



E-book Reader

A BPC/PP2 Project

By BenjoBoys – Jiří Navrátil, Tomáš Kříčka

Description

- This E-book Reader is able to read .txt, .pdf and .epub file formats
- It can be also used as a text editor/notepad saving as a .txt
- Program made in 243 lines of code with comments in 1 python file
- Includes testing files in said formats

Used libraries

- Tkinter
- PyMuPDF
- Os

Initial GUI setup

- Here the header is named and the grid of individual components (textbox, buttons, etc.) is structured

```
6  root = Tk()
7  root.title('Ebook Reader')
8  root.state('zoomed')
9  root.iconbitmap('pixel_book_logo.ico')
10 root.grid_columnconfigure((0,1,2,3,4,5,6), weight=1)
11 root.rowconfigure(0, weight=1)
```

- Textbox is created as shown with expandable/retractable sides that stick to the program window itself

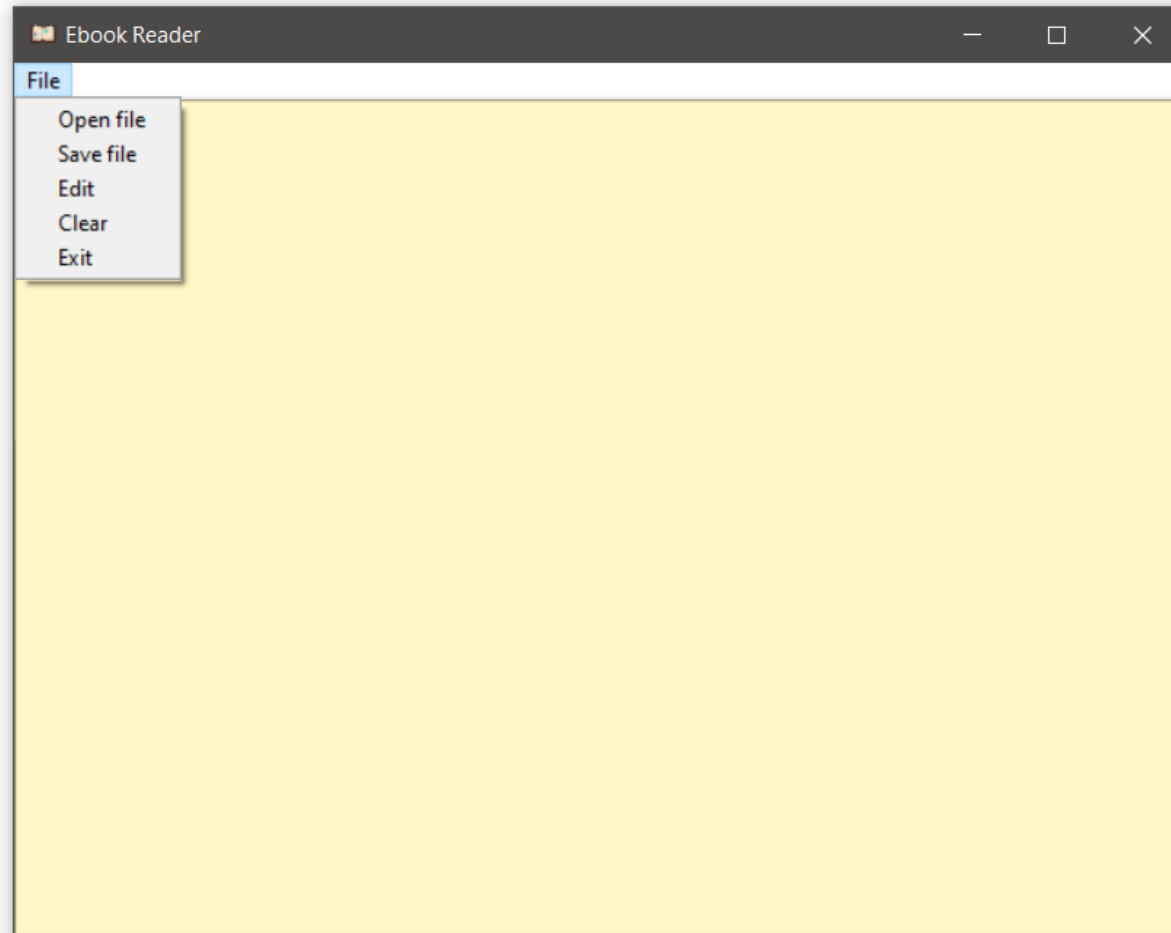
```
24 my_text = Text(root)
25 my_text.pack(pady=10)
26 my_text.grid(row=0, column=0, columnspan=7, sticky=W+E+S+N)
27 my_text.config(bg="#fff5c5")
28 my_text.configure(font=("Calibri", 12))
```

