Comments about the Game

The spawn rates for Fish and Reeds were far too much, so they were each reduced to a 10th of their original rate

No baby dinosaurs have been implemented. The win condition is met when a T-Rex is hatched

Recommendations for changes to the engine code

Naming a Map

- When a map is parsed into a method, there is currently no way of understanding what map specifically it is. Being able to give a map a name, via a string or an enum would open up new options that might allow for further functionality, where a condition may run depending on what map the player is on, eg:
    - A map might have a high altitude, and moving might slowly reduce the player's HP while in this map.
    - Actors would understand what map they are on

Removing the Player stops the game loop, but finishes the current turn

- An ideal world would be to end the game loop by removing the player. However, when the player is removed, the turn is still processed, where it tries to process the players turn (who by this point no longer exists). This throws a NullPointerException, as it tries to continue to process the turn. The actions menu is still shown even once the game loop is finished.
    - Fixing this could be achieved by exiting the game after the game loop is stopped, by using something such as a System.exit(0)

A function to combine GameMap.IsAnActorAt() and GameMap.GetActorAt()

- A function to combine the GameMaps IsAnActorAt() and GetActorAt() to return an actor at a surrounding tile if one is present would have made experiences such as tagging much easier
    - The solution is to create a method that returns an Actor in the location

A function to find the closest object of a specific type

- Would have proved extremely useful in implementing behaviour for AI. Instead of manually checking through tiles and then sorting a list.
- The solution would be to add a method that returns the closest object of that type. Similar to .IsAnActorAt()

Praises of the engine code

Ease of use of interfaces

- The use of interfaces to enable easy casting of its subclasses made coding much more readable and concise. Casting became extremely important for some cases including Actors to Dinos, Actors to Players and Items to SaleItems.

Creating Exits between maps gave unintended benefits

- Using the addExit() function to allow for the player to traverse between the two maps was simple, and gave unintended (but positive) effects. Allowing for trees and other actors to move between the maps.

Exits in general

- Having a function for getExits() was extremely useful in general to find where the actors could travel and was extremely useful in not breaking the game by trying to do things outside of the map.
- Before I used getExits I had some long checking codes to not let trees spawn outside of the map but using getExits made this process much easier

The use of Skills made bool checking simple

- Using Skills for bool checks made it simple, as almost everything could have a skill. This included the Ground having a skill, and then being used to grant actors skills, or used to check Exits, and to enable / disable certain actions

Actions were extremely easy to use

- They were implemented so easily that I still don't know exactly how they execute from the player's choice