autor: Marin Hrvaćanin, 8.C
mentor: prof. Sandra Ivković
Osnovna škola Vjenceslava Novaka, Zagreb

WonderBuilder

Projektna dokumentacija

Sadržaj:

1. Uvod

Priča o softveru

Odluka o izradi softveru pojavila se krajem protekle godine (2018.) iako se ona kao ideja skrivala već duže vrijeme u mojoj glavi. Bila je zapravo prisutna sve vrijeme mog „programerskog“ života i želje da roditeljima ne samo pojasnim što to radim već i da ih potaknem na učenje osnova programiranja u aplikacijama poput Pythona ili JavaScripta. Naravno, njima je to bilo previše kompliciranо i daleko, pa sam smišljao način kako da im to pojednostavim.

Jednog sam se dana sjetio da bi mogao napraviti softver kojim bih im mogao objasniti **princip rada takvih aplikacija**. Odlučio sam se za Python i krenuo programirati. Malo sam istraživao i pronašao PyGame i tkinter module.

Prvotna je ideja bila da korisnik složi program od programskih pločica koje svaka sadrži nekoliko linija koda, no ubrzo sam shvatio da kada bi se taj kod zapisivao u Python datoteku, **ona bi se mogla pokretati i samostalno**, bez mog softvera, samo pomoću odgovarajućih modula. Odjednom se ispred mene otvorila mogućnost široke, gotovo neograničene primjene mog softvera u stvaranju vlastitih programa, igrica, prezentacija, a posebno se to odnosi na velike mogućnosti korištena u nastavi kroz učenike i nastavnike/profesore kojima kompjutori nisu strani bez obzira što ne znaju programirati no spremni su raditi svoje aplikacije spajajući programske blokove (pločice) i koristeći tekstovne, zvučne i video datoteke koje imaju osobno ili su im lako dostupne na internetu, spajajući ih u svoje prezentacije i aplikacije.

S obzirom da se od petog razreda natječem na infokupovim natjecanjima u kategorijama algoritmi i Logo, bio sam upoznat s mogućnošću prijave i na ovu kategoriju te sam to s veseljem i učinio.

2. O softveru

2.1. Što je Wonder Builder?

Wonder Builder softver namijenjen je djeci i starijim početnicima za edukaciju o programiranju aplikacija, no njegova kasnija nadogradnja daje mu (i dat će mu) širu korisničku, upotrebnu vrijednost.

Korisnik slaže programske blokove koji se kasnije tijekom kompiliranja postupno zamjenjuju naredbama iz programskog jezika Python. Na odabranoj lokaciji od strane korisnika kreira se Python (.pyw) datoteka s programom i pripadajući multimedijski sadržaj.

Moguće je stvarati simulacije koje se mogu primjenjivati i u nastavi, za potrebe najrazličitijih prezentacija, početne i jednostavnije igrice, itd.

2.2. Po čemu je Wonder Builder poseban?

Wonder Builder je poseban, osim po svojoj jednostavnosti, većoj funkcionalnosti od drugih softvera na tržištu koji idu istim tragom te širokoj primjenjivosti i po tome što se **jednom kompilirani program može koristiti samostalno**, na računalima koja uopće nemaju instaliran Wonder Builder softver, a napravljene aplikacije mogu se neograničeno razmjenjivati i koristiti.

Naravno, za mene je poseban i po tome što se radi o vrlo složenom i u cijelosti autorskom radu koji sadržava za sada više od 2.000 linija koda.

2.3. Pozicija na tržištu

Široka, praktično neograničena primjenjivost uz moguću bogatu nadgradnju te činjenicu da je ovaj segment tržišta „plitak“ i tek u razvoju, izdvaja Wonder Builder kao dugoročan projekt s realnom tržišnom perspektivom.

2.4. Budućnost softvera

Wonder Builder se neprestano razvija što će se posebno odnositi na njegovu budućnost kao jednog od softvera korištenih i u edukacijske svrhe. Neki od planova za razvoj u bliskoj budućnosti su:

Kratkoročni:

- uklanjanje bug-ova (grešaka u kodu)
- dodavanje mnoštva novih funkcija

- olakšavanje rada i sintakse

Srednjoročni:

- dodavanje mogućnosti vizualnog uređivanja aplikacija
- stvaranje galerija gotovih audio i video materijala i isječaka
- Wonder Builder web stranica s mogućnošću prijave i dijeljenja programa s drugim korisnicima i forumom za upite i odgovore

Dugoročni:

- online škole i edukacije početnika za korištenje ovog softwera te poticanje za kasnije samostalno programiranje

3. Potrebne sistemske značajke

- Windows 8 , Windows 8.1 , Windows 10
- Python 3.7.1+
- Moduli: PyGame , tkinter , tendo , ctypes , PIL , PhotoImage (iz tkinter paketa), Image (iz PIL paketa), time , shutil , random , winsound, math, ttk
- Fontovi: Roboto (uključujući Roboto Light), Century Gothic, Consolas, helvetica, League Spartan
- Rezolucija: 1600x900 px

4. Operatori i operacije

4.1. Računske operacije:

Operacija	Primjer	Rezultat
Zbrajanje (+)	8 + 2	10
Oduzimanje (-)	8 - 2	6
Množenje (*)	8 * 2	16
Dijeljenje (/)	8 / 2	4.0

Cjelobrojno dijeljenje (//)	8 // 2	4
Ostatak pri dijeljenju (%)	8%2	0
Eksponenti (**)	8**2	64