Initial GUI setup

- A file dropdown menu with different functions is made here

```
228 my_menu = Menu(root)
229 root.config(menu=my_menu)
230
231 #add dropdown menus
232 file_menu = Menu(my_menu, tearoff = False)
233 my_menu.add_cascade(label="File", menu=file_menu)
234 file_menu.add_command(label="Open file", command=openFile)
235
236 file_menu.add_command(label="Save file", command = saveThisFile)
237
238 file_menu.add_command(label="Edit", command=editTextbox)
239 file_menu.add_command(label="Clear", command=clearTextBut)
240 file_menu.add_command(label="Exit", command=root.quit)
```

Initial GUI setup



Main function

- The whole program is based around the `open_file()` function where the user selects desired file, he wants to be presented, with a `filedialog`

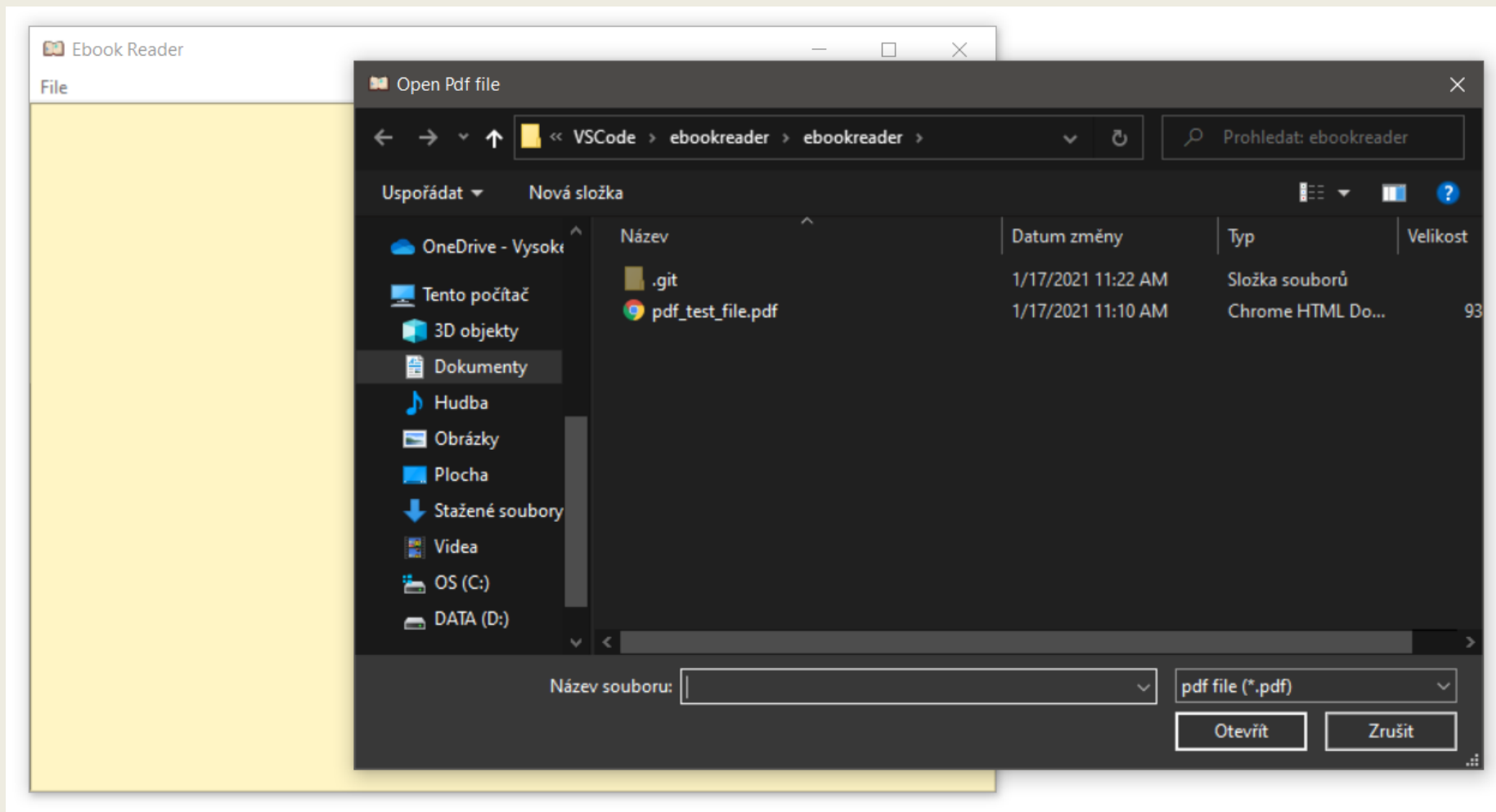
```
43     open_file = filedialog.askopenfilename(  
44         initialdir = os.getcwd(),  
45         title = "Open Pdf file",  
46         filetypes = (("pdf file", "*.pdf"),  
47                     ("epub file", "*.epub"),  
48                     ("txt file", "*.txt"),  
49                     ("All files", "*.*"))  
50     )
```

