Jelgavas novads

Elejas vidusskola

Riču račs

Zinātniskās pētniecības darbs programmēšanā

Darba autors: *11. klases skolnieks* ***Emīls Drošprāts***

Darba vadītājs: *datorikas skolotājs* ***Jānis Tumovs***

Eleja 2022

Anotācija

Zinātniskās pētniecības darbu “Riču račs” izstrādāja Elejas vidusskolas 11. klases skolēns Emīls Drošprāts. Darbā ir 13 lapas, 3 nodaļas, 1 tabula, 11 attēli un 2 pielikumi.

Pētījuma galvenie rezultāti ir riču rača latviešu versijas izcelsmes izpēte un datorprogrammas izveide riču rača spēlēšanai.

Būtiskākais secinājums – ar JavaScript palīdzību iesācējs programmēšanā ar darba vadītāja atbalstu, var izveidot vienkāršu datorspēli.

Atslēgas vārdi: JavaScript, programmēšana, skripti, mājaslapas, galda spēles.

Annotation

Eleja Secondary School 11th grade pupil Emīls Drošprāts developed the research work “Riču račs”. The work includes 13 pages, 3 chapters, 1 table, 11 pictures and 2 attachments.

The main results of the study are the exploration of the origins of the Latvian version of the riču račs and the creation of a computer program for the playing of the riču račs.

The most important conclusion is that with JavaScript, a beginner in programming with the support of a job manager can create a simple computer game.

Keywords: JavaScript, programming, scripts, homepages, board games.

Saturs

Ievads 4
1. Galda spēles ar metamo kauliņu izmantošanu 5
1.1. Ričuračs 5
1.2. Pačisi un ludo 5
2. Datorspēles izstrādes posmi 8
3. Riču rača kods 9
3.1. Programmēšanas vides izvēle 9
3.2. Spēles loga HTML iekārtojums 9
3.3. Spēles algoritmi 10
Secinājumi 12
Literatūras avotu saraksts 13

Ievads

Laikmetā pirms datoru un interneta plašas ienākšanas visās dzīves sfērās – 30 un vairāk gadu senā pagātnē – viena no retajām jaunāko bērnu intelektuālajām izklaidēm bija galda spēles. Latvijas PSR bija nopērkami – "Cirks" un "Ričuračs". Tās ir populāras spēles vēl aizvien. Tādēļ šī ZPD pētāmais jautājums ir noskaidrot:

- ✓ Vai drukātos avotos un tīmeklī ir atrodama informācija par ričuraču vēsturi?
- ✓ Vai iespējams izveidot datorā spēlējamu ričuraču?

Pētījuma mērķis ir programmēšanas valodā JavaScript radīt pārlūkprogrammā darbināmu spēli "Ričuračs".

Mērķa sasniegšanai tika izvirzīti šādi uzdevumi:

1. Analizēt informāciju par ričurača un tam līdzīgu spēļu vēsturi un noteikumiem,
2. Aprakstīt dizaina procesa soļus datorspēles izstrādē,
3. Izveidot darbotiesspējīgu kodu ričurača spēlēšanai,
4. Informāciju par ričurača latviešu variantu ievietot tiešsaistes enciklopēdijā.
5. Izveidot izdrukājamus spēles laukuma izklājumus.

1. Galda spēles ar metamo kauliņu izmantošanu

Daudzās galda spēlēs gājienu pozīcijas nosaka ar metamajiem kauliņiem. Klasisko kubveida kauliņu skaldnēs atrodas iekrāsoti punkti, kas atbilst skaitļiem no viena līdz seši. Galda spēlēs izmanto dažādu skaitu metamo kauliņu. Visbiežāk vienu, bet, piemēram, spēlē *yatzy* – piecus. Šī spēle ar nosaukumu "Pokers" Latvijā bija populāra 20. gadsimta 80-jos gados.

Kopš pagājušā gadsimta vidus Latvijā spēlē "Cirku" un "Ričuraču". To autors uzzināja no vecākajiem ģimenes locekļiem.

