

Školski Pomoćnik

SADRŽAJ

1. Naziv
2. Sadržaj
3. Problem
4. Ideja
5. Rješenje
6. Tijek izrade programa
7. Testiranje
8. Plan za budućnost
9. Korištenje programa
10. Resursi

PROBLEM

Spremajući se za ispit iz matematike, tražio sam način za ponoviti gradivo.

Sadržaj udžbenika nije bio dovoljan.

Skoro sve stranice i programi koje sam pronašao su imali već unesene zadatke i bile su na engleskome jeziku.

IDEJA

Za Python sam čuo na satu informatike. Radili smo unos i ispis.

Na prvu me nije interesirao jer još nisam znao koliko je s njim moguće. Preko ljeta, nakon sedmoga razreda, sam na YouTube-u vidio tutoriale za igrice u Python-u. Pogledao sam nekoliko tih tutoriala i napravio igrice iz njih, kao i neke vlastite. Već sam išao na natjecanje Algoritmi.

Dok sam na Infokupu gledao zadatke prošlih godina, vidio sam kategoriju natjecanja Razvoj Softvera. Htio sam napraviti svoj prvi program na kojemu ću raditi duži period vremena.

Napravio sam program koji nasumično zadaje zadatke.

Nedostatak već postojećih programa je količina zadataka.

Zadaci su već uneseni stoga ima ograničen broj zadataka.

Trebalo bi stvarati nasumične programe, tako da je jedino ograničenje broj kombinacija brojeva.

Rješenje

cilj je bio napraviti aplikaciju za sve razrede osnovne škole, ali sam stigao napraviti samo prvi i drugi.

Budući da se tako rano još ne uče decimalni brojevi, ima manje mogućih zadataka.

Tijek izrade programa

Ovo je moj prvi veliki projekt. Još nisam bio upoznat sa objektno orijentiranim programiranjem, stoga sam puno naučio dok sam izrađivao program.

Naišao sam na mnoge probleme za vrijeme izrade programa, ali sam ih većinom brzo riješio. Ovo su problemi na koje sam potrošio najviše vremena.

1. Prva prepreka na koju sam naišao je način mijenjanja prozora.

Moje prvo rješenje je bilo brisati i postavljati gumbe i na taj način mijenjati prozore, ali je to vrlo sporo.

Drugo rješenje je imati već napravljene prozore u klasama, pa samo pozvati onaj koji želimo. Ovaj način je puno brži stoga ga koristim.

```
19
20 #mijenjanje frame-ova u prozoru
21 container = tk.Frame(self)
22 container.grid(column = 0, row = 0)
23 container.grid_rowconfigure(0, weight=1)
24 container.grid_columnconfigure(0, weight=1)
25
26 self.frames = {}
27
28 #i ovom tuple se nalaze svi frame-ovi između kojih mijenjamo
29 for F in (StartMenu, GradeMenu, Grade1SubjectMenu, Grade1Math,
30         Grade1MathAddition, Grade1MathSubtraction, Grade1MathTasksAdd,
31         Grade1MathTasksSub,
32         Grade2SubjectMenu, Grade2Math,
33         Grade2MathAddition, Grade2MathSubtraction, Grade2MathMultiplication, Grade2MathDivision,
34         Grade2MathTasksAdd, Grade2MathTasksSub, Grade2MathTasksMul, Grade2MathTasksDiv,
35         ComputerScience, Algorithms, AlgorithmTasks, AlgorithmExplanation,
36         Renderer):
37
38     frame = F(container, self)
39
40     self.frames[F] = frame
41
42     frame.grid(row=0, column=0, sticky="nsew")
43
44 self.showFrame(StartMenu)
45
46 def showFrame(self, page_name):
47     frame = self.frames[page_name]
48     #stavi na vrh stacka
49     frame.tkraise()
```

Morao sam početi ispočetka jer sam već koristio prvo rješenje kada sam otkrio drugo.

2. Na sljedeći problem sam naišao kada sam mijenjao način na koji postavljam gumbe, labele i unose u prozor.

Samo se dio prozora učitao, dok je ostatak bio prazan.

Naišao sam na čudno rješenje. Naime kada sam mijenjao veličinu labele, primijetio sam da se veći dio ekrana prikazao nego prije.

