

Smeh življenja

Tomaž Štrus

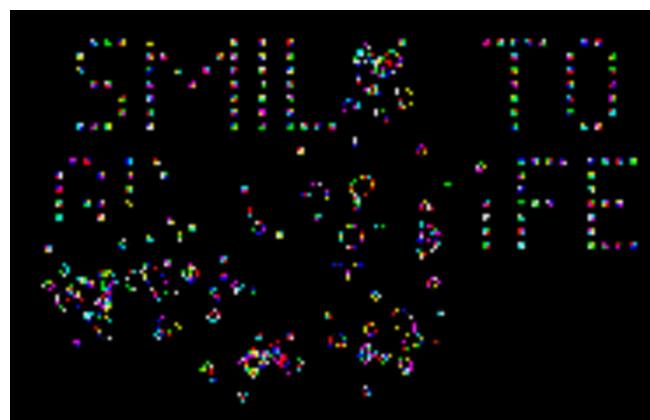
Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana

Povzetek. Smeh Življenja je prototip inštalacije, ki s pomočjo zaznave obraza omogoča opazovalcu interakcijo z igro življenja na zaslonu.

Ključne besede: Igra življenja · smehek · zaznava obraza.

1 Uvod

Tekom poročila bom predstavil projekt, ki sem ga ustvaril pri predmetu Interaktivnost in oblikovanje informacij. Cilj je bil narediti prototip inštalacije primerno za neko razstavo. Moja končna rešitev dela po naslednjem protokolu. Na razstavi imamo zaslon, kjer čez celoten zaslon prikazujemo našo aplikacijo. Ob zaslonu je tudi kamera, katera zajema sliko na zelo kratkem intervalu. Na začetku opazovalca na ekranu pričaka bel napis "Smile to give life" na črnem ozadju. Če na zajeti sliki zaznamo, da se opazovalec nasmehne, dodamo na ekran na relativni poziciji slike v primerjavi s svetu igre življenja nove žive celice igre. Ob prvem nasmehu se spremeni tudi barvni prizor, tako napis in celice niso več črno beli ampak postanejo obarvani, vsaka celica s svojo naključno barvo. Tako ima lahko opazovalec interakcijo z inštalacijo, kolikor časa hoče. Če pride naslednji opazovalec, je na voljo gumb za ponastavitev inštalacije, lahko pa kar nadaljuje v svetu, ki ga je predhodnik pustil za sabo.



Slika 1: Slika ekrana sredi interakcije.

2 Navdih in podobni projekti

Navdih za projekt je prišel iz projekta 15 sekund slave [3]. Tu mi je bil všeč pristop, da opazovalec preko kamere oblikuje umetnost. Sprva sem hotel, da bi ob detekciji obraza naredili sliko opazovalca, potem bi to sliko prevedli v barvno igro življenja, s katero bi lahko predstavili notranji boj posameznika, saj bi si lahko različne barve predstavljal kot dobre ali pa zle sile v posamezniku. To se je v praksi izkazalo za težko izvedljivo, prototipi, ki sem jih naredil niso zgledi v redu in celotna zadeva ni imela lepega efekta za razstavo, zato sem preusmeril idejo v trenutno verzijo.

Ker sem že imel detekcijo obraza in hkrati detekcijo točk ustnic, sem si rekел, da bi lahko preko tega izračunali, kdaj se opazovalec smeje. Preko tega je prišla končna ideja za projekt, da bi interakcijo z igro življenja imeli preko smeha.

Veliko je umetniških inštalacij, katere te poskusijo spraviti v smeh oziroma ti izboljšati voljo, kar je tudi postal cilj moje inštalacije. Podobna inštalacija je Smile ustvarjalcev studia Varvara Mar [2], kjer se ob nasmehu v kamero prižgejo luči, katere so pritrjene na steno in so v obliki smeškota. Sliko obiskovalke in prižganega smeškota vidimo na sliki 2.

