

Maribor, 19.06.2012

**Dokumentacija za prvi del projekta
Projekt - MOBILNE IGRE**

**Implementacija mobilne igre
LS RISK**

Študenta: Dominik Letnar, Tomaž Šuen
Letnik: 2. R-IT (UN)
Študijsko leto: 2011/2012

Vsebina

1. Uvod	3
2. O igri LS Risk	4
3. Algoritem – okrepitve.....	5
4. Algoritem – napadanje.....	5
5. Algoritem – postavitev vojske	6
6. Uporabniški vmesnik	6
7. Generator polja	8
8. Rezultati in zadovoljstvo	11
9. Zaključek.....	12

Kazalo slik

Slika 1: Logotip - Microsoft Visual Studio 2010.....	3
Slika 2: Logotip - Okolje XNA	3
Slika 3: LS Risk.....	4
Slika 4: Uporabniški vmesnik - meni igre	6
Slika 5: Uporabniški vmesnik - nastavitev	7
Slika 6: Uporabniški vmesnik - privzeta mapa.....	7
Slika 7: Uporabniški vmesnik - generirana mapa	7
Slika 8: Uporabniški vmesnik - pogled skupin (držav)	8
Slika 9: Generator polja - korak 1	8
Slika 10: Generator polja - korak 2	9
Slika 11: Generator polja - korak 3	9
Slika 12: Generator polja - korak 4	10
Slika 13: Generator polja - korak 4.....	10

1. Uvod

Pri projektu mobilne igre, je bil cilj izdelati 2D mobilno igro. Odločila sva se, da bova izdelala mobilno igro za Windows Phone, vendar žal ni se izšlo in je nastala namizna verzija. Seveda pa pri tem, sva potrebovala tudi kakšno igro bi naredila in sva si izbrala različico igre Risk, vendar malo prirejeno.

Najbolj pomembne komponente so se izdelale v sklopu predmeta umetna inteligenco:

- povezan graf,
- generator mape,
- algoritem okrepitev in postavitve vojske ter
- algoritem napadanja.

Pri izdelavi te igre, sva uporabila programsko okolje Microsoft Visual Studio 2010 z dodatkom XNA, orodje v katerem lahko delamo igro za Windows Phone, računalnik ali XBOX360.

Ker je igra bolj zahtevne narave in sva imela probleme v mobilni različici, sva igro preprosto pustila za računalniško igro (torej verzija igre), tako da imava vizijo za naslednje leto določene stvari tudi za izboljšati.



Slika 1: Logotip - Microsoft Visual Studio 2010



Slika 2: Logotip - Okolje XNA

2. O igri LS Risk

Risk je v osnovi strateška namizna igra, ki jo proizvaja Hasbro. Izumil jo je francoski filmski režiser Albert Lamorisse in je bila originalno izdana leta 1957 kot La Conquête du Monde ("The Conquest of the World") v Franciji. Risk je potezna igra namenjena 2 - 6 igralcem. Standardna verzija je igrana na politični mapi zemlje in je razdeljena na 42 teritorijev kateri so grupirani v 6 kontinentov. Cilj igre je prevladati svetu oz. zavzeti vse teritorije in posredno s tem odstranimo vse ostale igralce iz igralne površine, jih ubijemo. Igralci nadzirajo napadanje vojske z metanjem kock.

Seveda sva midva igro malo spremenila, tako, da je zadeva sledeča:

- naredila sva skupine, ki predstavljajo določena območja, ki potem dodajo enote, ko osvojiš celotno skupino
- v ozadju ni metanja kock, vendar je preprosto ubijanje v razmerju 1:1 z malenkostnimi pogoji
- v splošnem je zadeva zelo podobna, je pa dejansko igro Risk, igrati dosti težje, kot se izkaže na koncu po igranju najine igre



Slika 3: LS Risk

3. Algoritem – okrepitve

Algoritem ne temelji na nobenem znanem algoritmu, saj je bolj težko oceniti položaj ali simulirati igro do konca, ker je na voljo preveč potez. Odločil sem se zadevo poenostaviti in spisati svoj algoritem.