4.2. Logičke operacije:

Operacija	Operator	Primjer	Rezultat
I	and , &	True and False	False
ILI	or , |	True or False	True
NE	not	not True , not False	False , True

4.3. Uspoređivanje:

Operacija	Operator	Primjer	Rezultat
Jednako	== , is	3 == 5 , 6 == 6	False , True
Različito	<> , != , is not	3 != 3 , 6 <> 8	False , True
Je veće	>	3 > 1	True
Je manje	<	3 < 1	False
Je veće ili jednako	>=	4 >= 3 , 4 >= 4	True , True
Je manje ili jednako	<=	4 <= 5 , 6 <= 6	True , True

4.4. Operatori vezani uz varijable:

Operacija	Primjer
Povećaj za	a += 7 je isto što a = a + 7
Smanji za	a -= 7 je isto što a = a - 7
Pomnoži s	a *= 2 je isto što a = a*2
Podijeli s	a //= 2 je isto što a = a//2
Jednako	a = 6 (varijabla 'a' poprima vrijednost 6)

4.5. Napomene:

= -> postavlja vrijednost (a = 6)

== -> uspoređuje vrijednost (if a == 6)

Wonder Builder je napravljen u Python programskom jeziku i preuzeta je pripadajuća sintaksa. Gore navedeni primjeri i operacije/operatori nisu jedini koji se mogu koristiti, također koristi se PyGame-om, pa je tako djelomično preuzeta tablica događaja koja se sastoji od jednakih događaja (5.2).

5. Naredbe:

5.1. Popis naredbi

ON START – ono što je spojeno na pločicu 'on start' bit će izvršeno na početku programa i neće se ponavljati.

MAIN LOOP – programske pločice spojene na 'main loop' pločicu će se izvršavati sve dok se program ne zaustavi.

SET – programska pločica 'set' postavlja vrijednost varijable, liste, itd.

IF – pomoću 'if' pločice moguće je pod nekim uvjetom izvršiti dio koda (za gramatiku uspoređivanja pogledati 4.3. i 4.5.).

FOR – 'for' je pomoćna petlja koja se sastoji od varijable i broja ponavljanja.

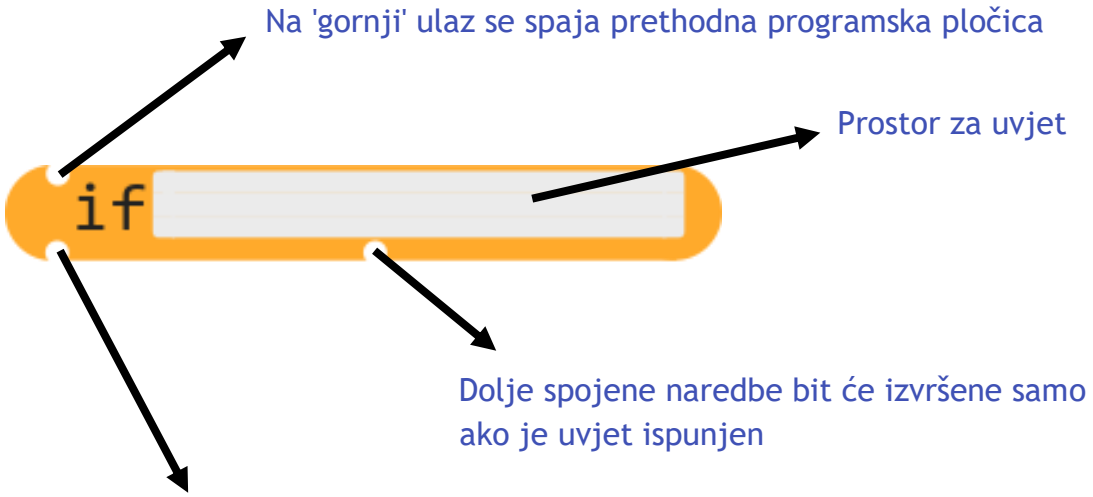
IMAGE – služi za učitavanje slika podržanih tipova (.jpg i .png) u prozor (5.2. – Primjer 2.).

SOUND – koristi se winsound modulom da bi se puštao odabrani zvučni zapis u pozadini (.wav oblika).

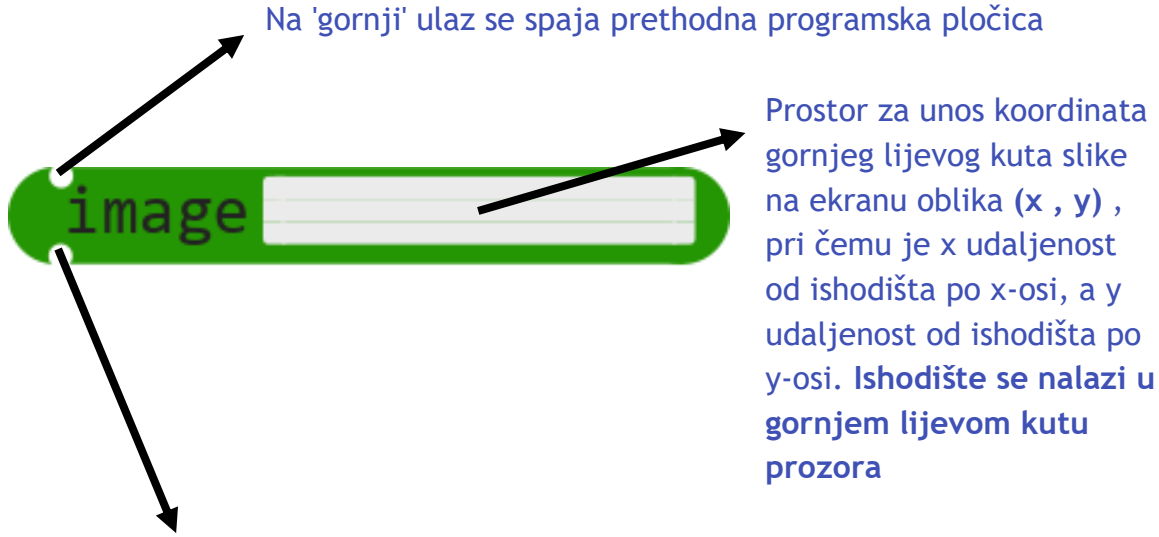
GIF – prikazivanje gif animacije u prozoru, koordinate se zapisuju jednako kao i kod 'image' naredbene pločice (5.2. – Primjer 2.).

