OOP Analysis

Answer the following two questions:
• How does your design implement the four pillars of OOP (abstraction, encapsulation, inheritance and composition, and polymorphism).

Our design captures the 4 pillars of OOP though
Abstraction: we use abstraction though having methods not needing to call the full calculation/setting for each method and having methods that can hide the details from the user

Encapsulation: not needing to show calculations of how damage/healing is calculated to the user. Having the raw data hidden from the user and only shown when requested.

Inheritance: though the use of having the classes based off of a abstract class that can hold all of the universal methods and attributes for the sub-classes

Polymorphism: when we use the attack/ability methods for each class these methods do different things while being under the same method.

•Why are your entity classes good abstractions (i.e., models) of the real-world entities?

Ours are good abstractions because they can do virtually what we want them to do. They can hold details that can be affected by events inside the script. Our abstractions could be used inside of a very basic game because they have a way of calculating and dealing damage to other objects inside of its self.