
d)Find more allies
e)Surrender (end war and lose?)
f)Done

Phase 3: The Event Loop Starts (war!!!)

(if in real mode, only AI entities are simulated. In design mode the human user interacts with the AI entities(countries))

- This is the process of war. Based on the initial preparations for war from phase 1 and 2 (or decisions from the previous war loop iteration), the **engine runs**.
- After the engine runs, a **report is given** about what happened.
- Further, the user is able to implement a strategy going forward by making use of **decisions**.
- The user then acts by either **defending or attacking**. This process repeats itself until the war comes to completion.

(The simulation is player turned based. Each country including the user receives a turn.)

```
public void warLoop () {
    while (dispute.active()) {
           EngineSimulation()
           printEngineReport();
           makeDecision(user input);
           act(this); //defend or attack
    }
}
private void EngineSimulation(){
    for(country c : countries){
        c.makeDecision(); //engine strategically choses a decision
        act(c); //defend or attack
    }
}
private void act(Country c){
    if(c.isAtacker()){
        c.attack();
    }
    else{
        c.defend();
    }
}
```

Decisions:

1. Change war theatre
2. Destroy Transportation line
3. Increase allies
4. Buy Weapons
5. Set traps
6. Surrender

Final Phase

Report of War, and announcement of victor

Text Based Interface:

rough idea:

- ***WELCOME TO WAR ENGINE***
- ***Please enter the name of the country you wish to operate: <user input>***
- ***What does your country prioritise? Capitalism or Socialism: <user input>***
- ***These are the countries in your world:***
- ***<list 8 countries (with power levels)>***
- ***Phase 1 begins. <some random story>***
- ***Phase 2 begins.***

- ***Phase 3***
- ***Phase 4 (report of war)***

GUI Based interface (Design mode)

- Button to pause the simulation.
- Dropdown menus to swap out states and strategies.

2.3 Design Patterns

1. Prototype

- AbstractCountry
- Country

2. Iterator

- Iterator
- CountryIterator

3. Memento

- CountryMemento
- CountryHistory

4. Decorator

- War Theatre
- Battle Ground
  - Land
  - Air
  - Sea
  - Space
- Trap
  - Trenches
  - barricades
  - SpaceMagnets
  - Mines

5. Strategy

- TransportContext
- Transport
  - WeaponTransporter
  - AttackVessel
  - SuppliesTransporter

6. Template Method

- Registry
    - AllyRegistry
    - BattleRegistry

7. Factory

[WEAPON FACTORY]

- AttackWeaponCreator
    - NuclearWeaponCreator
    - ExplosiveWeaponCreator
    - MeleeWeaponCreator
    - RangedWeaponCreator
- AttackWeapon
    - NuclearWeapons
    - ExplosiveWeapons
    - MeleeWeapons
    - RangedWeapons

[VESSEL FACTORY]

- VesselWeaponCreator
    - TankCreator
    - ShipCreator
    - JetCreator
    - SpacecraftCreator
- VesselWeapons
    - Tanks
    - Ships
    - Jets
    - Spacecraft

8. State

- People
    - Citizens
    - Refugee
    - Recruits
        - Soldier
        - Pilot
        - Marine
        - Guardian
- RecruiterContext