5.2. Način spajanja naredbi

Primjer 1. – „IF“



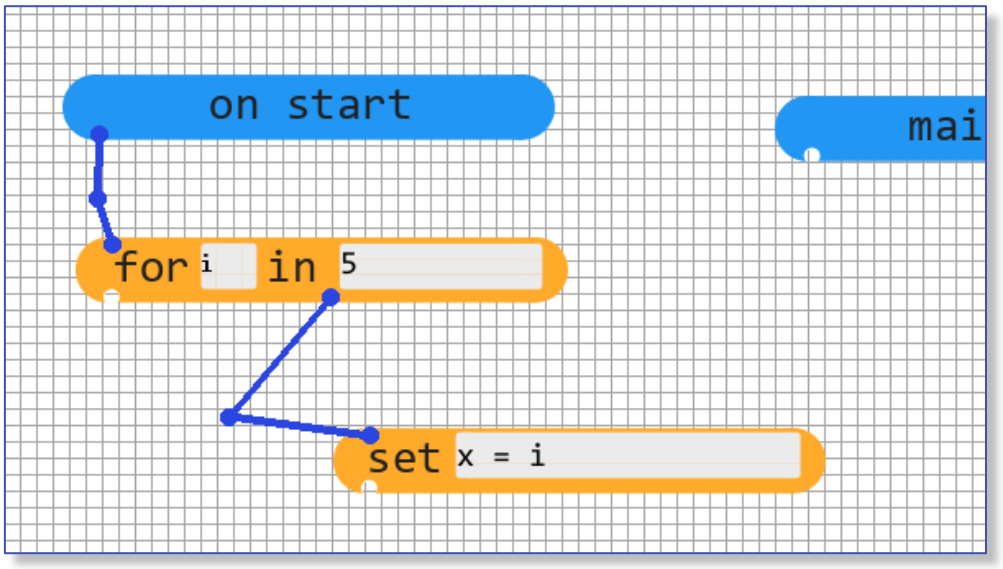
Primjer 2. – „IMAGE“



5.3 FOR petlja

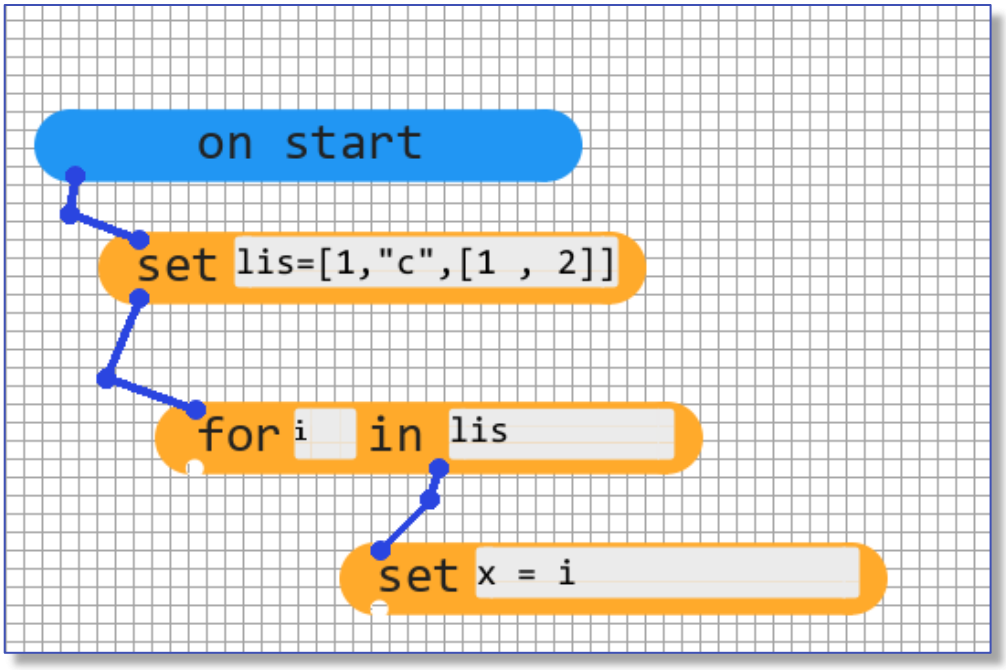
For petlja služi za ponavljanje neke radnje zadani broj puta.

Proučimo sljedeći primjer:



Pet puta će se pri pokretanju programa izvršiti naredba 'set x = i', a pritom će varijabla 'i' imati početnu vrijednost 0, zatim 1, pa 2 i tako do 4 (u svakom 'krugu' ponavljanja varijabla će se povećati za jedan cijeli broj).