Spēles ar metamajiem kauliņiem dod ieguldījumu bērnu intelektuālajā attīstībā un rakstura veidošanā:

- ✓ tās ir izglītojošas,
- ✓ māca izprast un ievērot spēles noteikumus,
- ✓ attīsta skaitīšanas iemaņas,
- ✓ rosina domāt stratēģiski un plānot darbības,
- ✓ māca izjust emocijas uzvaras vai zaudējuma gadījumā.

1.1. Ričuračs

Pētījuma autors drukātajos avotos neatrada atsauci uz spēli ar nosaukumu "ričuračs", ieskaitot pirmās brīvvalsts Konversācijas vārdnīcu, LPSR un mūsdienu Latvijas enciklopēdijas. Turpinot meklējumus tīmeklī, ričuračam līdzīgi izrunājamus spēļu nosaukumus[1] izdevās atrast krievu[2] valodā: *ричи-рače, рич-рачи, рич-рач*. Pēc šo avotu apgalvotā 20. gadsimta 20-jos gados spēli ar tādu nosaukumu Padomju Krievijā zināja un arī spēlēja.

Autora hipotēze ir, ka latviskais riču rača nosaukums ir aizgūts no krievu valodas, bet dizains no *ludo* spēles vācu varianta (skat. nodaļu 1.2.).

Ričurača spēlē var piedalīties divi līdz četri spēlētāji, kas izlozē kauliņu krāsu un pirmā gājiena tiesības. Sākumā spēlētāju kauliņu komplekti atrodas mājas lauciņos. Uz starta lauciņu iziet, ja uzmet 1 vai 6. Sešinieka gadījumā met un iet vēlreiz. Spēles uzdevums ir kauliņus novietot uz pēdējiem 4 savas krāsas lauciņiem krusta iekšpusē. Ja kārtējais gājiens beidzas uz lauciņa, ko aizņem pretinieks, tad kauliņš tiek sists un pārvietots uz māju. Uz viena lauciņa var atrasties vairākas vienas krāsas figūras, kas veido barjeru pretinieka figūrām. Ja beigās uzmestais punktu skaits ir lielāks par nepieciešamo, atlikums jāpaiet atpakaļ.

Veidojot datorprogrammu, autors noteikumus ir vienkāršojis, izslēdzot pēdējos divus nosacījumus. Riču rača interaktīvajā variantā nevarēs paiet uz lauciņu, kur jau atrodas savs kauliņš. Iemesls ir lauciņu attēlojuma veids – tikai vienā dimensijā, bet ne telpiski. "Atsitiens" pret lauciņu ceļa galu arī nebūs atļauts, lai nesarežģītu programmējamos algoritmus.

1.2. Pačisi un ludo

Ričuraču priekštece ir sena Indijas spēle pačisi. Indijas valdnieks Akbars I bija iecienījis to spēlēt ar dzīviem kauliņiem – 16 skaistulēm no sava harēma, kas pārvietojas pa laukumu atkarībā no tā kā viņš uzmeta kauri gliemežvāku. Tā laika marmora spēļu laukumi saglabājušies līdz mūsdienām. Pačisi ir tik populārs, ka to dēvē par indiešu nacionālo spēli[3].

Pačisi nosaukums hindi valodā nozīmē 25. Tas ir maksimālais punktu skaits, ko var uzmest. Pačisi radies 6. gadsimtā. Sākotnējā versija aprakstīta arī indiešu eposā Mahabharata[4].

[1] https://didacts.ru/termin/richi-rache.html
[2] http://igraydoma.ru/igri_rich.html
[3] https://en.wikipedia.org/wiki/Pachisi
[4] https://www.ludoleague.in/blog/Take-a-Look-on-The-History-of-Ludo-Game/34

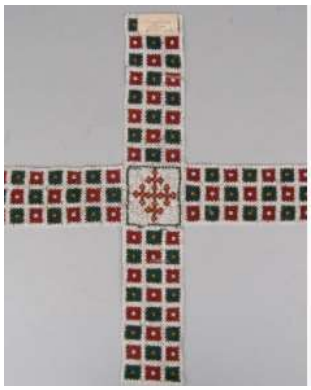
1938. gadā amerikāņu rotaļlietu un spēļu uzņēmums "Transogram" izlaida pārdošanā spēli ar nosaukumu "Game of India", ko vēlāk pārdēvēja par "Pa-Chiz-Si: The Game of India". Mūsdienīgu spēles pačisi komplektu var apskatīt 1. attēlā.