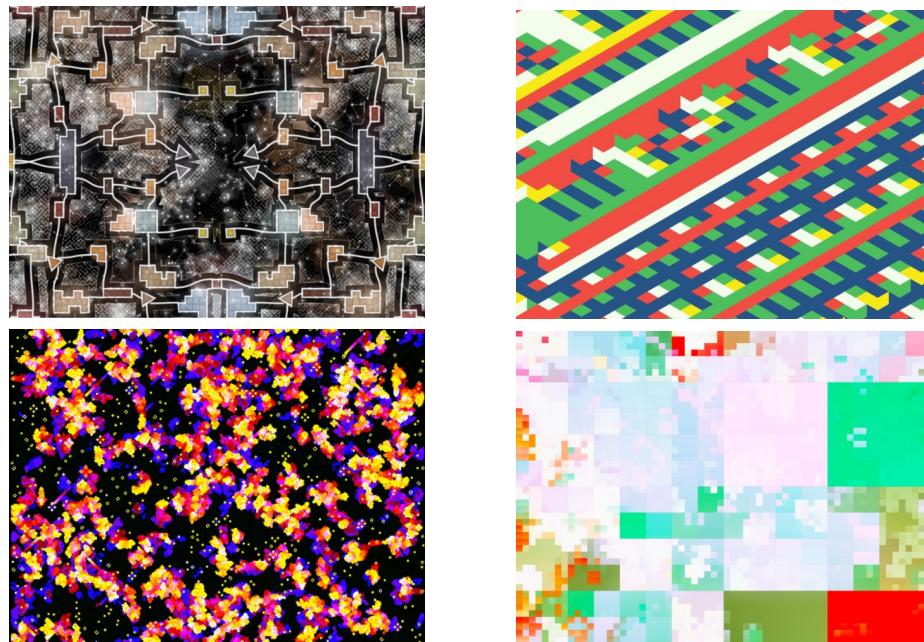


Slika 2: Inštalacija Smile: Varvara & Mar

Inštalacij iz strani Igre življenja je tudi veliko, par primerov je npr. Christian Lölkesova inštalacija Game of Life [4], katero vidimo na sliki 3. Popularne so tudi različne variacije igre, katere imajo drugačna pravila za igro življenja in uporabljajo posebna pravila za uporaba barv. Par primerov vidimo na sliki 4, tu vidimo, da lahko s pomočjo različnih pravil dobimo zelo različno barvno generativno umetnost, kar je tudi cilj moje inštalacije.

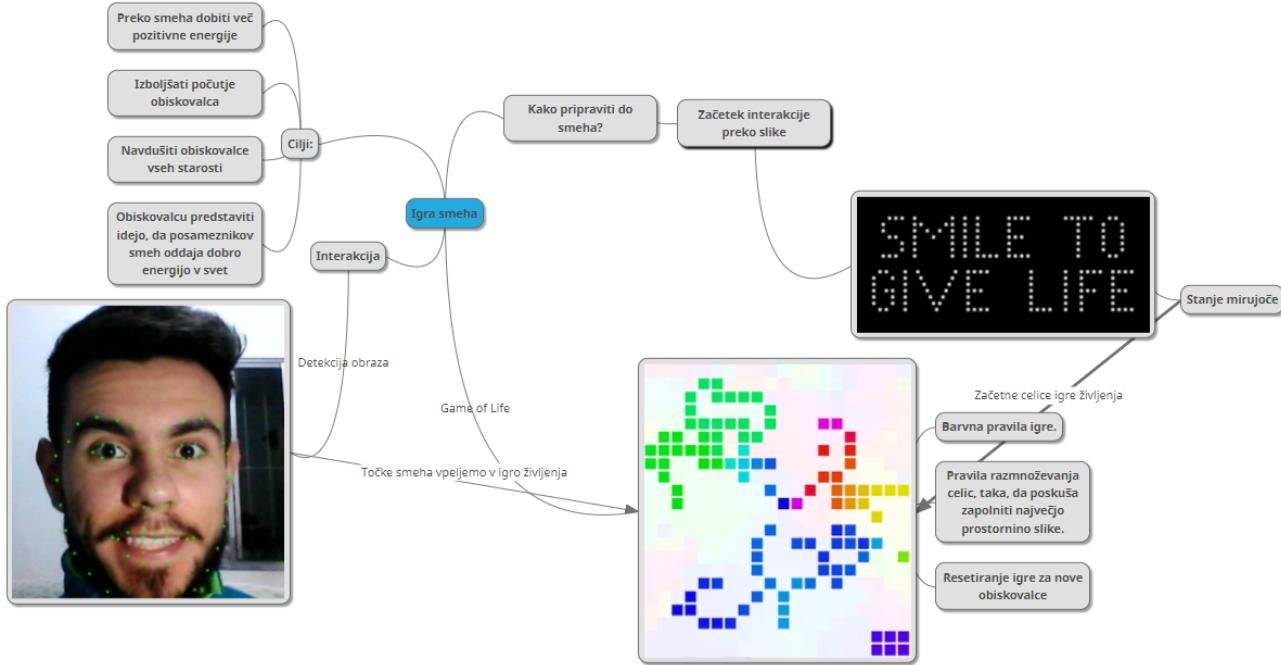


Slika 3: Inštalacija Game of Life: Christian Lölkes



Slika 4: Različne varijacije igre življenja [5]

