**Command line Tamagotchi - Brief**

**Overall**

Build a command line based, object oriented, virtual pet in Java.

**User interactions**

* Name it
  + Users should be able to name their pet when it is born.
* Feed it
  + Unless the pet is fed at regular intervals it will die. Pet can go for longer without food when asleep at night time and feeding is not required when pet is away at boarding house. Too much food will cause pet to get sick. Pet will not eat when sad.
* Play with it
  + Pets need to be played with to keep them happy. Playing takes the form of a guessing game of ‘higher or lower’, where the pet tells you a number and you must guess if the number the pet is thinking of is higher or lower than it. Numbers cannot be higher than 9 or lower than 1. How well you do out of five games determines how much the pet’s happiness bar is increased.
* Clean up waste
  + Pet will randomly produce waste which needs to be cleaned up. Waste is not produced if the pet is asleep or away at the boarding house. If waste is left for too long, the pet will become sick.
* Cure it (if sick from waste or overfeeding)
  + Pet can be cured from sickness using this action. If not cured within a short time frame (tbd), pet will die.
* Check on it
  + Check on stats of hunger and happiness to see when it will need feeding or playing with
* Send it to boarding house (for weekends/holidays)
  + When leaving the program for a long time, the pet can be sent to the boarding house to make sure it does not die when you are away. Time spent here will not contribute to the final score, just pauses the game.

**Pet actions**

* Hatch
  + Starts as an egg, will hatch within first 15 minutes of play
* Eat
  + Pet needs to eat at regular intervals while not asleep to stay alive. Eating refills hunger bar. Overeating causes sickness.
* Sleep
  + Pet will sleep between set hours overnight. While asleep, it does not need playing with or feeding and cannot get sick. If pet is sick when it goes to sleep, sickness timer is paused until it wakes up.
* Play
  + Pet has a happiness bar which slowly empties, if it gets too low pet will not eat. Playing takes the form of a guessing game of higher or lower described above. Happiness bar is increased based on success in game.
* Get sick (if waste is not cleaned up in time or overfed)
  + When fed, the hunger bar will increase up to a maximum amount. If pet is fed over this limit by a full meal, it will become sick.
  + If waste is left for too long (time tbd), pet will become sick
  + Sick pet will not eat or play until cured (by cure action). If sick for too long (time tbd), pet will die
* Cry (if not played with enough or hungry)
  + Pet will cry when sad or hungry which sends message to user that pet needs attention. Sad pets will not eat.
* Die
  + If left sick or hungry for too long pet will die. Game is over and stats about how long pet lived for are displayed
* Grow (3 stages)
  + If pet is kept alive long enough it will grow up. Time is not counted if pet is at boarding house.
  + Pet starts as a baby, then progresses to a child, then finally to an adult. Adults require more food and playing with.

**End goal**

The goal of this game is to keep the virtual pet alive for as long as possible.

There are two ways the pet can die, lack of food or sickness. To keep a pet alive for as long as possible, it must be fed and played with at regular intervals.

**Differences from original Tamagotchi**

The original Tamagotchi had no pause button, but we will add one in the form of a boarding house. This is added because as a desktop based game, it is not portable and would die over the first weekend. As the intended user is sitting at a desk, the pets waking hours will be between 9am and 5pm to ensure it can’t die unfairly in the evening, but to make up for this concession it will be more demanding during daytime.

The original instructions for a Tamagotchi are found [here](http://members.tripod.com/~Tamagotchi_Central/instructions.html). These include details about extra features like ‘discipline’ and different outcomes for your pet depending on how you treat it. We are leaving these features out of this version.

Instead of a graphical output, we will be using the command line. Instead of seeing objects on the screen (like waste), users will have to look at the most recent updates or check on the pets stats manually to find out what it needs. Timed events like warnings of hunger will be printed to the screen and will not be overwritten by the next update so users have a log of what has been going on.