Također, moguće je i koristiti for petlju za prolazak po listi:



U gornjem primjeru na prvi pogled se može primijetiti da druga varijabla u for petlji nije broj već ranije definirana lista 'lis'. Ova će for petlja proći kroz listu 'lis', a pritom varijabla 'i' neće biti broj nego će biti trenutni element u listi. U ovom primjeru 'i' će prvo imati vrijednost 1, zatim “c“ i na kraju [1 , 2], a nakon toga petlja će završiti.

6. Događaji (event's)

Prilikom izvođenja svakog ponavljanja glavne petlje („main loop“), prikuplja se niz događaja pomoću kojih možemo ostvariti interakciju korisnika sa tijekom izvođenja programa. Vrste događaja mogu biti: pritisak tipke tipkovnice (KEYDOWN), otpuštanje tipke tipkovnice (KEYUP), pomak računalnog miša (MOUSEMOTION), pritisak tipke na računalnom mišu (MOUSEBUTTONDOWN) i otpuštanje tipke na računalnom mišu (MOUSEBUTTONUP).

6.1. Pritisak tipke i otpuštanje tipke (Keyboard down , Keyboard up)

Dolje navedene tipke se bilježe u nizu događaja:

NAZIV TIPKE	SIMBOL
BACKSPACE	\b
TAB	\t
CLEAR	
RETURN	\r
PAUSE	
ESCAPE	^[
SPACE	

NAZIV TIPKE	SIMBOL
SEMICOLON	;
LESS	<
EQUALS	=
GREATER	>
QUESTION	?
AT	@
LEFTBRACKET	[

NAZIV TIPKE	SIMBOL
EXCLAIM	!
QUOTEDBL	“
HASH	#
DOLLAR	$
AMPERSAND	&
QUOTE	
LEFTPAREN	(
RIGHTPAREN	)
ASTERISK	*
PLUS	+
COMMA	,
MINUS	-
PERIOD	.
SLASH	/
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
COLON	:
U	u / U
V	v / V
W	w / W
X	x / X
Y	y / Y
Z	z / Z
DELETE	
KP0	
KP1	
KP2	
KP3	
KP4	

NAZIV TIPKE	SIMBOL
BACKSLASH	\
RIGHTBRACKET	]
CARET	^
UNDERSCORE	_
BACKQUOTE	`
A	a / A
B	b / B
C	c / C
D	d / D
E	e / E
F	f / F
G	g / G
H	h / H
I	i / I
J	j / J
K	k / K
L	l / L
M	m / M
N	n / N
O	o / O
P	p / P
Q	q / Q
R	r / R
S	s / S
T	t / T
F5	
F6	
F7	
F8	
F9	
F10	
F11	
F12	
F13	
F14	
F15	
NUMLOCK	

NAZIV TIPKE	SIMBOL
KP5	
KP6	
KP7	
KP8	
KP9	
KP_PERIOD	.
KP_DIVIDE	/
KP_MULTIPLY	*
KP_MINUS	-
KP_PLUS	+
KP_ENTER	\r
KP_EQUALS	=
UP	
DOWN	
RIGHT	
LEFT	
INSERT	
HOME	
END	
PAGEUP	
PAGEDOWN	
F1	
F2	
F3	
F4	

NAZIV TIPKE	SIMBOL
CAPSLOCK	
SCROLLOCK	
RSHIFT	
LSHIFT	
RCTRL	
LCTRL	
RALT	
LALT	
RMETA	
LMETA	
LSUPER	
RSUPER	
MODE	
HELP	
PRINT	
SYSREQ	
BREAK	
MENU	
POWER	
EURO	€

7. Tutorial

1. Pokrenimo Softver

2. Nakon pokretanja vidljiv je naslov „Wonder Builder“ i animacija učitavanja prilikom koje se učitavaju svi potrebni sadržaji da bi softver radio ispravno. Ovo može potrajati i do 10 sekundi, ovisno o zakrčenosti radne memorije i kvaliteti uređaja (slika 7.1.).

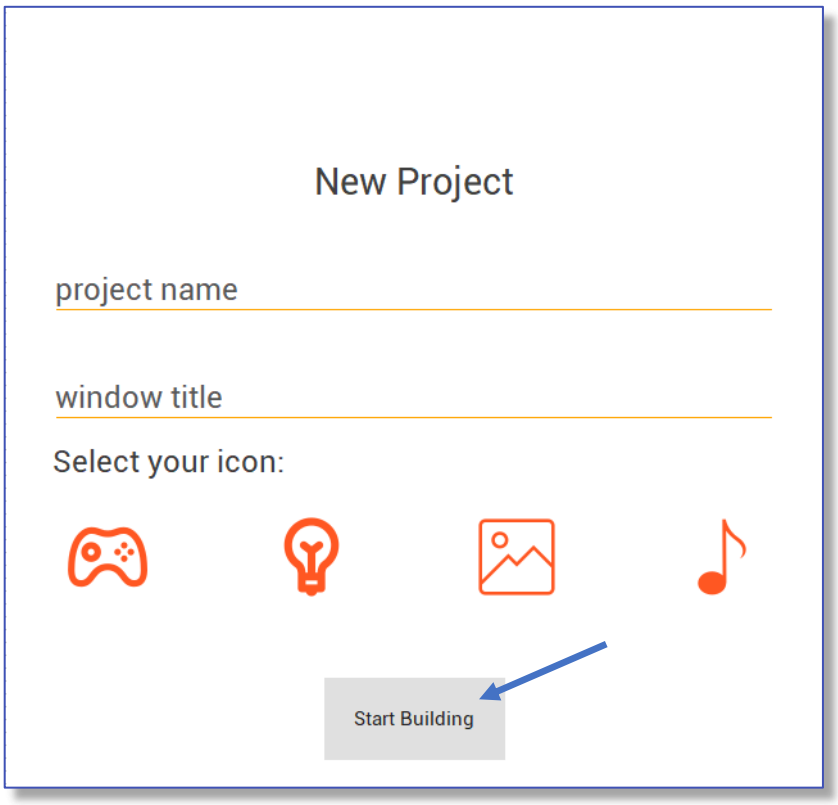
slika 7.1.

