

# Proposal for Python Tamagotchi Game Project

## **Team 3 Member:**

Haelee Kim, Jiwoo Suh, Kyuri Kim, Nayaeeun Kwon

## **Motivation and Background:**

Our team is thrilled to embark on the development of a Tamagotchi game for our computer science foundation project. Tamagotchi, a virtual pet simulation that gained immense popularity in the late '90s, has left an indelible mark on the gaming world. Its enduring charm and simplicity continue to resonate with enthusiasts of all ages. Choosing to recreate this classic in Python serves a dual purpose for our team. Not only does it promise a delightful and engaging project, but it also provides us with a unique opportunity to delve deep into fundamental aspects of computer science, particularly object-oriented programming (OOP) concepts and the application of basic Python methods and functions.

By revitalizing the Tamagotchi experience in a modern Python context, we aim to tap into the nostalgia associated with the original while introducing a contemporary twist. This project aligns perfectly with our passion for combining classic gaming elements with cutting-edge programming techniques.

## **Project Description:**

Our Tamagotchi game will immerse players in the joy of adopting, caring for, and witnessing the growth of their virtual pets. These pets will exhibit various needs such as feeding, playing, and resting, requiring users to actively participate in fulfilling these requirements to ensure their pet's well-being. The implementation will heavily leverage OOP principles, with distinct classes representing the pet and its diverse attributes. This approach not only reinforces our understanding of OOP but also lays the foundation for a scalable and modular game architecture.

To enhance accessibility for players of all levels, we plan to implement a simple yet engaging text-based interface. This decision aligns with our goal of making the game easy to understand while providing a platform for users to experience the timeless joy of Tamagotchi.

## **Scope and Timeline:**

Week 1 (November 7): Project setup and brainstorming.

Week 2 (November 14): Defining class structures and implementing basic functionality, such as feeding, playing, and resting.

Week 3 (November 21): Adding more advanced features like pet growth, happiness levels, and random events.

Week 4 (November 28): Testing, debugging, and refining user interface and finalizing the project, preparing documentation, and submitting.

**Git repository:** [https://github.com/Nayaeeun/23Fall\\_CSFoundation](https://github.com/Nayaeeun/23Fall_CSFoundation)

**Team Meetings:** Weekly meetings every Tuesday for progress updates and discussion via WhatsApp.

**Communication:** WhatsApp will serve as our primary communication channel for discussions, quick updates, and problem-solving.