9. Facade

- ArmoryFacade

10. Singleton

- WarEngine

Task 3: Implementation

- Please locate the System folder or view the implementation on github

Task 4: Report

4.1 Planning and Research

In war there are many factors that take place. War happens for a reason, this can be narrowed down to phases of war (Bloomfield), In the simulation 3 main phases will take place. Phase 1 will be where disputes between countries take place, Phase 2 the negotiations will go further and allies between countries will form, then finally Phase 3 is where the warfare will take place. For the purpose of the war engine in this simulation a country with the highest power will dominate and the implications after war will be seen as negligible as the engine will only produce an output of the war that took place. Who takes place in a war scenario? In the case of this project, the citizens will be acting as a way of increasing the population, while the refugees (from ally countries) will act as a way of decreasing the economy of countries. A war needs people to battle, in this project it is assumed that these people will be referred to as recruits and there are 5 different types of recruits. These recruits are as follows, Soldiers fight on Land, Pilots fight in the air, Marines fight in the sea, Guardians fight in space and finally Medics heal any other recruit in any war theatre (“Theater (warfare)”). Transportation is very important especially in the context of warfare (“Transport And Supply During The First World War”). Transport can be quite costly as this is the main mode for delivering supplies that are required by recruits. It will be further assumed that there will be 3 types of transport, Weapon transport, Vessel transport and Supplies transport. Weapon transport will enable recruits from a country to receive weapons from their homeland while fighting in other lands. Vessel transport is to transport Heavy duty weapons such as tanks to a war theatre. Supplies transport will transport food, medical supplies and so on in order to allow the recruits to have enough energy to heal faster.

4.2 Design Patterns

How we applied the design patterns to address the functionality required by the system

We make use of the Singleton design pattern to run the WarEngine which is the class containing the functions that start the war, initialises the set up for the war and simulates the war.

The Prototype Design Pattern is used on the Country class, which is our Concrete Prototype, to create multiple clones/copies of a Country object so that several countries are created and may then participate in the war. In the case of this system there is a limit of 8 countries that may go to war.

The Iterator pattern allows us to traverse through the countries stored in the WarEngine which is the aggregate where we will create the iterator and the CountryIterator class will be our Concrete Iterator that we can use to step through each country and return to the current country. It also checks if the end has been reached so as to not go out of bounds.

The Memento Pattern can be used to create a memento of a country using the CountryMemento class and then storing the state of this country at the time the memento was created in the CountryHistory class which will be the caretaker.

The Decorator Pattern is used to decorate the battlefields by allowing us to place different types of traps on the WarTheatres such as Trenches, Barricades, SpaceMagnets and Mines. The traps class is our decoration and the different kinds of traps will be our Concrete Decorators that we implement. The WarTheatres class will be our component . The BattleGround class that inherits from WarTheatres will have the derived classes Land, Air, Sea and Space that are the Concrete Components we decorate with the traps.

The Template method design pattern is put to use with the Registry class(Abstract class) and its derived classes AllyRegistry and BattleRegistry, our Concrete Classes. The addRecord method and getRecords method will be the template methods which are redefined in our Concrete Classes. The AllyRegistry class is where we store an unordered pair of vectors containing pairs of countries that are allies and the same applies to the BattleRegistry where instead it stores pairs of countries that are at battle with each other(enemies).

The Factory Method is applied by the AttackWeaponCreator class and the VesselWeaponCreator class which are 2 factory methods used to create weapons and act as our Creators. The Concrete Creators for AttackWeaponCreator are NuclearWeaponCreator, ExplosiveWeaponCreator, MeleeWeaponCreator and RangedWeaponCreator. This will allow us to produce weapons from these factories. The AttackWeapon(Product) class has the derived classes NuclearWeapons, ExplosiveWeapons, MeleeWeapons, RangedWeapons that are Concrete Products that can be produced by our factories.

The Concrete Creators for VesselWeaponCreator are TankCreator, ShipCreator, JetCreator and SpacecraftCreator. This will allow us to produce vessels from these factories. The VesselWeapons(Product) class has the derived classes Tanks, Ships, Jets and Spacecraft are Concrete Products that can be produced by our factories.

The Strategy pattern takes place by the Transport class with Transport as our Strategy, TransportContext as our Context for interchanging between the different Concrete Strategies which are WeaponTransport and AttackVessel. These strategies are changed depending on what the user wants to purchase for their recruits, weapons or vessels. The recruits are passed by reference through purchase that will call one of the factories depending on the user choice and produce their weapons or vessels.