3. Pred nama se otvorio obrazac u kojem kreiramo novi projekt („New Project“) (slika 7.2.). Na crtu za unos „project name“ treba upisati ime novog projekta. Za potrebe ovog prikaza nazvat ćemo ga „MojProjekt“.

   U 'C:\Users\...\Documents\Wonder Projects\' će se pojaviti mapa s tim nazivom. **To je polje obvezno ispuniti.**

   Polje za unos „window title“ služi za postavljanje naslova naše aplikacije. Kao primjer unesimo ne baš pretjerano kreativan naziv „MojNaslov“.

   Također, moguće je i izabrati jednu od ponuđenih ikona prozora koje mogu opisivati temu naše aplikacije. Kada smo sve postavili po željenim postavkama, pritisnimo dugme „Start Building“ (slika 7.2.).

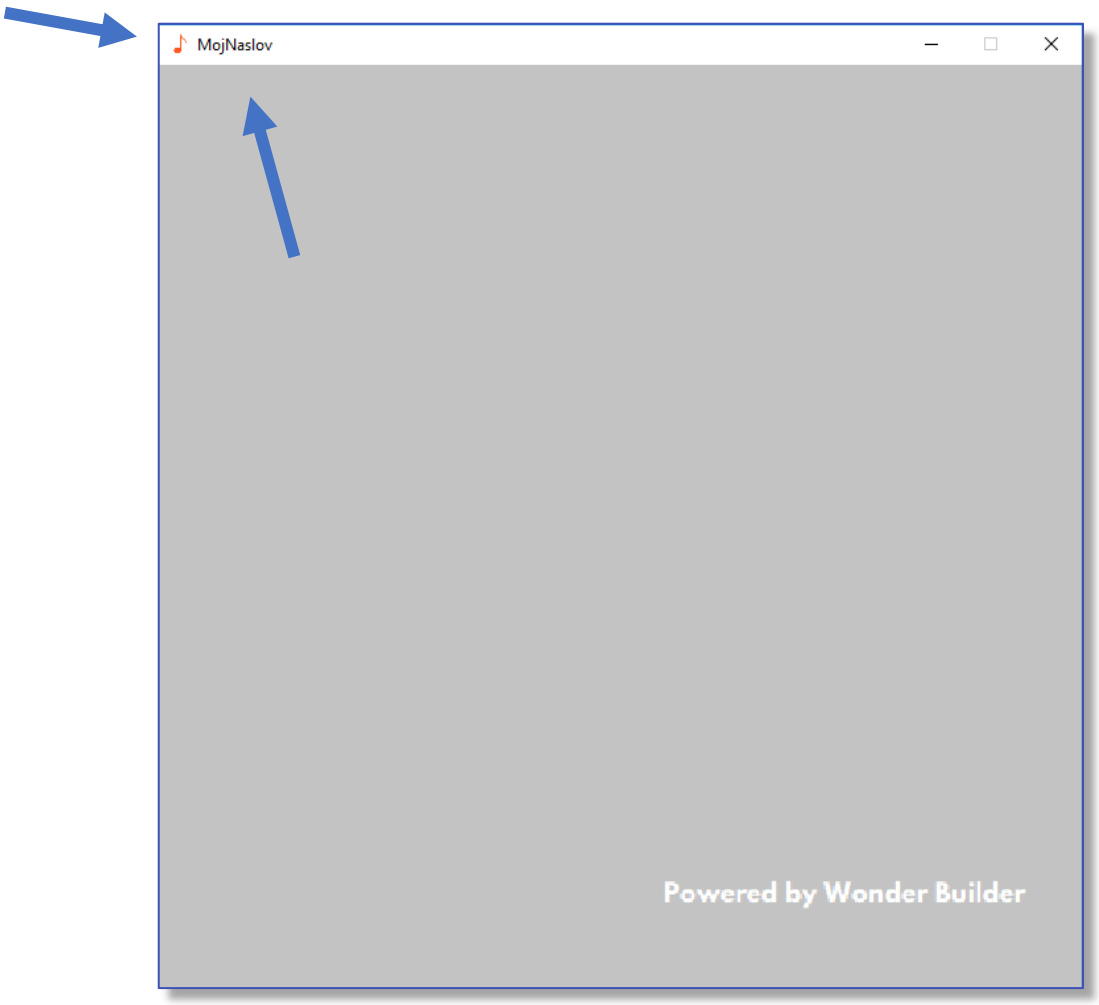


slika 7.2.

4. Na mrežastoj podlozi se već od početka nalaze programske pločice *on start* i *main loop* (slika 7.3.).

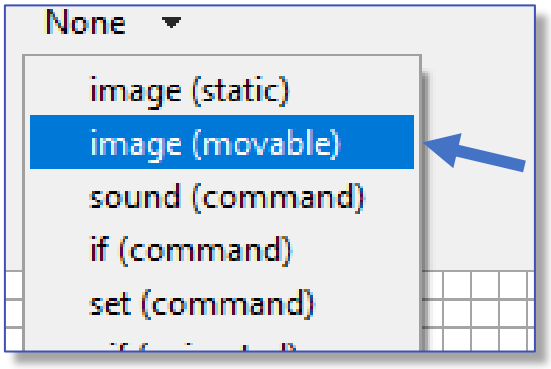
slika 7.3.

