

Snap that grammar!

Teaching youngsters basic English grammar rules in a playful way

Learning Unit 1, Methods in CS: Analysis

Daniele Solombrino, 1743111

Introduction

“Snap that grammar!” is an **interactive chatbot** created in **Snap!**, which can be used by **1st year middle school** teachers as a teaching support tool.

The way a chatbot interacts with students promotes **in-class interaction**, offering an attractive and stimulating way of learning **basic English grammar** rules.

Learning goals

These are the learning goals aimed by Snap that grammar!, grouped by “macro” competences.

1. Interdisciplinary, consolidate knowledge about:

- 1.1. Basic English sentence **structure** (subject + verb + object)
- 1.2. Adding -‘s’ suffix for singular third person
- 1.3. Formulation of an **interrogative** sentence
- 1.4. Formulation of a sentence talking about **possibility** (“would”, “should” or “could”)
- 1.5. Formulation of a **future** sentence (bonus)
- 1.6. **Present** (and **future**, bonus) English verb **tenses**

2. Computer Science

- 2.1. Adoption **computational thinking** to solve real-world problems
- 2.2. The idea of pseudo-code
- 2.3. Consolidate knowledge about **Int**, **String**, **List** and **Tables**¹ manipulations
- 2.4. Consolidate usage of **functions** (created on their **own** or given by **someone**)
- 2.5. Use interactive **chatbots**

3. Soft skills

- 3.1. **Group** work and interaction
- 3.2. Visual **design**
- 3.3. The idea of **adapting** own solutions to already existing work
- 3.4. **Learn** from and **reason** about **mistakes**

¹ In advanced version

Learning goals interaction/motivations/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, basic English grammar rules (1.x) offer a very good **starting point**.

In order to be modeled, grammar requires the creation and **manipulation** of appropriate **data** structures (2.3) and **algorithms** (2.2), taking into consideration what is **already given** to the students (more on this in the “Delivery to students” section).

The last statement encourages students to:

- **Make decisions** according to whatever is already given (2.4 and 3.4), a very common scenario in Computer Science.
- Learn how to deal with data that has been prepared by someone else, meaning that **access logic** may not necessarily coincide with their own (3.4)

Grammar rules are modeled in correct and wrong ways, so the chatbot covers both cases.

Players will answer whether the presented sentence is correct or not.

Learning happens here:

- A correct answer means that students know English grammar, so the chatbot helped them improve what they studied.
- In case of wrong answer, a “redemption” quiz is presented, with the following goals (3.5):
 - Proactively guide students towards the correction of mistakes
 - Teach the importance of observing and reasoning about mistakes, which are the inevitable way towards the eventual success

Pre-requisites

1. Interdisciplinary, studied the following grammar rules:
 - 1.1. Basic English sentence **structure** (subject + verb + object)
 - 1.2. Adding -'s' suffix for singular third person
 - 1.3. Formulation of an **interrogative** sentence
 - 1.4. Formulation of a sentence talking about **possibility** ("would", "should" or "could")
 - 1.5. Formulation of a **future** sentence (bonus)
 - 1.6. **Present** (and **future**, bonus) English verb **tenses**
2. Computer Science
 - 2.1. Non-trivial level of computational thinking
 - 2.2. Interactive chatbots
 - 2.3. Advanced level of Snap!

Delivery to students

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

The project idea will be presented in class in a lecture **co-held** by English and Computer Science teachers.

Students will receive:

- **Pre-loaded data** structures, storing: subjects, verbs and objects (example in appendix 2)
- **Front-end** of chatbot quiz
- **Guidelines** about inputs and outputs that produced code should sport, for a **successful** back \longleftrightarrow front-end **communication**

At the end of the presentations from all groups, the Computer Science teacher will present their version (co-held lecture with the English teacher), hosting an **open discussion** about pros and cons of all delivered solutions.

Delivered from students

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

- A list of group **participants**
- Chatbot back-end (**code** producing correct and wrong sentences, described in previous sections)
- Accompanatory slides for the **oral presentation** (approx. 15 mins)

Evaluation

Whenever encountered, the word “**conceptual**” refers to the **pseudo-code** that the students adopted, which can be **inferred** by looking at the actual delivered code.

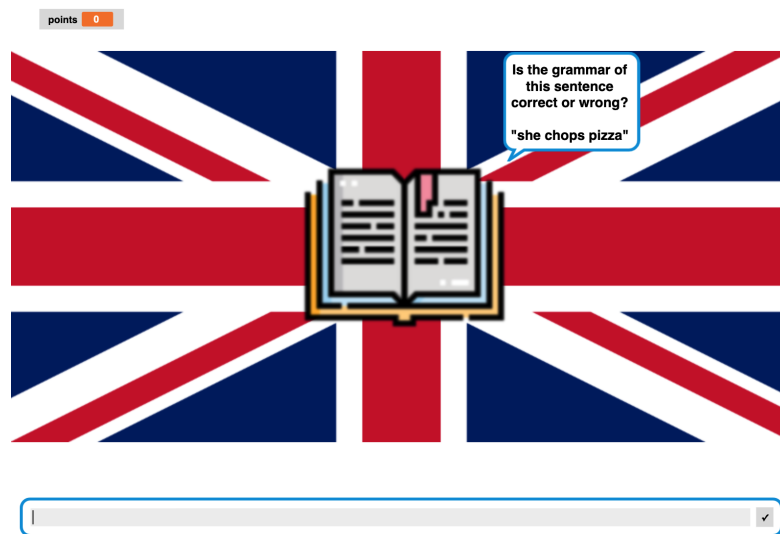
Topic	Points (up to)	Topic level
Conceptual modeling of “generate_correct_random”	+1	Basic
Conceptual modeling of “make_wrong_interrogative”	+1	
Handle “s’ rule” for third singular person	+1	
Correct usage of given data structures and functions	+1	
String \longleftrightarrow list manipulations	+1	
Presentation	+1	
Conceptual modeling of “make_correct_interrogative”	+1	Advanced
Table manipulations	+0.5	
Code quality	+0.5	
Conceptual modeling of “make_correct_wcs”	+0.75	Proficient
Conceptual modeling of “make_wrong_wcs”	+0.25	
Looping over list elements	+1	

Bonuses:

- +2: added an extra, **non-trivial rule**.
For example: “make_future”, which requires handling particular verbs, like those ending with specific vowels, when adding “-ing” suffix

Appendix

1) Chatbot interface



2) CSV files structure

```
→ Snap-that-Grammar git:(main) cat subjects_CSV.csv
subject,person,plurality,auxiliary,be_verb
I,1,S,do,am
you,2,S,do,are
he,3,S,does,is
she,3,S,does,is
it,3,S,does,is
we,1,P,do,are
you,2,P,do,are
they,3,P,do,are
```