Evolving robots with the Revolve framework

Aart Stuurman
Computational Intelligence Group
Content

- What is Revolve?
  - Hardware: the physical part
  - Software: simulation and optimization

- Example research projects
  - The Effects of Learning in Morphologically Evolving Robot Systems
  - Robots interacting and evolving together in simulation
  - The Influence of Robot Traits and Evolutionary Dynamics on the Reality Gap

- What can you do
  - E.g. use your own novel algorithm

- Experience building software side Revolve
What is Revolve?
What is Revolve?

- Core
- Brick
- Active hinge
What is Revolve?

- Research Evolutionary Algorithms
- Brain usually vector based
  - Differential evolution
  - Bayesian optimisation
- Body currently not
  - Tree-based (direct encoding)
  - CPPN (compositional pattern producing network)
Example projects

- *The Effects of Learning in Morphologically Evolving Robot Systems*, J. Luo, A. C. Stuurman
- *Robots interacting and evolving together in simulation*, M. de Carlo
- *The Influence of Robot Traits and Evolutionary Dynamics on the Reality Gap*, F. van Diggelen
The Effects of Learning in Morphologically Evolving Robot Systems

- By Jie Luo
- youtu.be/UYClGJdRhg4?t=9
Robots interacting and evolving together in simulation

- By Matteo de Carlo
- [youtu.be/uzUTNuJEg9w?t=207](youtu.be/uzUTNuJEg9w?t=207)
The Influence of Robot Traits and Evolutionary Dynamics on the Reality Gap

- By Fuda van Diggelen

**Figure 2:** Test suite of six robots as rendered in the simulator.

**Figure 3:** Test suite of six robots in the real world.
What can you do with Revolve?

- [github.com/ci-group/revolve2](https://github.com/ci-group/revolve2)
- Easy access to a difficult black box problem
- Test your own (evolutionary) optimization algorithm
  - Either for brain,
  - Or body
- Compare with physical robot
Building scientific software

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tables (14)</td>
<td></td>
<td>CREATE TABLE environment_name</td>
</tr>
<tr>
<td>environment_name</td>
<td></td>
<td>CREATE TABLE environment_names</td>
</tr>
<tr>
<td>environment_names</td>
<td></td>
<td>CREATE TABLE environment_names_item</td>
</tr>
<tr>
<td>genotype_with_meta</td>
<td></td>
<td>CREATE TABLE genotype_with_meta</td>
</tr>
<tr>
<td>measures</td>
<td></td>
<td>CREATE TABLE measures (id, name, value, unit)</td>
</tr>
<tr>
<td>parameters</td>
<td></td>
<td>CREATE TABLE parameters</td>
</tr>
<tr>
<td>parameters_item</td>
<td></td>
<td>CREATE TABLE parameters_item</td>
</tr>
<tr>
<td>population</td>
<td></td>
<td>CREATE TABLE population</td>
</tr>
<tr>
<td>population_individual</td>
<td></td>
<td>CREATE TABLE population_individual</td>
</tr>
<tr>
<td>program_root</td>
<td></td>
<td>CREATE TABLE program_root</td>
</tr>
</tbody>
</table>

```python
from .__serializable import Serializable

t = TypeVar("T", bound=Union[int, float, str, Serializable])

class SerializableList(List[T], Serializable):
    """A python list that can be serialized to the database."""
    _db_base: type = TODO proper type
    _item_type: Type[T]
    _is_basic_type: bool
    item_table: Any

@classmethod
def __init_subclass__(cls, /, table_name: str, value_column_name: str, **kwargs: Dict[str, Any]) -> None:
    ...
    initialize this object.
    ...
    super().__init_subclass__(**kwargs)

assert len(cls.__orig_bases__) == 1, "# type: ignore # TODO
cls._item_type = get_args(cls.__orig_bases__[0]) # type: ignore # TODO
dbtype: Union[Type[Integer], Type[Float], Type[String]]

if cls._item_type == int:
    cls._is_basic_type = True
dbtype = Integer
```
Building scientific software

- Get a dedicated software engineer
  - Domain knowledge is not so relevant
  - Make sure they want to build software
Building scientific software

- Get a dedicated software engineer
  - Domain knowledge is not so relevant
  - Make sure they want to build software
- Spend time on software, not research
Building scientific software

- Get a dedicated software engineer
  - Domain knowledge is not so relevant
  - Make sure they want to build software
- Spend time on software, not research
- Share within your group
Contact me

- Email a.stuurman@vu.nl
- github.com/ci-group/revolve2
- About using Revolve,
- And about building your own software platform