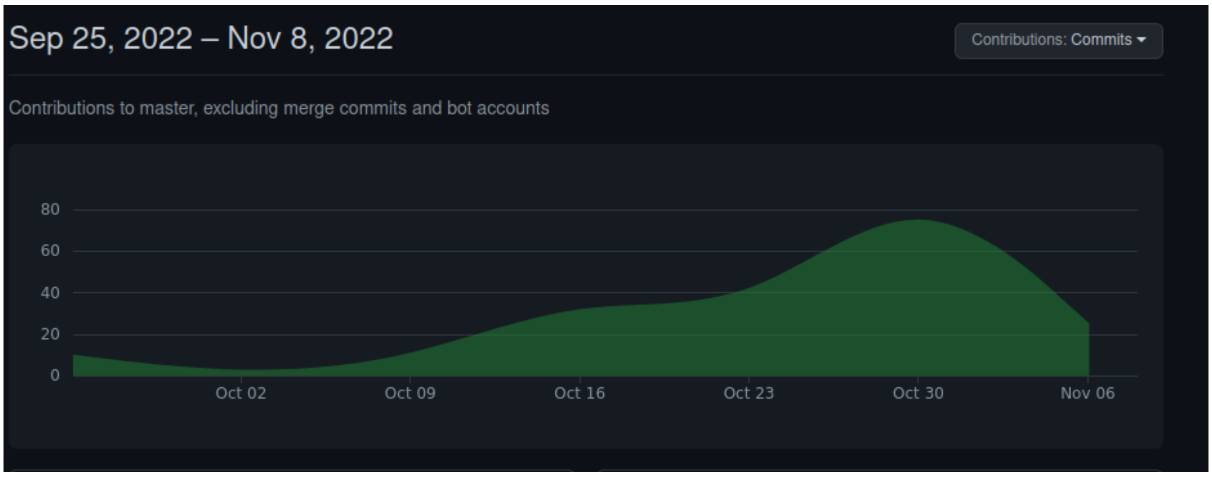
The State pattern consists of the People class which is our State and its derived classes Recruits, Citizens and Refugees which are the Concrete States and can be swapped depending on the RecruiterContext which will be the Context. The Recruits class has derived classes for the different military recruits, namely Soldier, Pilot, Marine, Guardian and Medic.

A Facade design pattern bridges our Country, RecruiterContext and TransportContext so that weapons or vessels may be bought for recruits through the TransportContext class by choosing the recruits through the RecruiterContext(swapping it to Recruits) and passing the recruit by reference in the purchase function where in the WeaponTransport or AttackVessel classes the appropriate strategy will be chosen depending on what needs to be bought for the country containing these recruits and weapons.