Algoritem deluje na naslednji način:

- naključno razdelimo proizvodnjo, ki je na voljo, na ozemlja, trenutnega igralca
- shranimo si postavitev
- izvedemo simulacijo napada in štejemo število zavzetih ozemljij
- če je število zavzetih ozemljij večje od največ zavzetih ozemljij, potem si shranimo postavitev, kot najboljšo potezo
- algoritem se izvaja 400 ms (skrajšano iz 1 s, zaradi testiranja).

Simulacija okrepitev, deluje na enak način, kot algoritem za okrepitve, le da vrača novo igro, torej začasen objekt igre.

4. Algoritem – napadanje

Algoritem ne temelji na nobenem znanem algoritmu, saj je bolj težko oceniti položaj ali simulirati igro do konca, ker je na voljo preveč potez. Odločil sem se zadevo poenostaviti in spisati svoj algoritem.

Algoritem deluje na naslednji način:

- nad trenutnim stanjem izvedemo simulacijo napada
- nato izvedemo nad tem stanjem, ki ga vrne simulacija napada, še postavitev enot
- shranimo si stanje kot najboljšo potezo
- potem pa naredimo preprosto simulacijo
 - zamenjamo igralca na potezi
 - izvedemo vse tri faze igre (okrepitve, napadanje, postavitev enot)
 - in to zadevo ponovimo, za vsakega igralca, ki je v igri
- po končani simulaciji, pogledamo, če imamo več naselij kot v primeru najboljše poteze, če imamo si shranimo kot to najboljšo potezo, v nasprotnem primeru, zadevo ponovimo
- algoritem se izvaja 350 ms (skrčeno iz 500 ms, zaradi testiranja).

Simulacija napada deluje tako, da gremo skozi vsa ozemlja trenutnega igralca in izvajamo napade, dokler jih lahko. To pomeni, da se napadi izvajajo glede na število enot na posameznem ozemlju. Za osvojitev ozemlja, potrebujemo več enot, kot jih ima branilčevno naselje.

5. Algoritem – postavitev vojske

Algoritem ne temelji na nobenem znanem algoritmu, saj je bolj težko oceniti položaj ali simulirati igro do konca, ker je na voljo preveč potez. Odločil sem se zadevo poenostaviti in spisati svoj algoritem.

Algoritem deluje na naslednji način:

- vzamemo vsako ozemlje, ki ga ima trenutni igralec na potezi in pogledamo, če je kateri od sosedov, njegovo ozemlje, saj lahko le v tem primeru premikamo enote
- če ima trenutno ozemlje, tudi sosed, katerega lastnik je trenutni igralec na potezi, potem poiščemo najboljša sosedja, teh dveh ozemljij
- potem preverimo nekaj pogojev in primerno razdelimo vojsko:
 - o najboljša sosedja sta enaka, seštejemo enote, pogledamo, če damo enote na polovico, če je to več kot ima sosed, če je razdelimo tako, če ni preverimo ali imamo več enot na obeh naselij kot sosed, potem zaščitimo eno ozemlje zelo, drugo malo manj, če ne, razdelimo naključno
 - o če imamo na enem ozemlju več, na drugem manj enot, potem pogledamo kolikšna je razlika in poskusimo, da bo razlika vsaj blizu 0
 - o če imamo zelo malo enot in je razlika vseh enot negativna, potem se potrudimo zaščiti vsaj eno naselje, če še to ne gre, naključno razdelimo enote na ozemlja

Simulacija postavitve enot, deluje na enak način, kot algoritem za okrepitve, le da vrača novo igro, torej začasen objekt igre.

6. Uporabniški vmesnik

Uporabniški vmesnik igra pomembno vlogo pri takšni aplikaciji oziroma igri. Če se na uporabniškem vmesniku pojavi preveč barv, je moteč faktor, zato nisva dosti komplikirala.

Meni igre izgleda, tako:



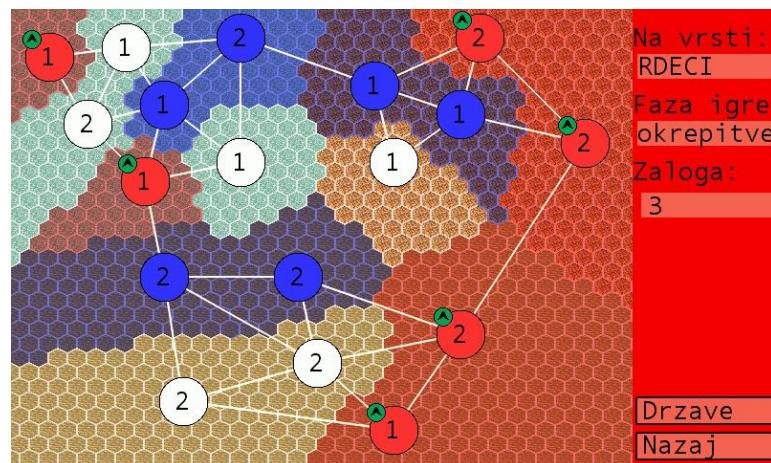
Slika 4: Uporabniški vmesnik - meni igre

