

**Fakultät für Physik und Astronomie
Universität Heidelberg**

Masterarbeit in Physik
eingereicht von

Dane Späth

geboren in Blaubeuren, Deutschland

2019

Developing Software for the Waltz Telescope

Diese Masterarbeit wurde durchgeführt von Dane Späth an der
Landessternwarte Königstuhl in Heidelberg
unter der Betreuung von
PD Dr. Sabine Reffert

1 Introduction

- History of Waltz

Motivation

- Scientific goals
- Educational goals
- Difficulties with using multiple different (open source) software components
 - Cartes du ciel
 - Self wirtten scripts
 - PHD2
 - (Andor SDK)
- Goal:
 - Integrated solution
 - Tailored exactly for needs of Waltz Telescope and Hardware components
 - All in one program
 - Easy to use
 - Possible remote usage
 - Safety
 - Understandable for students

- Cleanly written, well structured and well documented for future adjustments (e.g. adding of dome, adding of guiding, change of components, change of GUI or terminal UI)

2 Basics

2.1 Choice of Python

- Easy to read
- most used in science (easy access for new students and other scientists with little knowledge of other languages)
- No need for fast calculations (possible biggest disadvantage of Python but solveable by using CPython built-ins or C)

2.2 Implications of using python

- One possible paradigm: Object Oriented Programming but also allows for functional programming
 - Leads to well structured code
 - We have real world objects (telescope, coordinates, targets, CCD images, cameras, mirrors) and OOP is best way to model real world things
 - Easy to apply widely known design patterns leading to easy to understand code (many different sources explaining ideas behind design, future readers will not depend on author of this thesis)
 - GUI interacts nicely with OOP (???)
 - Leads to modularity, using of Plug Ins to be able to not depend on components
 - Combine with functional programming for more logic based code

- Combine with Duck-typing
- Pythonic Conventions (PEP8) leads to nice to read code (realistically if everything works no one will want to look at everything to make small changes. So it is important that the program is cleanly structured and easy to read to make small changes)
- Also:
 - Dynamically, strongly typed
 - No interfaces
 - Multiple inheritance

3 BookContents

3.1 Moore18

([Moore, 2018](#))

Choosing Tkinter

- in Standard Library
- stable in future (will not change much)
- only GUI toolkit no additional stuff. Good for decoupling
- simple and straightforward but may look a bit old fashioned

Literaturverzeichnis

Moore, A. D. (2018). *Python GUI Programming with Tkinter*. Packt Publishing.