1. attēls. Pačisi.

2.attēls. Kauri[5] gliemežvāki.

Tradicionālajā indiešu pačisi spēlētāju metamie kauliņi bija 6 vai 7 kauri gliemežvāki (skat. 2. att.). Spēli sāka lielākā skaitļa uzmetējs, un kustība notika pulksteņa rādītāja virzienā.

Britu muzeja ekspozīcijā pačisi iekļuvusi starp 10 slavenākajām vēsturiskajām spēlēm. Attēlos zemāk aplūkojami 19. gadsimtā Indijā un Šrilankā veidotie pačisi no muzeja krājumiem[6].



3. attēls. *Pahisi* no Britu muzeja.

Cits riču rača priekšteča nosaukums ir ludo. Vārds "ludo" atvasināts no latīņu valodas vārdiem ar nozīmi spēlēt – *ludere* un spēle – *ludus*.

Ludo patentēja Anglijā 1896. gadā. Tajā izmantoja kubveida metamo kauliņu, bet spēles laukumu ievietoja kvadrātveida galdiņā (skat. 4. att.).

4. attēls. Ludo.

Ludo līdzīgas spēles ar citiem nosaukumiem ir daudzām pasaules tautām. Īpaši populāra kļuva vācu "*Mensch ärgere Dich nicht*", kas latviski nozīmē "Nedusmojies!" – acīmredzot par tava kauliņa sišanu.

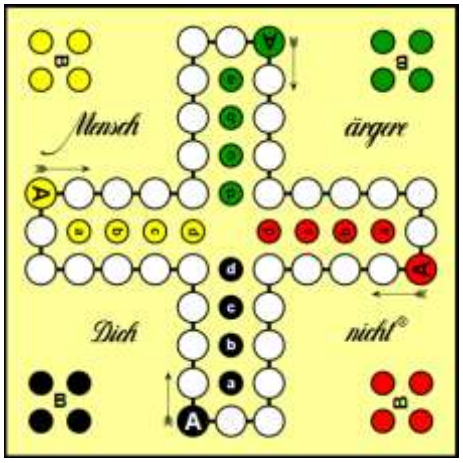
[5]https://www.amazon.com/PEPPERLONELY-Polished-Tiger-Cowrie-Shells/dp/B01N1JQ7CO?th=1
[6]https://blog.britishmuseum.org/top-10-historical-board-games/

Šo spēli izveidoja Jozefs Frīdrihs Šmits savā darbnīcā Bavārijā netālu no Minhenes. 1914. gadā bija pārdoti jau aptuveni 70 miljoni eksemplāru daudzās Eiropas valstīs. Mūsdienās tā joprojām ir vispopulārākā ģimenes galda spēle.

Lai arī vācu varianta pamatā bija angļu ludo, Šmits atteicās no ludo un pačisi ietekmes noteikumos, kā arī no dizaina izņēma to izcelsmes simboliku.

Vācu spēle kļuva īpaši populāra Pirmā pasaules kara laikā. Šmits nosūtīja uz lauka hospitāļiem 3000 spēļu, lai slimnieku veseļošanās laiku padarītu jautrāku. Pateicoties šādam mārketingam un reklāmai no mutes mutē, līdz 1920. gadam pārdeva miljonu spēļu komplektu par 35 feniņiem gabalā[7].

Mūsdienu *Mensch ärgere Dich nicht* (skat. 5. att.) dizaina ziņā līdzinās latviešu ričuračam. Atšķiras krusta izmērs, orientācija un kauliņu krāsas.



5. attēls. Vācu spēle[8] *Mensch ärgere Dich nicht*

[7]https://de.wikipedia.org/wiki/Mensch_%C3%A4rgere_Dich_nicht
[8]https://upload.wikimedia.org/wikipedia/commons/a/a6/Mensch_%C3%A4rgere_dich_nicht_4.svg

2. Datorspēles izstrādes posmi

Datorspēles izveide ir ilga un sarežģīta, jo darba gaitā jāīsteno dizaina procesa soļi, līdzīgi kā radot jebkuru jaunu izstrādājumu. Tikai šoreiz rezultāts nav vis materiāla vērtība, bet gan intelektuāls produkts.

