

# Space Bunnies Reflection

In completing this project, I generally met my expectations and stated goals, especially with regards to the user interface, which I hadn't planned out very well. However, my initial ambitions were still a little underestimated: after creating the underlying objects and implementing basic game functionality, I had nearly surpassed the code line limit. Many of these lines serve to polish the user interface as I didn't feel comfortable "releasing" a functioning program with an unusable user interface. In addition to polishing the UI, many lines of code were probably wasted by implementing heavy functional programming in the Game object. Having now written the program, I believe I could more efficiently design the objects, methods, and attributes used. For example, I like the way I subclassed the Location class, implemented the timestep() method, and used the inhabitants attributes to manage objects. However, using a single inhabitants attribute to hold all types of objects resulted in deeply nested for-if statements within the Game object.

It was interesting to see my code evolve as it became larger. I initially used `__str__` methods in the objects to display information to the user, however, the output soon became unusably long when looped over a large number of objects (such as viewing all the bunnies on a planet). To avoid this, I abandoned the use of the `print(object)` functionality and expanded the number of attributes defined in each class so I could create neatly formatted tables by referencing the appropriate attributes. I also noticed that few of my methods actually return a value up the stack. Had I more concretely planned out the classes and functionality of the game, I believe I could have moved code from the Game class to the Location subclasses and Bunny class, where the code would've easier to reuse.

If I were to continue developing this game, I would add some complexity in the user decisions. For example, a user cannot currently order "x" male bunnies and "x" female bunnies to board a spaceship, nor can they specify an individual bunny. I would also add functionality for spaceships to be destroyed by unlikely, in-flight, accidents and would consider the ability for the user to direct the creation of additional ships, ship upgrades, etc. (at the cost of bunny time/reproduction). Furthermore, the game parameters (defined in the `run_default()` method) should be adjusted so that the map is more interesting and strategic.

## Notes

- Some functionality of the code requires you to finish the game. To reduce the length of the game you can pass an optional argument "maxiterations=" to the game object in the `start()` function.
- After starting the game (Entering 1 in the Main Menu), holding down "Enter" will result in random choices being made. The game can easily be completed this way, however, scoring points using this method usually takes about 150 iterations.
- The code will create a .txt file in the current working directory if you finish the game. Scores are stored here to enable high score tracking functionality.
- For replay, testing, and debugging purposes, the `start()` function has an optional argument "fast". Setting fast to "True" will disable the slow display of the intro text.

- Most planets only have one spaceship “dock,” while a couple have two. If a ship reaches a planet it will continue orbiting that planet and will not return to the other end of the route until it lands and takes off again.
- Bunnies can exceed the planets population limit. If the population is under the limit, an entire ship will be allowed to unload, even if the final number exceeds the planet capacity. Similarly, if the planet’s population is under the limit, all bunnies will be given a chance to reproduce during the next time step, even if the new population exceeds the planet capacity. If the capacity is met or exceeded, then the bunnies will not be able to disembark ships or reproduce on that planet during the next time step.
- Boarding all your bunnies early on may really hurt your reproduction rate, especially if the gender ratios don’t work out in your favor. It might pay off to wait an extra time step before loading.
- Ordering your fat or skinny bunnies to a new planet could have a strategic benefit. Baby bunnies’ mass is determined by a combination of their parents’ mass, their parents’ genetic strength, and random chance. Starting a new planet with fat bunnies could make for a really good gene pool, but might significantly reduce the quality of the origin planet’s gene pool.
- Score is calculated as a simple sum of the total mass of Bunnies arrived to earth.
- Your lieutenants may ask if you want to board bunnies from Earth onto a spaceship. They are not aware that Earth is the final destination for all bunny arrivals. Similarly, the Planet Stats table will show Earth’s Bunny Population as  $\#/0$ . This numerator is the number of bunnies that have arrived. There is a limitless capacity for more bunnies on Earth.
- Bunnies on Earth or in a spaceship are not able to reproduce.