Lahko pa odpremo tudi nastavite, ki izgledajo takole:

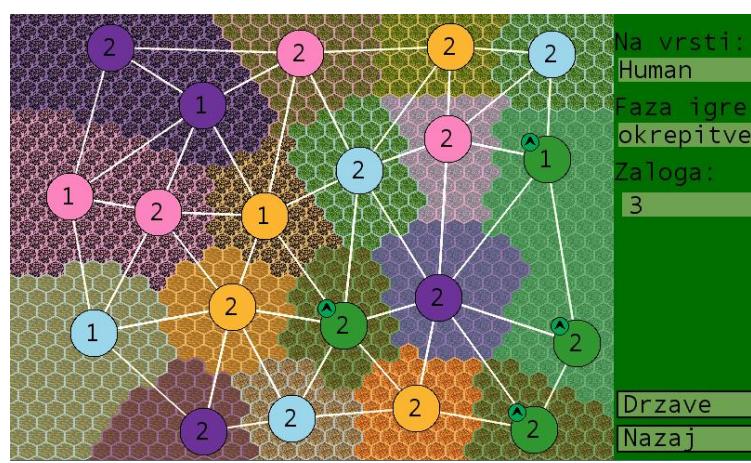


Slika 5: Uporabniški vmesnik - nastavite

Poglejmo si še vmesnik igre, ki izgleda tako:

