

OdyssPy

An interactive Odyssey narration, with some catches...

Learning Unit 3, Methods in CS: Analysis

Daniele Solombrino, 1743111

Introduction

OdyssPy is an **interactive story** realized in **Python** via **RenPy SDK**, which can be used by **1st year high school** teachers as a teaching supporting tool.

Interactiveness of OdyssPy promotes **in-class interactions** and **discussions** between students, offering an attractive and stimulating way of studying Homer's **The Odyssey**

Learning goals

1. Interdisciplinary

- 1.1. Consolidate knowledge about principal **characters** of the Odyssey
- 1.2. Consolidate knowledge about principal **events** of the Odyssey
- 1.3. Observe the **cause-effect** link between events (which may be a soft skill too, being not necessarily limited to the Odyssey only)

2. Computer Science

- 2.1. Adoption **computational thinking** in a real-world scenario
- 2.2. Consolidate programming logic flow logic (**jump** and **if-else** statements)
- 2.3. Consolidate simple usage of pre-made **Object**
- 2.4. Consolidate **String** and **boolean** manipulations

3. Soft skills

- 3.1. **Group** work and discussion
- 3.2. Visual **design**
- 3.3. **Creativity** (measured according to how students develop the story)

Learning goals interaction and challenges

Every learning goal is tied to the other ones, creating multiple constructive challenges for the students.

Computational thinking (learning goal 2.1) is the main **challenge** of this project: most likely, students of targeted age **never** had any **programming** experience.

For this reason, narrating a story (1.x) offers a very good **starting point** to understand programming flow logic.

In fact, narration is composed of subsequent and alternative events, which will need to be managed in the produced code (2.2).

Managing the design of the narration requires customization and handle of character state, which is the perfect use-case for learning goals 2.3 and 2.4

Pre-requisites

1. Interdisciplinary

- 1.1. Odyssey **characters**
- 1.2. Odyssey **events**

2. Computer Science

- 2.1. Computational thinking
- 2.2. Programming flow (**jump** and **if-else** statements)
- 2.3. Basic notions of **Objects** and **Classes** (just **initialization** and **usage**)
- 2.4. **String** and **boolean** data manipulations
- 2.5. Management of “simple” files (images and audios¹)
- 2.6. RenPy

Delivery to students

Pre-requisites listed above will be delivered in class by the appropriate teachers.

The Project will be presented in class in a **co-held** Literature and Computer Science lecture.

Students will receive **no code** nor **data**, they are expected to populate narration dialogues and scenes on their own.

RenPy ships by default with a simple example story, which can be used as an additional reference.

¹ In proficient version

Delivered from students

Students will have **3 weeks** of time to deliver:

- A list of group **participants**
- A list of Odyssey chapters that the group decided to implement²
- A visual representation of the implemented narration tree (example in appendix 1)
- Project
- Accompanatory slides for the **oral presentation** (approx. 15 mins)

Evaluation

Topic	Points (up to)	Topic level
Implemented studied scenes	+2	Basic
Placed speaker avatars	+1	
Used different colors for different speakers	+1	
Placed background image	+1	
Used RenPy SDK	+1	
Added extra, alternative/artificial paths	+0.5	Advanced
Used Python status variables to handle alternative paths	+0.5	
Presentation quality	+1	
Added dynamic scenes, governed by decisions taken in alternative paths	+0.5	Proficient
Used Python status variables to handle dynamic scenes	+0.5	
Usage of audio clips	+1	

Bonuses:

- +1: **replaced** RenPy **default** opening screen, wording and/or program icon
- +1: added possibility of **resuming** game from a **random scene**, in case of wrong path

² Potentially, different groups can tackle different parts of the poem.

“Potentially” because, in this case, some extra time would be needed, in order to coordinate the different groups and to provide a common “structure” that the project should follow, so as delivered parts can successfully communicate.