5. Pritisnimo dugme *compile* (sastaviti), a potom dugme *run* (pokrenuti). Pojavio se prozor s našim naslovom i odabranom ikonom (slika 7.4.). Dugme *compile* pretvorit će program složen programskim pločicama u datoteku programskog jezika Python (.pyw) koju dugme *run* pokrene. Pozadina je siva sa natpisom „Powered by Wonder Builder“.

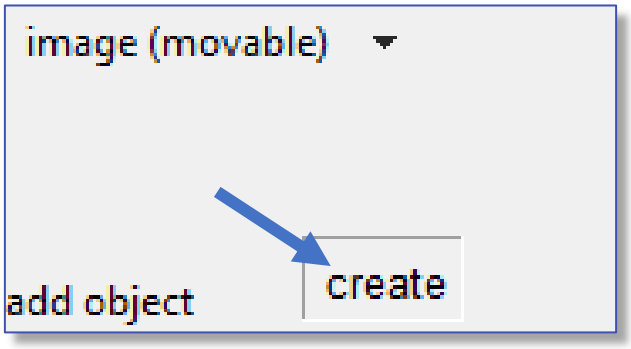


slika 7.4.

6. Promijenimo pozadinu za početak. Odaberimo *image (movable)* pod sekcijom *add object* (slika 7.5.) i pritisnimo dugme *create* (slika 7.6.).

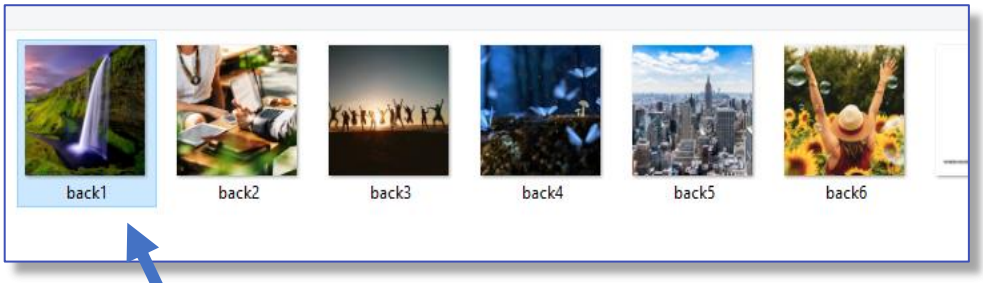


slika 7.5.

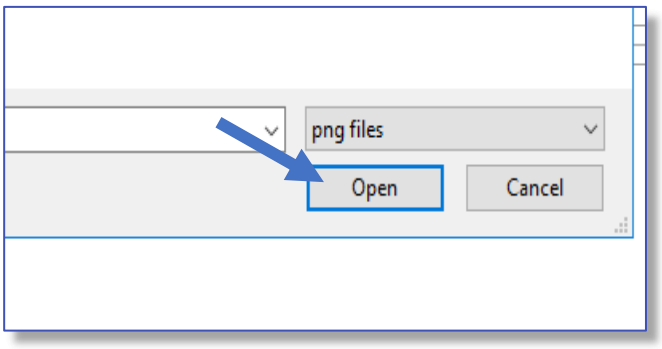


slika 7.6.

7. Otvorio se izbornik za odabir slike (slika 7.7.) (.jpg i .png formati slike su podržani). Pritiskom na dugme *Open* vraćamo se u glavni prozor (slika 7.8.).



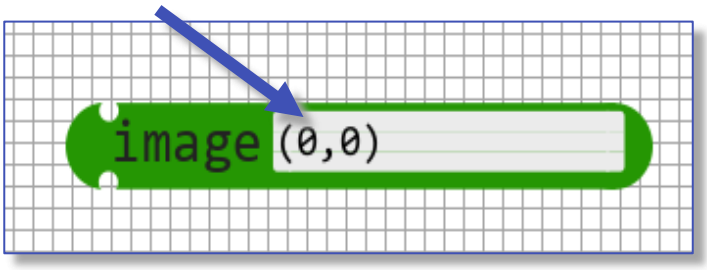
slika 7.7.



slika 7.8.

8. Dočekala nas je pločica *image* (slika), koja sadrži dio u kojem se nalaze dva broja odvojena zarezom u zagradama (slika 7.9.), a na početku su (0 , 0). To su koordinate gornjeg lijevog kuta slike. Gornji lijevi kut prozora je ishodište i također ima koordinate (0 , 0).

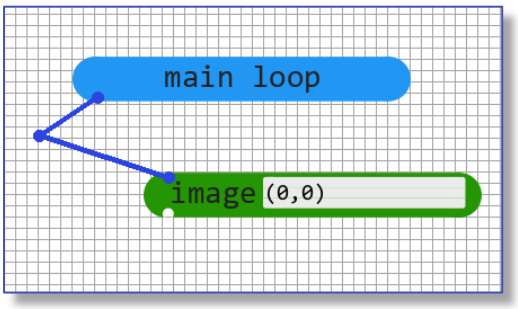
   Ako poželimo promijeniti koordinate slike, potrebno je mišem pritisnuti sivu površinu na pločici na kojoj se nalaze koordinate (slika 7.9.). U tom se trenutku otvara polje za unos u koje je potrebno upisati koordinate oblika (x-koordinata , y-koordinata). Ipak, ostavimo koordinate pozadine na (0 , 0) kako bi ona prekrila cijeli ekran*.



slika 7.9.