Task 5: Development Practices

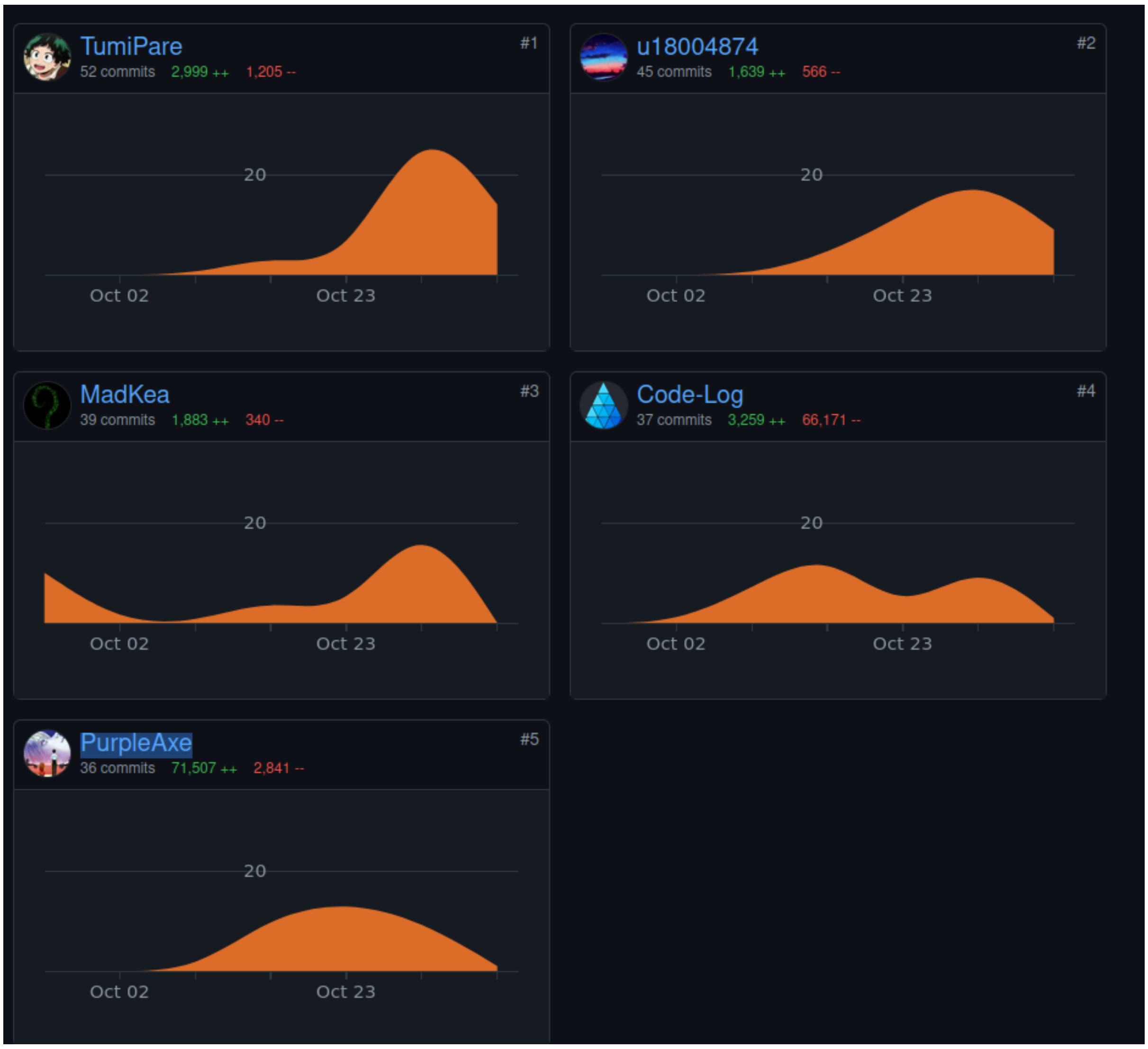
Version Control System (GIT)

Used GitHub to maintain a Repository accessible to everyone involved in the project. GitHub Actions were used to ensure no accidental changes were made to the main branch unnecessarily.



