PyBreak Escape

Mini project by

Aurélie, Ceci, Filip, Fritzi, Johanna

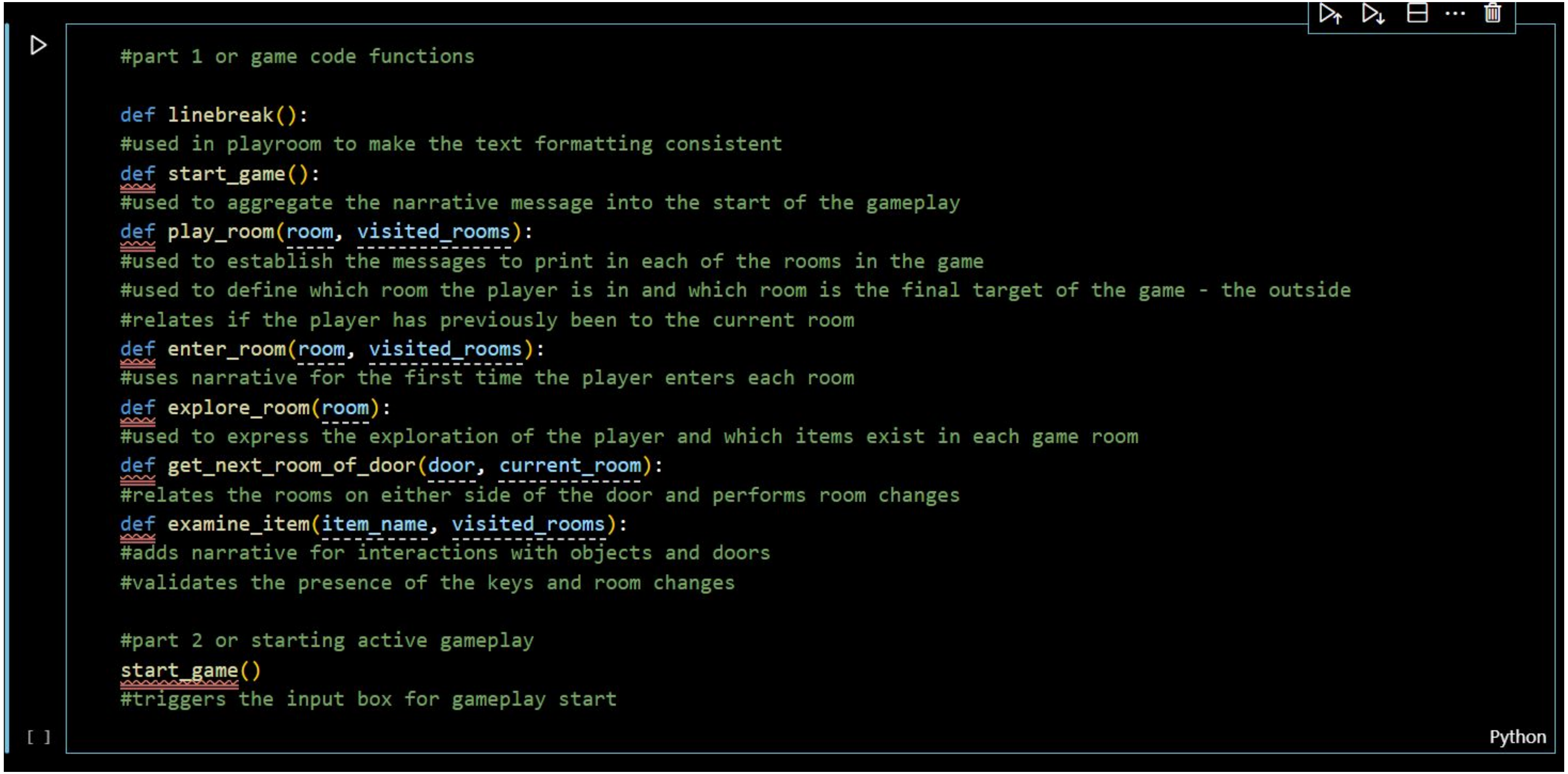
25 October 2024

PyBreak Escape – Game Structure

- General idea:
  - Text-based user interaction
  - RPG (role-playing game) style

- Data structures:
  - Dictionaries:
    - defining features of rooms, objects, keys, messages
    - defining relation between rooms, objects, keys, messages
    - defining start and end of the game
  - Lists:
    - classifying rooms and doors
  - Sets:
    - saving visited rooms

PyBreak Escape – Functions



```
#part 1 or game code functions

def linebreak():
#used in playroom to make the text formatting consistent
def start_game():
#used to aggregate the narrative message into the start of the gameplay
def play_room(room, visited_rooms):
#used to establish the messages to print in each of the rooms in the game
#used to define which room the player is in and which room is the final target of the game - the outside
#relates if the player has previously been to the current room
def enter_room(room, visited_rooms):
#uses narrative for the first time the player enters each room
def explore_room(room):
#used to express the exploration of the player and which items exist in each game room
def get_next_room_of_door(door, current_room):
#relates the rooms on either side of the door and performs room changes
def examine_item(item_name, visited_rooms):
#adds narrative for interactions with objects and doors
#validates the presence of the keys and room changes

#part 2 or starting active gameplay
start_game()
#triggers the input box for gameplay start
```