3 Mindmap



Slika 5: Mindmap projekta

4 Uporabljene knjižnice in delovanje

Celoten projekt je spisan v programskem jeziku Python 3. V njem pa uporabljamo naslednje knjižnice in programe:

- Knjižnica OpenCV2 [6]
- Knjižnica DLIB [7]
- Knjižnica threading [8]
- Knjižnica NumPy [9]
- Čista python implementacija Game of Life

4.1 Knjižnica OpenCV2

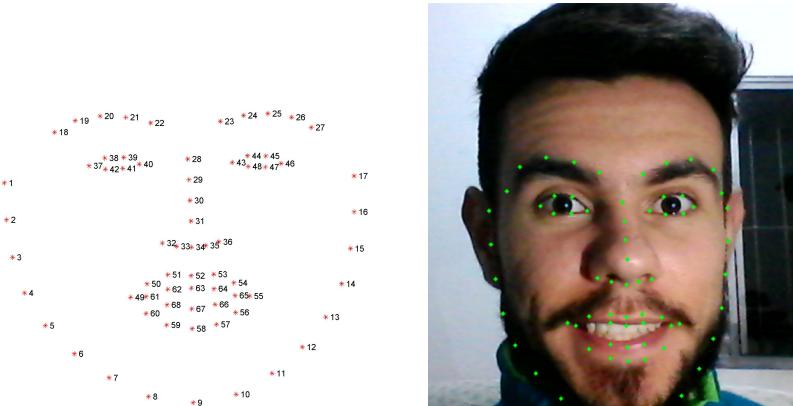
S pomočjo knjižnice upravljamo naslednje stvari:

- Zajem slike preko sistemске kamere.
- Procesiranje Game Of Life celic, da se lahko prikažejo kot slika.

- Upravljanje in prikazovanje slike v aplikacijskem oknu.
- Upravljanje z dogodkim iz tipkovnice.
 - Tipka ESC za izhod iz aplikacije.
 - Tipka R, ki ponastavi stanje inštalacije. Ta tipka simulira fizičen gumb, ki bi bil nastavljen pred inštalacijo.

4.2 Knjižnica DLIB

Knjižnica DLIB je namenjena zaznavavanju obraza. DLIB omogoča uporabo njihovih vnaprej treniranih modelov za klasifikacijo. Za klasifikacijo obraza sem uporabil njihov Haar kaskadni klasifikator. Ta deluje tako, da za vhod dobi sliko in na nej pregleda ali zazna obraz. Ob zaznamem obrazu nam klasifikator vrne 68 vnaprej določenih točk obraza, katere nam povedo kje na naši sliki je nek del obraza. Indeksi točk so vedno isti, tako npr. vemo da so točke od 49 do 68 tiste, ki ležijo na ustnicah. Primer take detekcije obraza in razporeditev točk lahko vidimo na sliki 6. Preko točk ustnic nato izračunam, ali se osebek smeji ali ne.



Slika 6: Primer indeksov DLIB točk na levi in praktične aplikacije na sliki obraza na desni strani

4.3 Knjižnica threading

Knjižnica threading je uporabljena, da paralelno z glavnim procesom zajamemo sliko in na njen naredimo detekcijo obraza, ter preračunamo ali smo zaznali smehek. S tem metodo se izognemo zelo kratkih intervalov, kjer se igra ne odziva. Tako lahko igra konstantno teče, točke obraza pa nam paralelni proces vrne sproti in opozori glavni proces, da ima na voljo rezultate. Tako brez ustavljanja obdelujemo dobljene točke. Za dodatno hitro obdelavo paralelnega procesa ne prekinemo do konca programa. Ta nam, kolikor hitro zmore vrne iskane točke (približno 20-krat na sekundo).