Sep 25, 2022 – Nov 8, 2022

Contributions to master, excluding merge commits and bot accounts



TumiPare #1
52 commits 2,999 ++ 1,205 --

u18004874 #2
45 commits 1,639 ++ 566 --

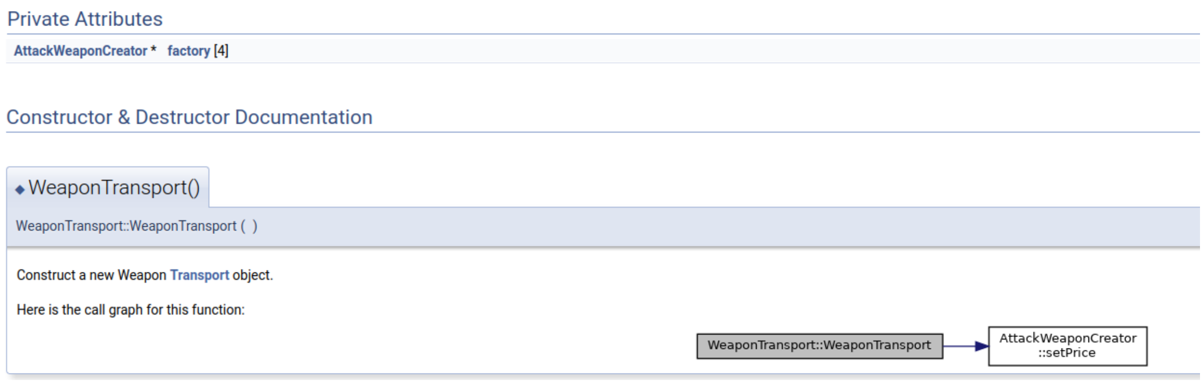
MadKea #3
39 commits 1,883 ++ 340 --

Code-Log #4
37 commits 3,259 ++ 66,171 --

PurpleAxe #5
36 commits 71,507 ++ 2,841 --

Doxygen

Used to document code. This was mainly used to allow other developers to follow along what functions are doing what. This allowed for Contract first design of functions. Below is an example of the doxygen generated.



Private Attributes

AttackWeaponCreator * factory [4]

Constructor & Destructor Documentation

◆ WeaponTransport()

WeaponTransport::WeaponTransport ( )

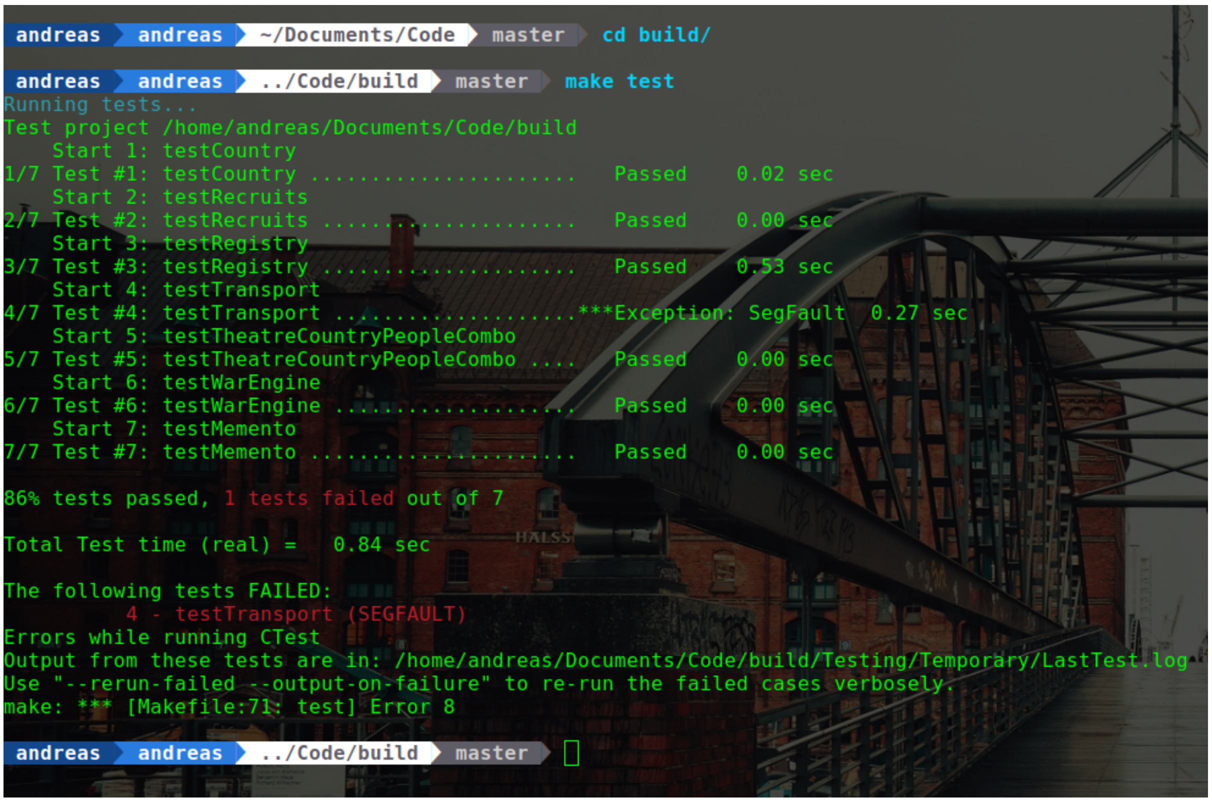
Construct a new Weapon Transport object.