Pēc skola2030 nostādnēm, veicot lielus projektus, būtu jāiziet diagrammā attēlotie soļi (skat. 6. att.).

6. attēls. Dizaina procesa[9] soļi.

Šī ZPD autors ir izvēlējies programmēšanas priekšmetu. Darba vadītāja piedāvājums izveidot datorspēli autoram šķita interesants un noderīgs personiskajā izaugsmē.

Autors nolēma programmēt ričuraču, jo tam ir vienkārša grafika – tikai krāsaini aplīši noteiktās ekrāna vietās. Arī gājienu izdarīšanu nav grūti attēlot, ja nomaina aplīšu krāsu. Autors izpētījis, ka internetā spēlējamu ričuraču latviešu versiju līdz šim neviens nav publicējis un arī neeksistē tās apraksts tiešsaistes enciklopēdijā Wikipedia. Tā kā autors skolā sācis apgūt JavaScript pamatus, tad tieši šajā programmēšanas vidē notika spēles koda rakstīšana. Kods tiek darbināts HTML dokumentā caur jebkuru pārlūkprogrammu.

Plānojot datorspēli, vispirms jāformulē spēles noteikumi un paredzamais spēlētāju vecums. Autors klasiskos riču rača noteikumus ir minimāli vienkāršojis, lai spēles algoritmi būtu vieglāk programmējami.

Tālāk bija jāizdomā, kā izskatīsies spēles laukums un kā to attēlos ekrānā ar grafiskajām komandām.

Mūsdienu datorspēlēs parasti iespējams sacensties ar "mākslīgo intelektu" jeb spēles radītāja izdomātu algoritmu. Riču račā tas ir svarīgi, jo nebūtu interesanti, ka pie viena datora pēc kārtas nāktu spēlētāji un izdarītu gājienus. Tādēļ vispirms bija jāizveido datora – kā pretinieka – spēles modelis. Kad tas sekmīgi darbojās atbilstoši spēles noteikumiem, varēja ķerties pie grafiskās saskarnes radīšanas, lai programma uztvertu datorpeles notikumus: pārvietošanu, klikšķināšanu u.tml. Cilvēka darbībām spēles laukumā jāatbilst tās noteikumiem.

Lai datorspēle vispār kādu piesaistītu, tai jādod emocionāls gandarījums, piemēram, pēc uzvaras pret datoru. Riču račā tas ir iespējams, jo neeksistē uzvarošas stratēģijas. Var spēlēt "gudrāk", vai "ne tik gudri", tomēr visu noteiks metamā kauliņa nejauši uzkritušie cipari. Nav šaubu, ka koncentrēšanās un uzmanības noturība riču rača spēlēšanā ir nozīmīga.

Kad spēle darbojas atbilstoši autora iecerei, to ir vērts dot izmēģināt reāliem spēlētājiem, lai atklātu iespējamās kļūdas un uzzinātu spēlētāju viedokli par izstrādājuma kvalitāti. ZPD autors piedāvāja spēli notestēt gan klasesbiedriem, gan arī desmitklasniekiem, kas Elejas vidusskolā arī apgūst programmēšanas priekšmetu. Ar aizrautību ričuraču spēlēja visi 8.b klases skolēni.

Kad datorprogrammā veikti vajadzīgie uzlabojumi, dizaina procesu noslēdz izstrādājuma ieviešana, jeb šajā gadījumā – publicēšana tīmeklī, kas ļauj spēli izmantot jebkuram interesentam. Spēle būs pieejama skolas mājaslapas sadaļā ar skolēnu IT tēmām veltītajiem projektiem www.elejasvsk.lv/zpd.php.

[9] Attēls no datorikas programmas 1.–9. klasei.