*slika će prekriti cijeli ekran pod uvjetom da je veća ili jednaka veličini prozora (700x700).

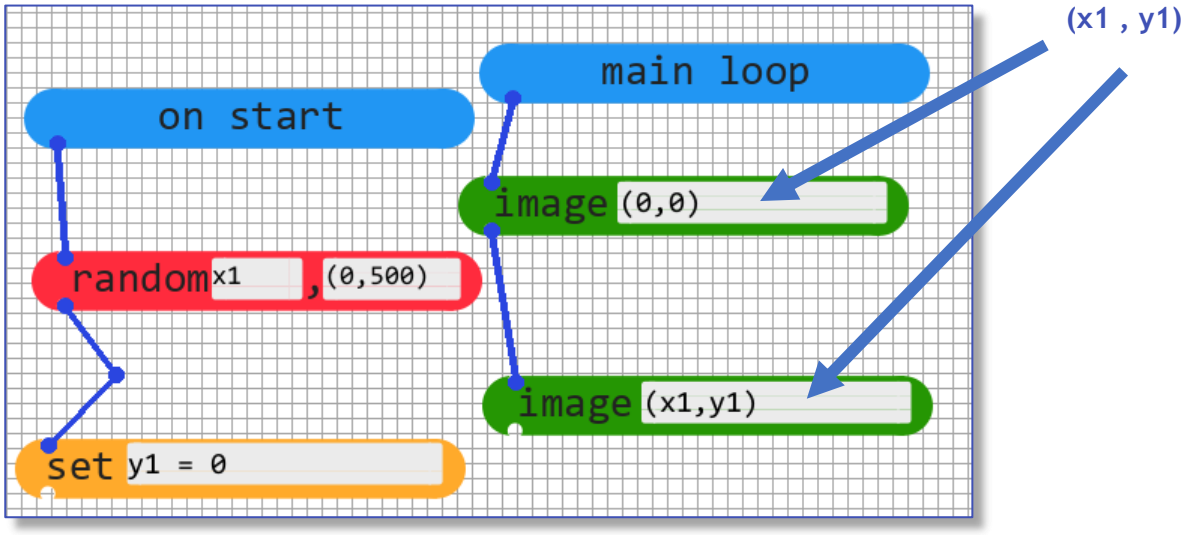
9. Kada pritisnemo dugmad „compile“+“run“, primijetit ćemo da se pozadina nije promijenila, to je zato što pločicu nismo uvrstili u program. Spojimo je direktno na *main loop* (slika 7.10.), da bi se na početku svakog 'kruga' u main loop petlji prvo na ekranu prikazala pozadina (slika 7.11.).



slika 7.10.

slika 7.11.

10. U našoj igri glavni će se igrač izmicati padajućim blokovima (slika 7.12.). Postavimo blok (sliku 200x100 piksela) na y koordinatu 0 i nasumičnu x koordinatu u rasponu od 0 do širina prozora-širina slike, u ovom slučaju od 0 do 500 (slika 7.13.).



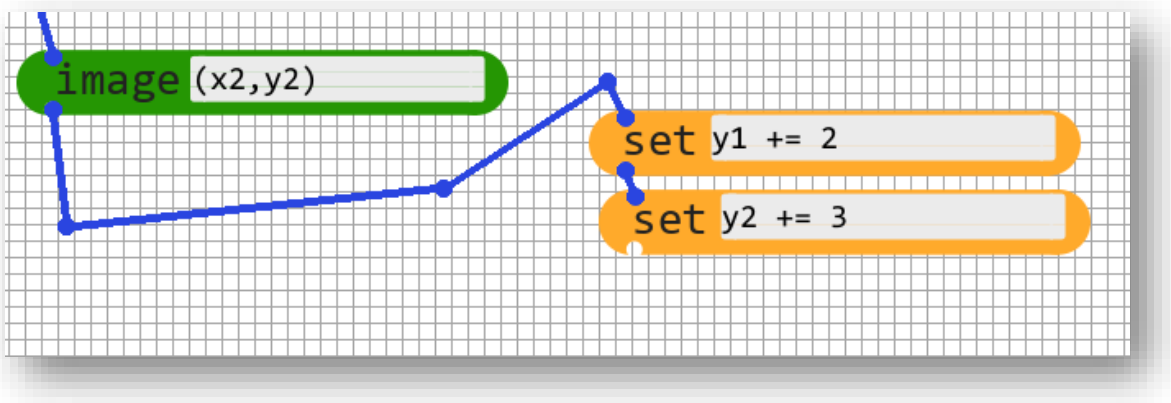
slika 7.13.

slika 7.12. 

Ukoliko nakon pritiska na dugme *compile* pokrenete program nekoliko puta, primijetit ćete da se blok svaki put nalazi na drugoj udaljenosti od lijevog ruba prozora (slika 7.14.). Dodajmo još jedan blok istim postupkom samo ovaj put nazovimo koordinate x2 i y2.

