AUD722 Creature Project - App Summary

What it is

A SuperCollider class library for the AUD722 Computer Music course that lets you build autonomous sonic creatures from audio buffers and behaviors. It also provides an environment to run groups of creatures through timed state cycles.

Who its for

AUD722 Computer Music students and SuperCollider users who want to design and run creature-like sound agents.

What it does

- Defines a Creature base class with default actions and action dispatch.
- Auto-loads per-creature audio buffers from the CreatureAudioFiles folder on server boot.
- Supports adding or overriding actions via addActions or subclass methods.
- Provides the EvoLab environment to cycle states and durations via patterns.
- Loads shared synthdefs from CreatureSynthDefs on server boot.
- Includes example creature subclasses and usage examples in the repo.
- Offers lifecycle control for synths and tasks via add, substitute, and release.

How it works (architecture)

- Components: Creature (base class), EvoLab (environment), CreatureSynthDefs (synthdef loader), CreatureAudioFiles (audio assets).
- Startup flow: on server boot, Creature adds synthdefs and loads buffers for all Creature subclasses.
- Runtime flow: EvoLab.play builds a pattern over states and durations, then calls each creature’s performAction for the current state.
- Action execution: performAction runs a matching method or an action function that creates synths or tasks using buffers and synthdefs.

How to run (minimal)

- Clone the repo and place the AUD722_CreatureProject folder in your SuperCollider Extensions directory.
- Recompile the SuperCollider class library.
- Boot the server in SuperCollider: s.boot.
- Define the creature classes and the state-duration list.
- Start the simulation with EvoLab.play(...); stop with EvoLab.release(...).