- After selection, the buttons and page counter are created

```
91     button_back = Button(root, text="<<", height=2, width=10, state=DISABLED)  
92     button_back.grid(row=2, column=1)
```

```
95     page_label = Label(root, textvariable=label_text)  
96     page_label.grid(row=2, column=3)
```

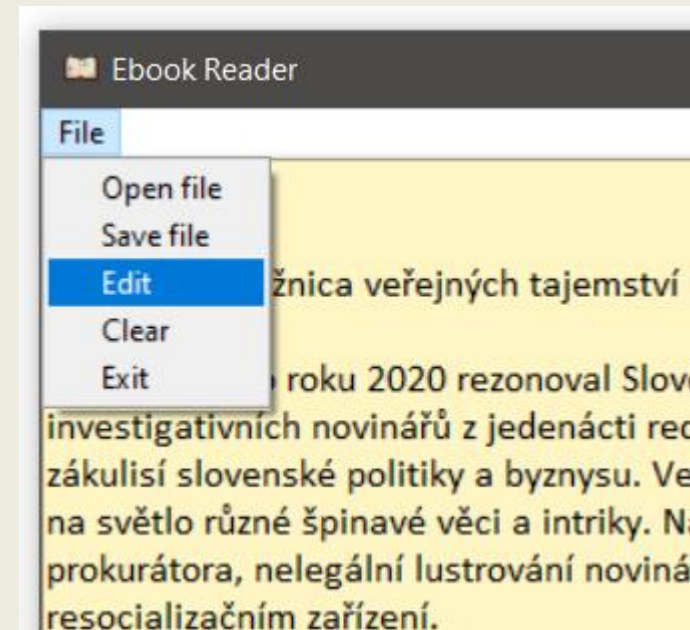
Main function



Editing

- As was mentioned earlier, you can edit currently viewed page of an opened file
- The editing feature must be enabled as it is disabled initially after opening any file

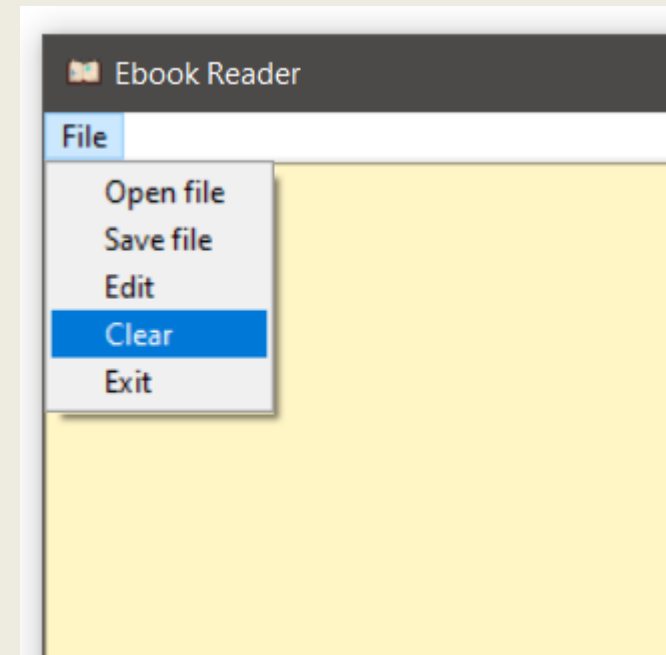
```
117 def editTextbox():  
118     my_text.configure(state=NORMAL)
```



Clearing

- If the user wants to use the program as a notepad but has a file opened, a simple clear function can be called to get rid of any text and unnecessary buttons and labels

```
100 def clearTextBut():
101     global doc
102
103     #init doc
104     doc = None
105
106     #delete text in textbox
107     deleteTextboxContent()
108
109     #delete buttons and label with page numbers
110     button_forward.destroy()
111     button_back.destroy()
112     page_label.destroy()
```