Here is the call graph for this function:

WeaponTransport::WeaponTransport
AttackWeaponCreator ::setPrice

Unit Testing

CTest was used as the automated testing framework for this project.
As seen below tests are able to pass and fail.



```
 andreas  andreas  ~/Documents/Code  master  cd build/
 andreas  andreas  ../Code/build  master  make test
Running tests...
Test project /home/andreas/Documents/Code/build
    Start 1: testCountry
1/7 Test #1: testCountry ......................   Passed    0.02 sec
    Start 2: testRecruits
2/7 Test #2: testRecruits .....................   Passed    0.00 sec
    Start 3: testRegistry
3/7 Test #3: testRegistry .....................   Passed    0.53 sec
    Start 4: testTransport
4/7 Test #4: testTransport ....................***Exception: SegFault  0.27 sec
    Start 5: testTheatreCountryPeopleCombo
5/7 Test #5: testTheatreCountryPeopleCombo ....   Passed    0.00 sec
    Start 6: testWarEngine
6/7 Test #6: testWarEngine ....................   Passed    0.00 sec
    Start 7: testMemento
7/7 Test #7: testMemento ......................   Passed    0.00 sec

86% tests passed, 1 tests failed out of 7

Total Test time (real) =   0.84 sec

The following tests FAILED:
          4 - testTransport (SEGFAULT)
Errors while running CTest
Output from these tests are in: /home/andreas/Documents/Code/build/Testing/Temporary/LastTest.log
Use "--rerun-failed --output-on-failure" to re-run the failed cases verbosely.
make: *** [Makefile:71: test] Error 8
 andreas  andreas  ../Code/build  master
```

Contributions

Amber-Leigh Lezar (u18004874)

- Lead the integration of design patterns.
- Design, planning and research.
- Implemented code..

Keabetswe Madumo (u20438614)

- Lead unit testing.
- Design, planning and research.
- Implemented code..

Jaco Malan (u20416076)

- Maintained git repository and code cleanup.
- Design, planning and research.
- Implemented code..

Tumi Pare (u21452832)

- Lead final development of war engine..
- Design, planning and research.
- Implemented code..

Andreas Visagie (Group Leader) (u21430901)

- Lead overall operations of the project.
- Design, planning and research.
- Implemented code.

References

Bloomfield, Lincoln P. “warend2.” *Massachusetts Institute of Technology*, http://web.mit.edu/cascon/warend.html. Accessed 25 October 2022.

“First, Second, and Third World.” *Nations Online Project*, https://www.nationsonline.org/oneworld/third_world_countries.htm. Accessed 25 October 2022.

Siemienowicz, Kazimierz, and Ernest Brooks. “Artillery.” *Wikipedia*, https://en.wikipedia.org/wiki/Artillery. Accessed 20 October 2022.

“Theater (warfare).” *Wikipedia*, https://en.wikipedia.org/wiki/Theater_(warfare). Accessed 25 October 2022.

“Transport And Supply During The First World War.” *Imperial War Museums*, https://www.iwm.org.uk/history/transport-and-supply-during-the-first-world-war. Accessed 25 October 2022.

“Weapon.” *Wikipedia*, https://en.wikipedia.org/wiki/Weapon#By_function. Accessed 30 October 2022.