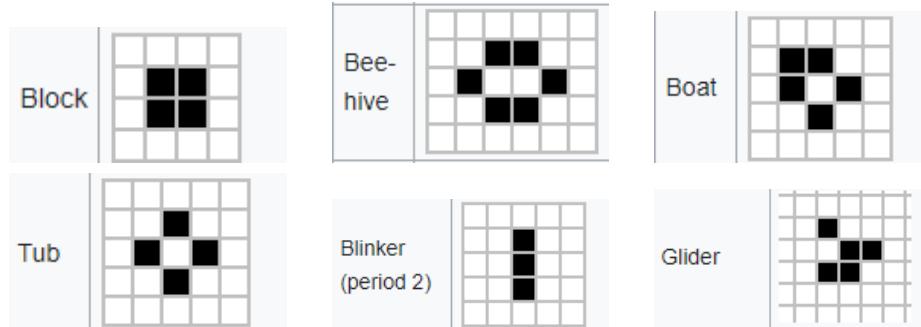
4.4 Knjižnica NumPy

S pomočjo NumPy knjižnice operiramo s podatkovnim strukturami, katere držijo informacije o celicah. Prav tako preko NumPy knjižnice dobimo ven naključni šum in sicer preko Poissonovega šuma. Ta je uporabljen, da vsaki celici dodeli svojo RGB barvo.

4.5 Python implementacija Game of Life

Po prvotnem poskusu implementacije igre, je bil program prepočasen pri večjem številu celic, zato sem se odločil poiskati bolj optimalno rešitev, ki sem jo našel na strani [10]. Preizkusil sem dodati svoja pravila za množenje celic, ampak nisem našel lepega pravila, pri katerem se ne bi preveč hitro celice širile in zapolnile cel ekran. Zaradi tega sem se odločil, da bom pustil ista pravila igre. Torej če ima celica 2 ali 3 sosedje ostane živa. Če ima manj ali več sosedov pa umre. Če ima mrtva celica točno 3 žive sosedje, potem tudi sama ozivi.

Dodaten problem je tudi bil, kako dodati nove točke v igro življenga, tam, kjer se posameznik smeje. Če dodamo samo en piksel v vsako točko ustnic, potem te točke takoj odmrejo. Če zapolnimo celo polje med točkami ustnic, kot da bi obarvali notranji lik postane preveč točk in tudi te takoj odmrejo, saj imajo vse več kot tri sosedje. Zato sem se odločil, da v vsako od točk dodam različne strukture elementov igre življenga, ki vem, da se samostojno zelo zanimivo obnašajo. Nekatere od teh elementov se vidi v sliki 7. Te točke obraza tudi naključno rotiram in zrcalim, da čim bolj povečamo možnost, da se celice razširijo čez celotno površino. Točke v moji implementaciji niso omejene na stranice zaslona, tako da ko pridejo do roba, se takoj preslikajo in nadaljujejo pot iz druge strani zaslona.



Slika 7: Nekaj od elementov igre življenga, ki jih dodajam v točke obrazu.

4.6

5 Zaključek

V tem projektu sem naredili prototipno verzijo inštalacije Smeh življenja. Mislim, da bi z malenkost boljšo stabilnostjo program bil primeren za realno uporabo in bi bila zelo zanimiva in zabavna inštalacija za obiskovalce. Sam sem dokaj zadovoljen s končno podobo projekta. Izboljšave bi bile lahko, da bi po boljši logiki izbirala barve in bi tako ustvaril predvidljive oziroma ponovljive barvne vzorce. Prav tako bi bilo dobro najti neko lastno pravilo za razmnoževanje v igri, ki bi omogočilo večjo površino živih celic in poleg tega večjo barvno površino.

Literatura

1. https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life(Dostopano 9. 1. 2022)
2. <https://var-mar.info/smile/>(Dostopano 9. 1. 2022)
3. Juvan, Samo Solina, Franc Batagelj, Borut Peer, Peter. (2002). 15 sekund slave - interaktivna umetniška inštalacija. <http://www.lrv.fri.uni-lj.si/~peterp/publications/erk02.pdf>
4. <https://zkm.de/en/game-of-life>(Dostopano 9. 1. 2022)
5. <https://www.katevassgalerie.com/game-of-life-emergence-in-generative-art-1>(Dostopano 9. 1. 2022)
6. <https://opencv.org/>(Dostopano 10. 1. 2022)
7. <http://dlib.net/>(Dostopano 10. 1. 2022)
8. <https://docs.python.org/3/library/threading.html>(Dostopano 10. 1. 2022)
9. <https://numpy.org/>(Dostopano 10. 1. 2022)
10. <https://towardsdatascience.com/from-scratch-the-game-of-life-161430453ee3>(Dostopano 10. 1. 2022)