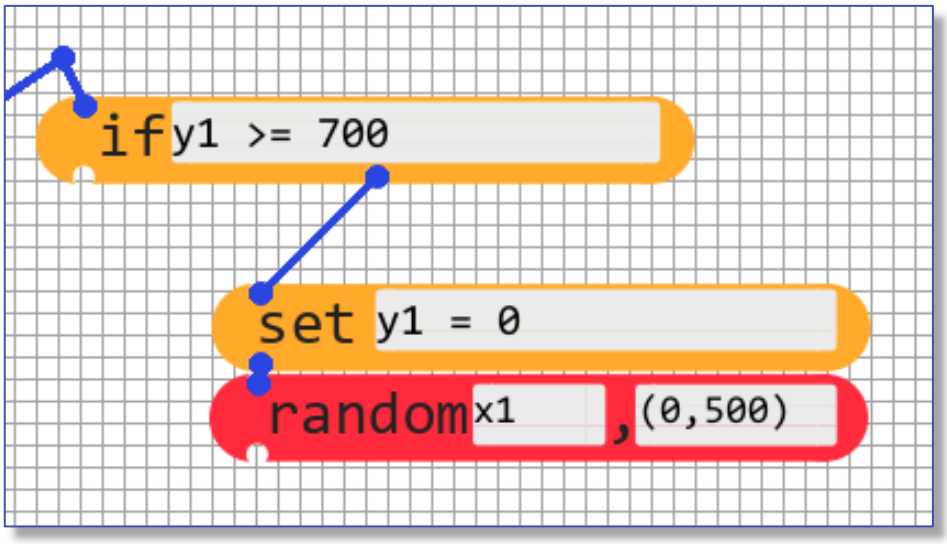
slika 7.14.

11. Da bi blokovi počeli padati, potrebno je stvoriti 'gravitaciju'. Taj efekt postižemo povećavanjem udaljenosti od gornjeg ruba ekrana u svakom ponavljanju (slika 7.15.). Nakon ponovnog kompiliranja programa, vidljivo je da se blokovi kreću prema 'dolje' nejednakom brzinom.



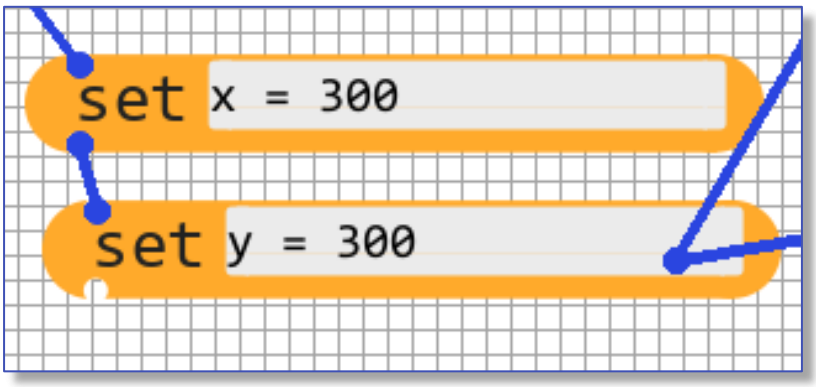
slika 7.15.

Sada se trebamo pobrinuti da jednom kada blok padne ispod donje razine prozora, počinje padati ispočetka. To ćemo riješiti *if* uvjetom (slika 7.16.) za obje dvije slike.



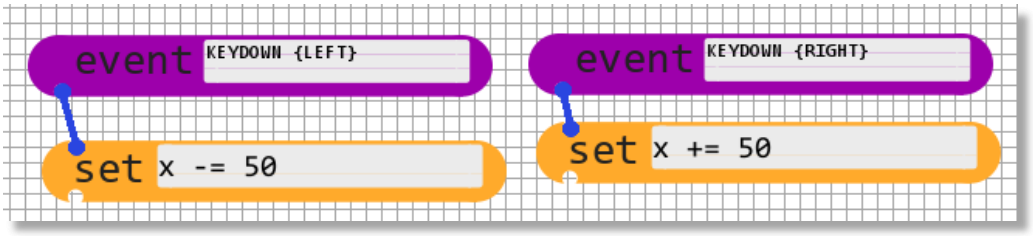
slika 7.16.

12. Vrijeme je da dodamo igrača. Stvorimo prvo, na početku programa, varijable x i y koje će označavati igračevu trenutnu poziciju u prozoru (slika 7.17.). Učitajmo sliku igrača dimenzija 50x50 piksela. Slika će se pojaviti na koordinatama (x , y).



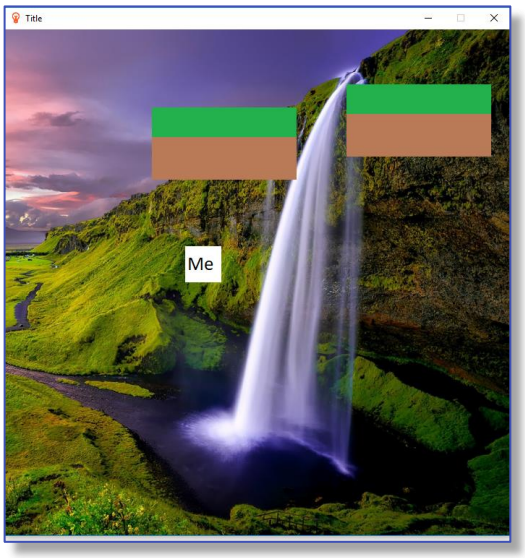
slika 7.17.

13. Da bismo postigli kretanje igrača pritiskom strelica '←' i '→' potrebno je dodati provjeru događaja (slika 7.18.). Ako korisnik pritisne lijevu (left) strelicu smanjit ćemo x koordinatu slike igrača za 50 piksela, a ako pritisne desnu (right) strelicu, povećat ćemo x koordinatu igrača za 50 piksela.



slika 7.18.

14. Na kraju ovog krajnje pojednostavljenog prikaza/primjera korištenja, trebali bi dobiti nešto ovako kako slijedi na slici 7.19.



slika 7.19.