3. Riču rača kods

ZPD autora radītais kods satur 500 rindas ar tekstu un apmēram 12 000 rakstzīmju. Kodā ir HTML objekti – komandpogas, izvēles rūtiņas, sadaļu un rindkopu birkas un arī kvadrāts zīmēšanai, ko sauc par kanvu. Stilu sadaļā norādītas krāsas, izmēri, rakstzīmju veids un lielums, apmaļu un atkāpju platumi. Paša skripta kods ietver konstanšu un mainīgo aprakstu, kā arī 13 autora izveidotas JavaScript funkcijas, kas veic dažādus uzdevumus, lai ričurača spēle darbotos.

3.1. Programmēšanas vides izvēle

Darba autors izvēlējās JavaScript valodu spēles koda veidošanai, jo atšķirībā no komplicētajām programmēšanas vidēm C++ un VB.NET, šeit var iztikt pat ar visvienkāršāko teksta redaktoru, piemēram Notepad. Darba produktivitātes nolūkā autors izmantojis bezmaksas lietotni Code Writer.

Pēc autora domām JavaScript valoda ir pietiekami viegla, lai to spētu saprast un izmantot jebkurš mājaslapu veidotājs, kurš pārzina HTML un CSS[10]. JavaScript kods labi sadarbojas ar HTML objektiem un CSS stiliem. Izmantojot JavaScript, var iegūt informāciju no mājaslapas apmeklētāja datora, piemēram, pārlūkprogrammas versiju un ekrāna izšķirtspēju. Tādējādi lapas veidotājs ar JavaScript palīdzību pielāgo savu programmu dažādu lietotāju datoriem, izvairoties no mājaslapas attēlošanas kļūdām.

Tā kā JavaScript tika attīstīts līdz ar HTML valodu, kurā veidotas visas tīmekļa lapas, tad šādi skripti tajās darbojas efektīvi un ļoti ātrdarbīgi.

3.2. Spēles loga HTML iekārtojums

Spēles loga kreisajā malā apskatāma informācija par spēlētājiem (skat. 7. att.), gājiena kārtu, uzmesto ciparu (skat. 8. att.) un statistika par visu krāsu gājieniem – kāds ir noietā ceļa garums, cik reižu sisti pretinieka kauliņu un cik reižu tā noticis pašam (skat. 9. att.).

Informatīvajā rindā sākumā lasāms – "izvēlies pretiniekus", beigās tiek nosaukts uzvarētājs, rakstot to ar attiecīgās krāsas burtiem.

Spēli darbina ar trim komandpogām – "Spēlēt", "Gājiens" un "Beigt". Dažādos spēles posmos mainās to ieslēgšanas iespēja. Neaktīvie objekti attēlojas blāvākā krāsā (skat. 10. att.).

7. attēls. Spēlētāju izvēle.

8. attēls. Gājiena kārta.

9. attēls. Gājienu statistika.