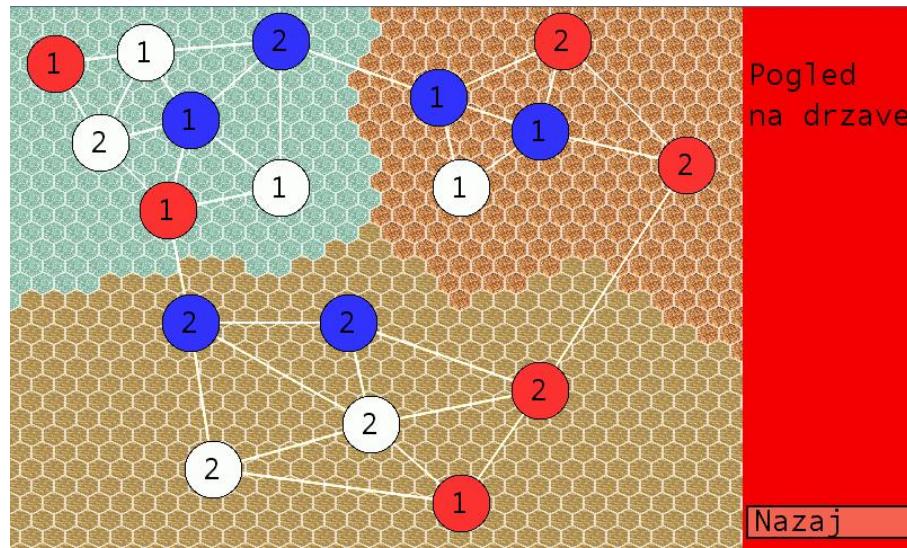


Slika 6: Uporabniški vmesnik - privzeta mapa



Slika 7: Uporabniški vmesnik - generirana mapa

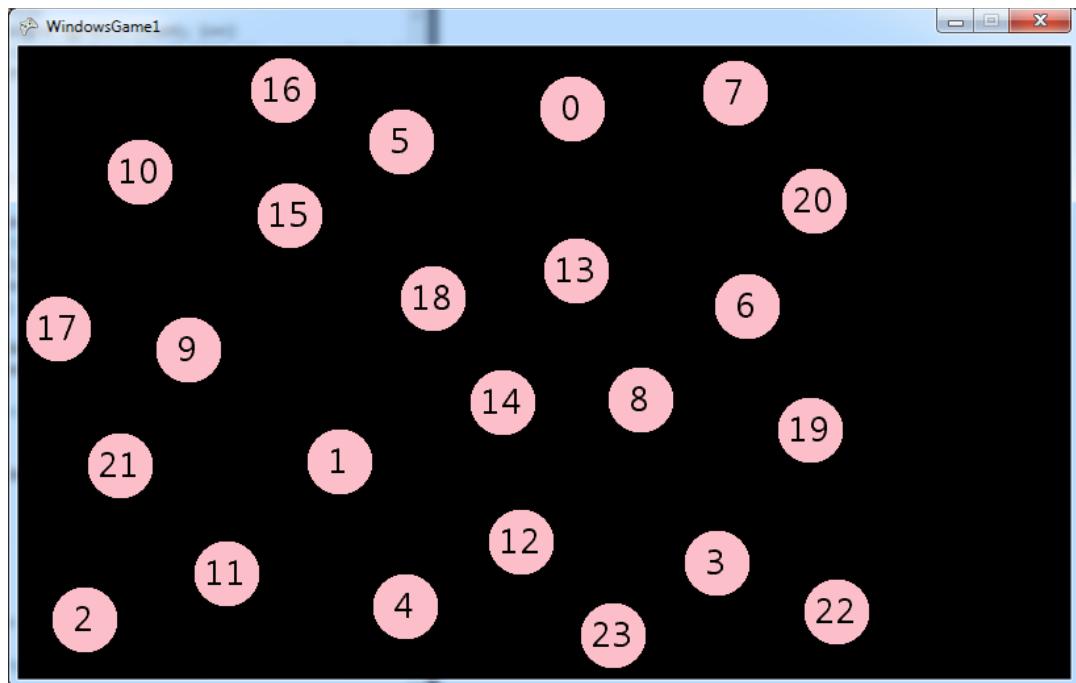
Poglejmo si še, kako izgledajo skupine (države):



Slika 8: Uporabniški vmesnik - pogled skupin (držav)

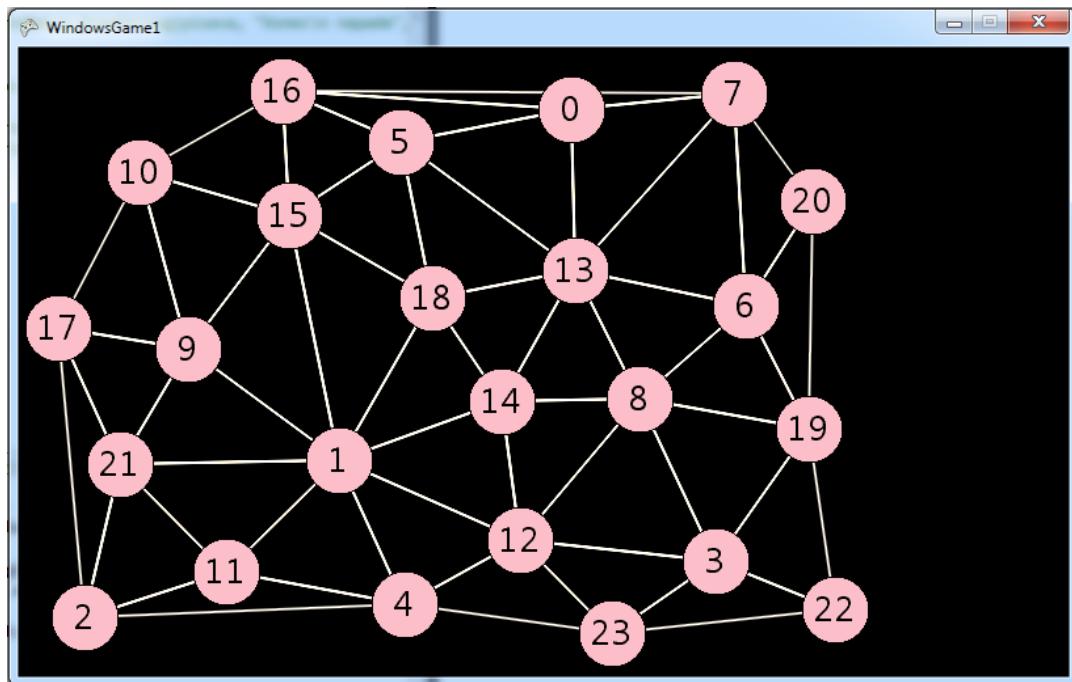
7. Generator polja

V prvem koraku razporedimo polja tako da so med sabo oddaljena za manj kot 100 pikslov.



