

Overview

Introduction

Purpose

Purpose of GUnit

Target Audiences

Features and Design

Software Engineering

Impact

Demo

Astrophysical Multipurpose Software Environment (AMUSE)

- ▶ Start with modules, move upwards -> MPI -> AMUSE python code .
- ▶ Discuss how our architecture builds on AMUSE
 - ▶ Uses AMUSE units framework
 - ▶ Created wrappers for modules to add functionality.

State of AMUSE

- ▶ Partnership between Drexel and the Leiden Observatory in the Netherlands, sponsored by NOVA.
- ▶ NOVA = Netherlands Research School for Astronomy
- ▶ Mention large scale again
- ▶ Written by hand = hard to share
- ▶ Waste of work to replicate someone else's diagnostics to fit your exact circumstances.

Purpose of GPUUnit

- ▶ Ease the creation, execution, and analysis of experiments with AMUSE
- ▶ Create experiments with minimal to no programming
- ▶ Repeatability
- ▶ Sharing Experiments
- ▶ API for results / diagnostics

Target Audiences

- ▶ Physics Student
 - ▶ A user with minimal to no programming experience and minimal knowledge of astronomy
 - ▶ Has an interest in doing simulations to learn and understand basic concepts in astronomy
- ▶ Observational Astrophysicists
 - ▶ Little programming experience
 - ▶ High understanding of astronomy
 - ▶ Interested in simulating specific cluster of stars to compare to real data
- ▶ Theoretical Astrophysicists
 - ▶ Large programming experience
 - ▶ High understanding of astronomy
 - ▶ Interested in creating own simulation with custom diagnostics

Features

- ▶ Explain how features satisfy requirements.
- ▶ Configurable experiments -> less programming.
- ▶ Diagnostics -> common API for metrics
- ▶ Code is generated to run actual experiment -> advanced users can tweak it
- ▶ Storage of state -> repeat experiment if it crashes

Design

- ▶ Expand bullets.
- ▶ Challenges:
 - ▶ Figuring out AMUSE
 - ▶ Making a useful tool that simplified experiment creation without taking away any of AMUSE's power/features.
 - ▶ Allow future developers to expand on this work:
 - ▶ Modular diagnostics w/API to do the work
 - ▶ Experiment storage abstraction: allows for remote backup, sharing of results

Tests

- ▶ Table of tests that pass.

User Testing

- ▶ Tested with customers (Steve/Tim)

Project Plan

- ▶ AMUSE codebase is large and complex (as we have mentioned)
- ▶ Before we could plan our project we needed to figure out how AMUSE worked.
- ▶ Learning continued throughout the project.

Team Management

- ▶ Bi-weekly team meetings helped get a lot of work done
- ▶ Able to code and discuss at the same time in person (useful)

Potential Benefits

Demo

- ▶ Demonstration of a simulation.

Questions

- ▶ Questions?