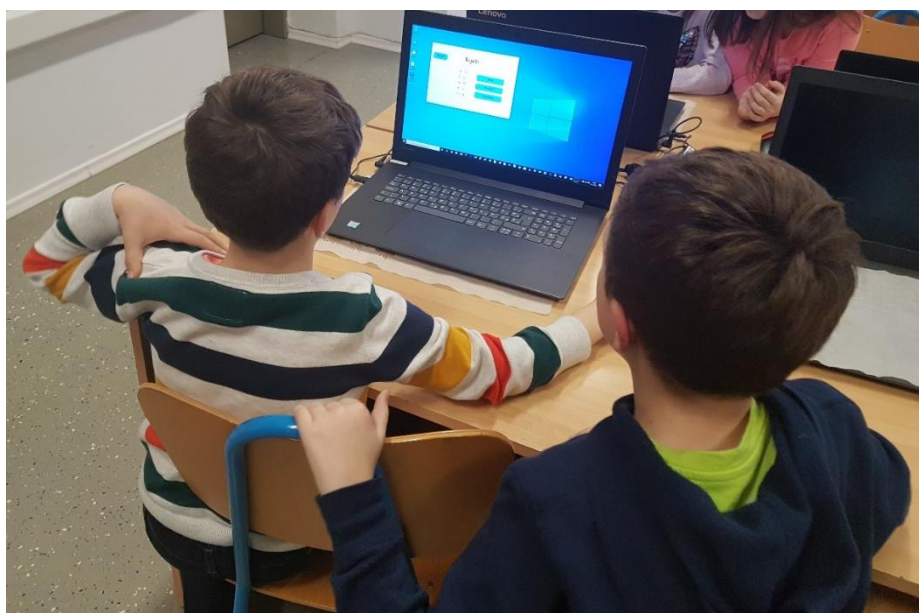
To sam iskoristio tako da sam napravio klasu `Renderer`. U njoj postavim veliku labelu u prozor, tako da se veći dio prozora prikaže.

```
51
52
53 class Renderer(tk.Frame):
54     def __init__(self, parent, controller):
55         #init frame
56         tk.Frame.__init__(self, parent)
57         #init
58         label1_ = tk.Label(self, text = "\n", font = ("Comic Sans MS", 1000, "bold"))
59         #place
60         label1_.grid(column=0, row=0)
61
62
63
```

Testiranje

Provedeno je testiranje u 2.c razredu Osnovne škole Milana Langa. Učenici su se s lakoćom snalazili u programu uz minimalnu pomoć.

Povratna informacija je vrlo pozitivna uz ideje za poboljšanje. Neke od njih su: dodati animacije u algoritamske zadatke, povećati font i dodati slike u program.



Plan za budućnost

Program se može lako nadograditi.

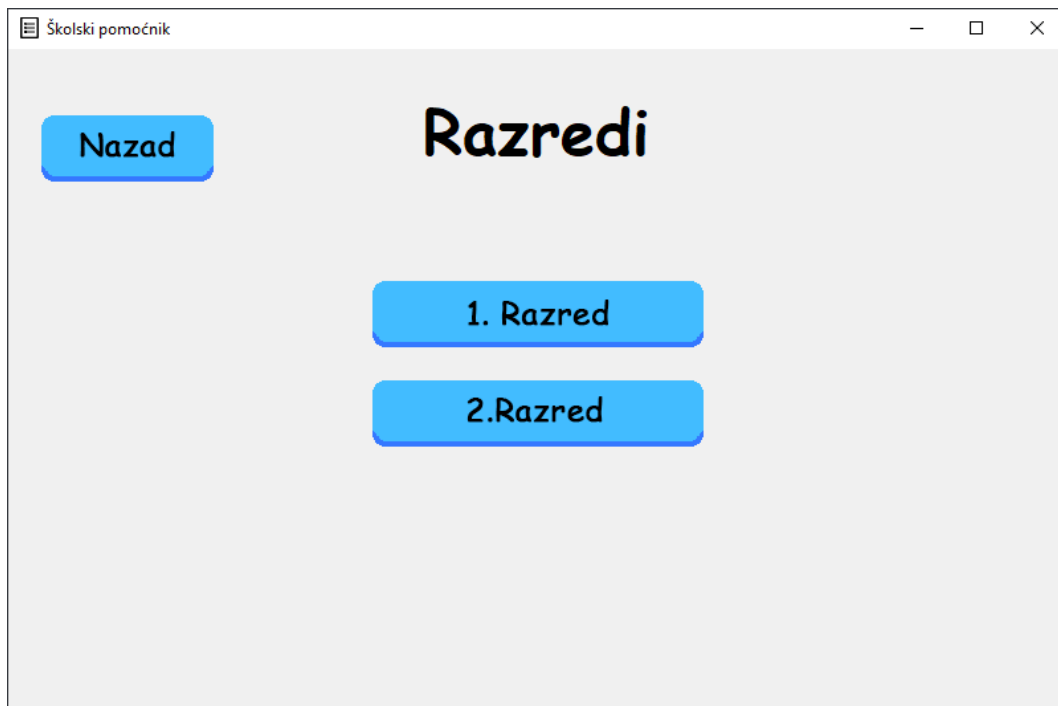
Mogu dodati još razreda, zadataka i predmeta.