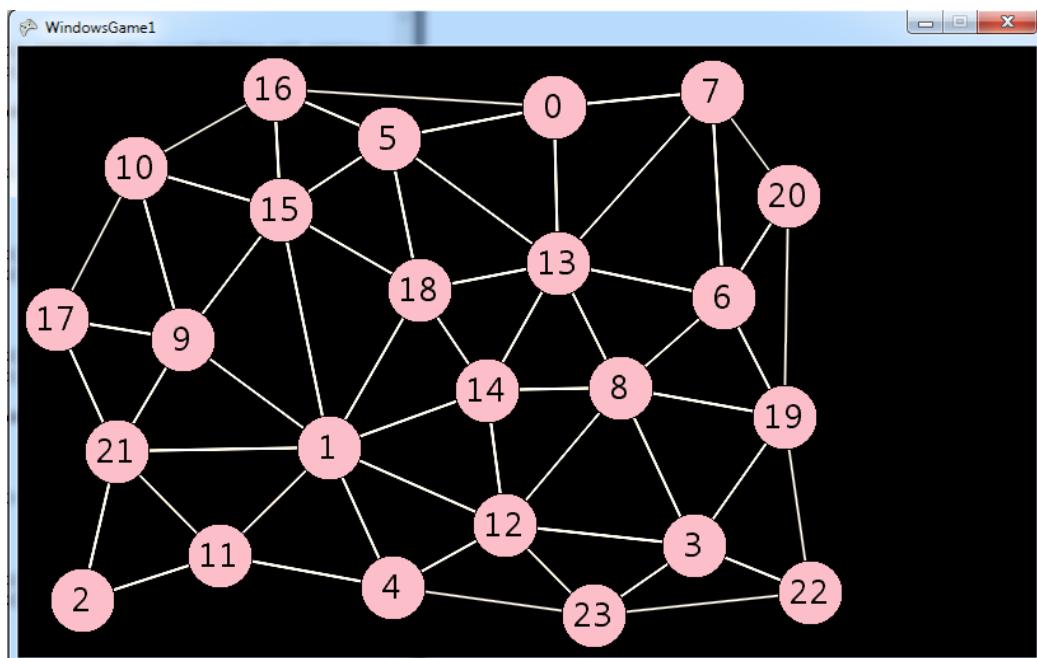
Slika 9: Generator polja - korak 1

Nad njimi naredimo Delaunay-evo triangulacijo in s tem ustvarimo povezave med mesti.



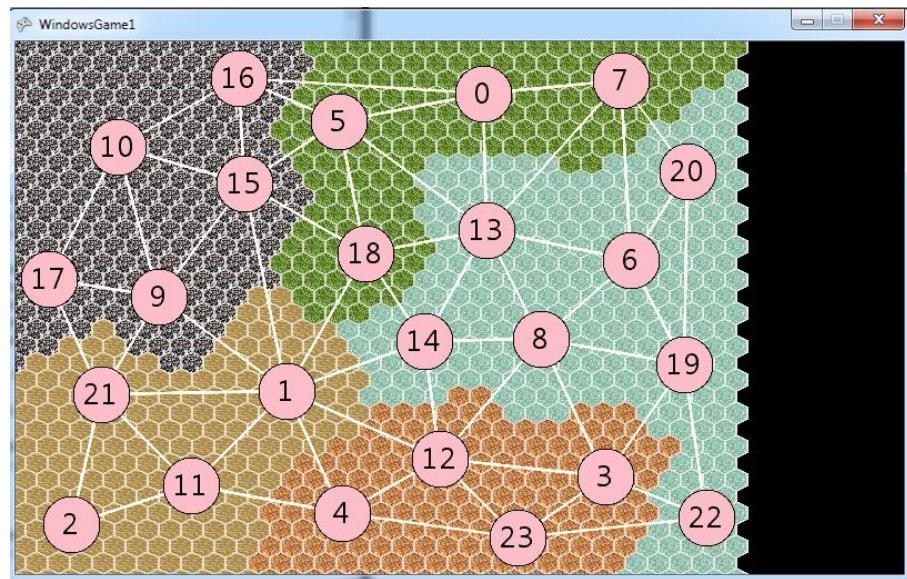
Slika 10: Generator polja - korak 2

Ker vse povezave niso potrebne in tudi ne zgledajo lepo jih nekaj odstranimo.



