

Practical 2

Janco Spies, u21434159

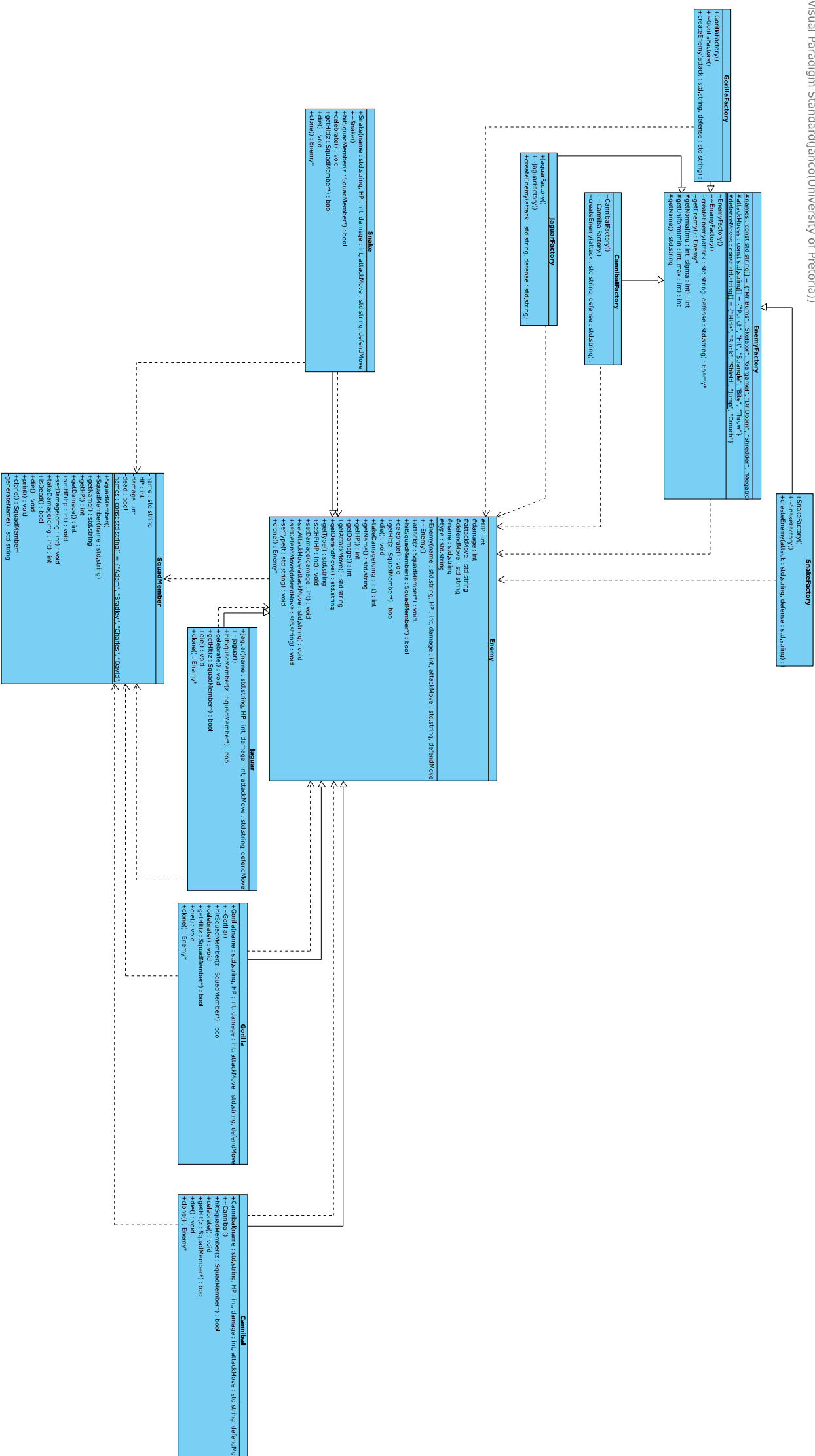
August 15, 2022

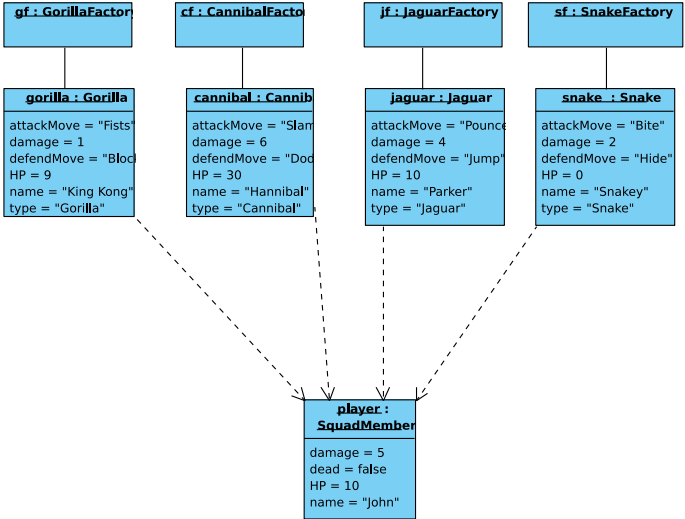
Task 1

1.3 The Template Method pattern has been implemented.

Task 2

2.4





2.6 The Gang of Four Factory Method pattern is realized in my implementation with the following participants:

Creator: EnemyFactory

ConcreteCreator: CannibalFactory, JaguarFactory, SnakeFactory and GorillaFactory

Product: Enemy

ConcreteProduct: Cannibal, Jaguar, Snake and Gorilla

The Gang of Four Factory Method pattern is realized in my implementation with the following methods:

anOperation(): getEnemy()

factoryMethod(): createEnemy()

2.7 The Factory Method design pattern together with the Template Method design pattern has been implemented.

Task 4

4.2 a) The first way I expanded my game was by adding a checkpoint save system which allows the player to save the game at any time and then restore the most recently saved checkpoint. The functions of the Memento pattern are :

getState(): State is set in the CheckPoint() parameterized constructor.

setState(): State is restored using getter methods in the CheckPoint class.

The Game class acts as the Originator, the CheckPoint class acts as the Memento and the Backup class acts as the Caretaker. State is defined as the SquadMember* array storing the names, HP and damage of each squad member as well as the current score.

b) The second way I expanded my game was by adding an undo move system which allows the player to undo the last move. The functions of the Memento pattern are :

getState(): State is set in the Move() parameterized constructor.

setState(): State is restored using getter methods in the Move class.

The Game class acts as the Originator, with the Move class acting as the Memento and the MoveHistory class acting as the caretaker. State is defined as the SquadMember* array storing the names, HP and damage of each squad member as well as the current score and current enemy being faced in the encounter.