PyBreak Escape – Features

- Clear and exciting story-telling:
  - Added descriptions and narrative for each room
  - Added some hidden messages
  - Encouraging user to explore the game

- Smooth movement through rooms:
  - Going back and forth between rooms
  - Returning different phrasing when the player re-enters a room

- Intuitive interaction with objects:
  - Added messages to the game objects that did not provide any interaction

Technical Challenge

- Improve user friendliness
  - Make interaction more simple and intuitive
    - e.g., not always enter “examine” or “explore” and instead directly type the object
- Google Colab as a platform:
  - not the most intuitive tool
  - causing problems saving versions”)
- Interpersonal relations
  - Constructing ourselves as a group was challenging in the first day(s)
    - Difficulties to handle the new and stressful situation

Big Mistake

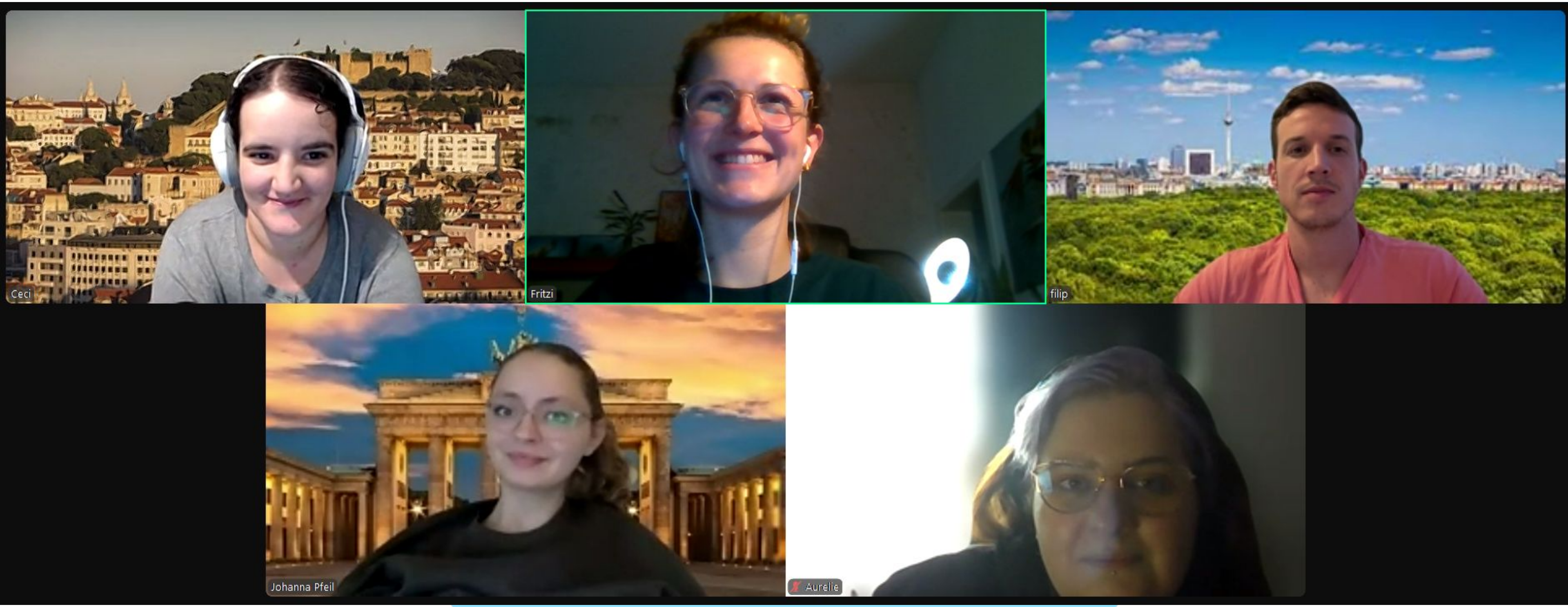
- Losing updated code:
    - We lacked experience with GoogleColab and didn't make a proper plan how to save versions.
        - Learning: Never trust the "automatic saving". Make a backup plan!

- Not having an outline of the project at first:
    - Instead we got ost in small features and needed to come back to change them.
        - Learning: Plan ahead to be more efficient and save time, energy and nerves.

PyBreak Escape – Demo

Try it if you dare! Good luck finding your way out…

https://colab.research.google.com/drive/1LHXl0uv4iZVzRsxh9ZP_UVgwdhOVzRWs#scrollTo=0Ypgsd5BkhM-



Thank you!

IRON HACK FRANCE

IRON HACK GERMANY

IRON HACK PORTUGAL

IRON HACK LISBON

IRON HACK BERLIN

IRON HACK PARIS