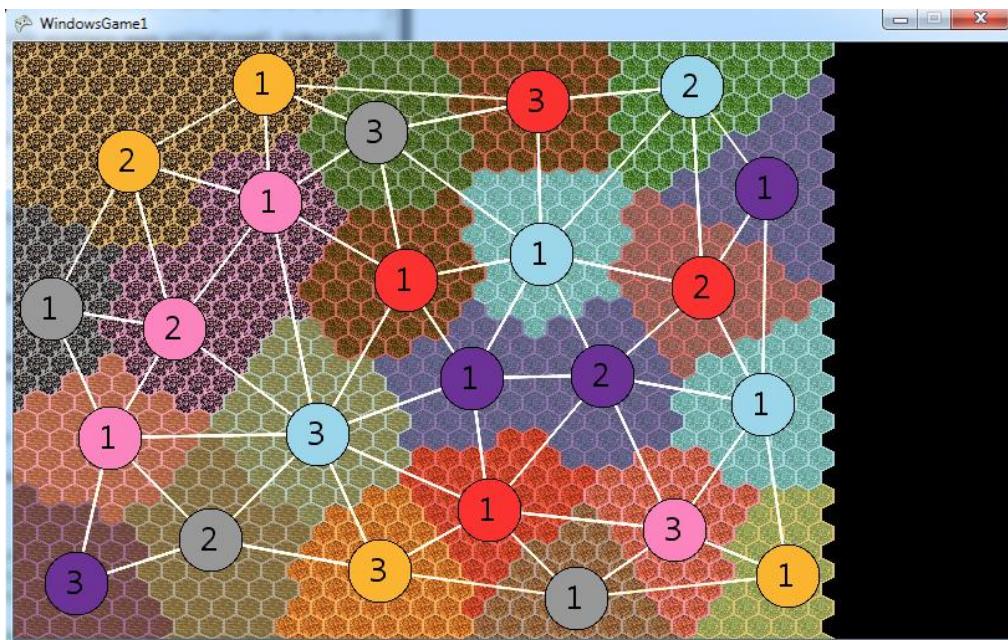
Slika 11: Generator polja - korak 3

Sledi dodelitev mest v države (skupine) in sicer to naredimo z gručenjem. Gručenje naredimo tako da se ustvari na sredini polja krog s premerom višine in na krožnico enakomerno porazdelimo n točk (število skupin, ki ji želimo imeti). Nato gremo čez vsa mesta in gledamo kateri točki so najbližji in tisti, ki je najbližje tej točki, dodani v tisto skupino. Enako počnemo s šest-kotniki. Gremo čez vse in gledamo katero je najbližje mesto in tistem dodelimo trenutni šest-kotnik, lahko bi rekli, da ustvarimo Voronojev diagram.



Slika 12: Generator polja - korak 4

Na koncu še samo dodelimo mesta igralcem in dodamo jim za začetek nekaj vojske. Državam (skupinam) pa dodelimo proizvodnjo po formuli $\text{število mest} \times \text{skupini polovic}$.



Slika 13: Generator polja - korak 4

8. Rezultati in zadovoljstvo

Končni rezultati so seveda zadovoljivi. Igra deluje, ni tako slab izgled, vendar je pa bilo zelo malo časa, kar bo seveda nov izziv za naslednje leto. Projekt po prvem delu izgleda super, sva zadovoljna, vendar želiva še veliko stvari izboljšati. Izboljšati bo potrebno algoritme, ter sam izgled igre, da bo še bolj privlačen.

9. Zaključek

Z narejeno aplikacijo oziroma mobilno igro sva zelo zadovoljna, saj sva sama sebi pokazala in dokazala, da ni bilo tako hudo kot je izgledalo na začetku. Za boljši izgled, boljše algoritme in še mogoče katere pomanjkljivosti, je žal zmanjkalo časa, vendar naju to ne bo ustavilo, saj želiva drugo leto vse skupaj izboljšati in že imava zamisli kako in kaj bi nadaljevala najino delo.