Brief for turn based battle game framework

# Abstract

You have been engaged to construct a framework/engine for turn based, party-centric battle games. A sample implementation should be included to help client developers understand the framework.

# Functional Requirements

1. A system for two or more ‘parties’ of at least one character to engage in some form of turn based competition.
2. Each party member takes a turn in the competition, during which they can perform set actions.
3. Party members have their own individual attributes that influence their performance and possibly the turn order.
4. An interface for at least one player to interact with the system, controlling one or more party members, allowing for action selection, result output and display of game state.
5. A method of storing game state and predetermined gameplay variables.
6. A method of persisting game state if program is ended and allowing the player to continue from that state when the program is run again.

# Non-functional Requirements

1. Portability
2. Extensibility
3. Interoperability

# Architectural Use Cases

* Asynchronous rock, paper, scissors game:
  + Developer wants to create a game for two people to play the rock, paper, scissors game together on the same computer, taking turns to enter their choice and then evaluating a winner as though the choices had taken place at the same time.
* Monster battle game (in the style of Pokemon):
  + Developer wants to create a game where the player controls a group of monsters in battles with other groups of monsters. Each monster has a number of moves they can perform, but each move can only be performed a certain number of times during a single battle. Each turn, the player can choose to switch out their monster or make a move. The game is won when one player’s monsters are all incapacitated.
* Digital implementation of trading card type games:
  + Developer wants a game where players can take turns placing cards on a virtual table, similar to games like Magic: the Gathering. Cards can have various effects on either player, such as increasing or decreasing their ‘life’ score, allowing them to draw extra cards from a deck, etc.