Cilj je da program postane osnovno sredstvo za ponavljanje gradiva. Plan je nadograditi ga tako da se njime mogu koristiti svi razredi osnovne škole, za većinu predmeta predmete.

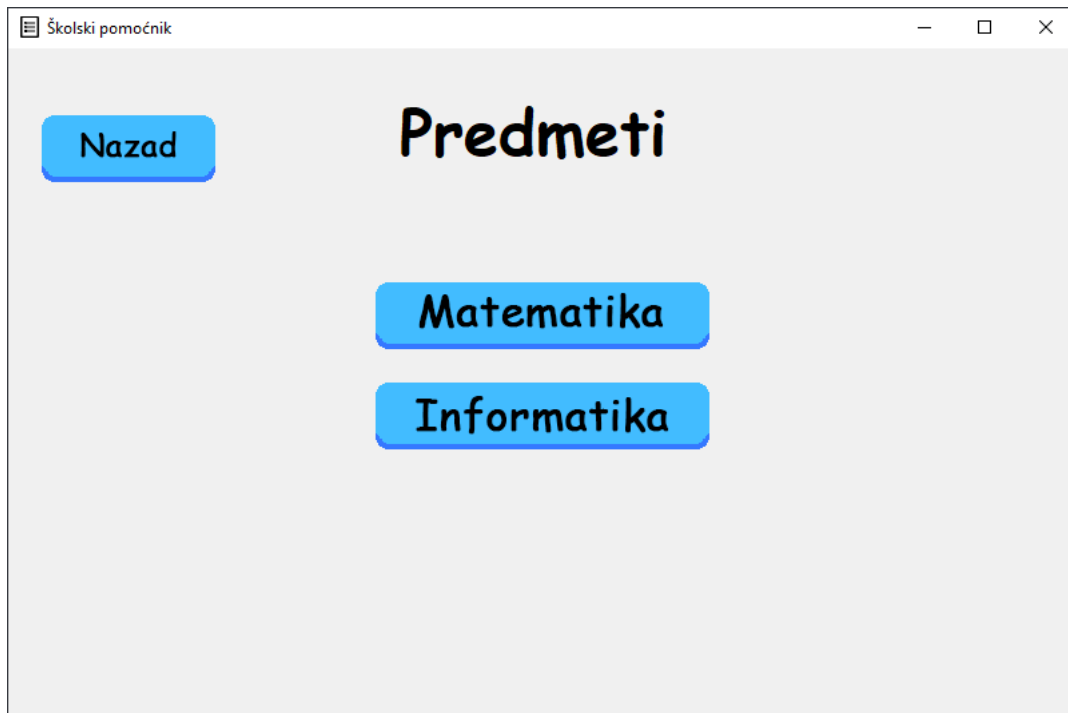
Pitao bih profesore i učenike svoje škole za ideju, kako poboljšati program.

Korištenje programa

Nakon što ste stisnuli kreni na glavnome meniju, trebate odabrati razred.



A zatim ćete trebati odabrati predmet.



Ako ste izabrali matematiku, izaberite koje gradivo želite.



Resursi

Programski jezik: Python 3.7.2 [Preuzmi](#)

Uređivač teksta: Visual Studio Code [Preuzmi](#)

Moduli:

1. Tkinter - za grafičko sučelje
 2. random - za generiranje nasumičnih brojeva
 3. webbrowser - za otvaranje poveznica
- Svi moduli dolaze sa Pythonom 3.7.2