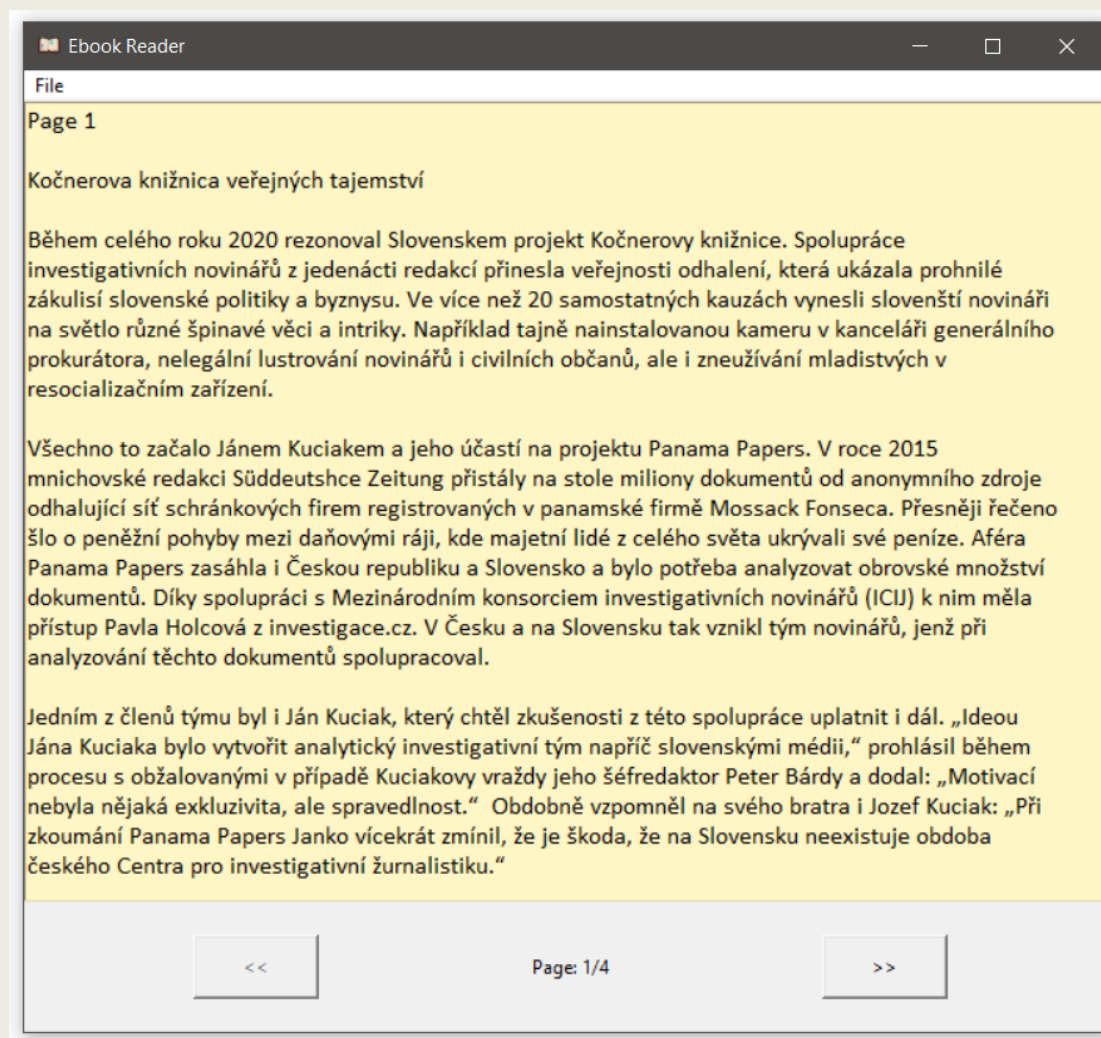


Saving

- After the user edits his parts of a text, it can be saved as a .txt file with the changes made on the specific page

```
131     if created_file:
132         saved_file = open(created_file, 'w')
133
134         #for every page insert coresponding text
135         for i in range(0,number_of_pages):
136
137             #if a change was made in a speciffic page, include it in a save
138             if i == page_number_label-1:
139                 saved_file.write(my_text.get(1.0, END))
140             else:
141                 page_content = doc.loadPage(i)
142                 page_text = page_content.getText("text")
143                 saved_file.write(page_text)
144
145         saved_file.close()
```

Example



Problems

- A scrollbar is not included
- Words cut in the middle when the window is small enough to do so
- .pdf and .epub files with only 1 page don't set the page counter correctly
- Only 3 file formats are supported (.pdf, .epub, .txt)
- Saving in only .txt format

Collaboration between programmers

- For communication we used Discord
- Work was done through Live Share module in Visual Code
- Files themselves were transferred with GitHub's Desktop app