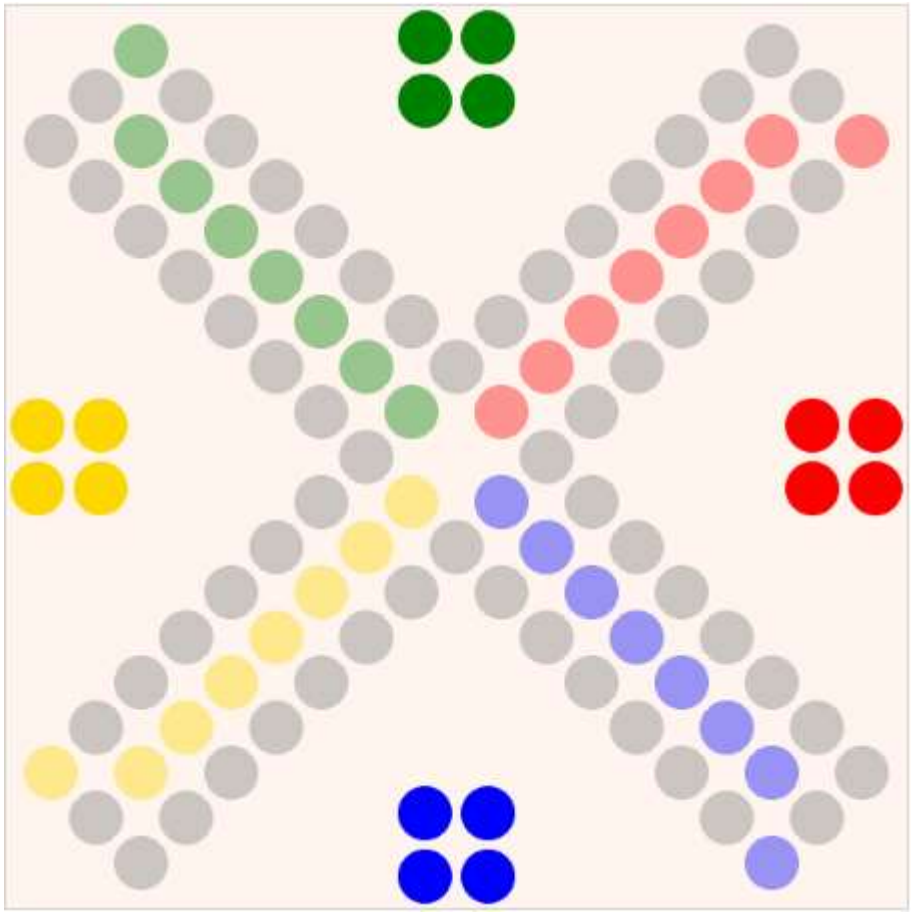
10. attēls. Komandpogas.

[10]CSS (*Cascading Style Sheets*) – mājaslapu noformēšana atsevišķi no satura

11. attēlā redzams spēles sākuma skats. Lauciņi, uz kuriem neatrodas neviens kauliņš, ir bālākā krāsā (40% no maksimālā spilgtuma). Sākumā spēlētāju 4 kauliņu komplekti atrodas mājas lauciņos. Mājai tuvāko krusta stūra lauciņu virzienā, kas pretējs pulksteņa rādītāja kustībai, sauc par starta lauciņu. Kad kauliņš pa pelēkās krāsas ceļu apgājis visu krusta ārējo kontūru, tas atkal nonāk starta lauciņā, lai ceļu turpinātu uz krusta iekšpusi pa savas krāsas ceļu. Pretinieki nevar sist kauliņus uz starta lauciņa.

Kauliņam, izejot no mājas, tā vieta tiek aizpildīta fona krāsā. Tādēļ jebkurā spēles momentā jābūt redzamiem tieši 4 sarkaniem, zaļiem, ziliem un dzelteniem spilgtākas krāsas aplīšiem.

Laukuma iekārtošana notiek HTML objektā kanva, ko ievieto ar birku <canva>. Pašus aplīšus zīmē JavaScript funkcija arc(), kurai norāda loka grādus radiānos, centra koordinātas un rādiusu ekrāna pikseļos[11].



11. attēls. Riču rača spēles sākuma skats.

3.3. Spēles algoritmi

Autors kopējo problēmu – izveidot spēli – sadalīja mazākos uzdevumos, kuru izpildi nodrošina JavaScript valodas lietotāja funkcijas. 1. tabulā aprakstīta šo funkciju nozīme. Dažas no šīm funkcijām ne tikai izpilda savu uzdevumu, bet arī izsauc citu funkciju izpildi atkarībā no spēles situācijas.

[11] *pixel (picture element)* – angl. ir mazākais ekrāna gabaliņš, ko var apstrādāt aparatūra

1. tabula. Riču rača apakšprogrammas.

Nosaukums kodā	Funkcijas uzdevums
zime_laukumu()	Uzzīmē spēles laukuma sākuma pozīciju un izvada krāsu sadalījumu spēlētājiem (dators vai cilvēks).
zime_apliti()	Atbilstoši koordinātu kārtas numuram uzzīmē vienu aizkrāsotu spēles laukuma aplīti.
gajiena_info()	Uzmet kauliņu, informē par gājiena kārtu, izsauc funkciju, kas nosaka kauliņu iespējamos gājienus. Ja gājiens neeksistē, to nodod nākamajai krāsai.
iespejami_gajieni()	Noskaidro, vai vispār ir gājiens, un, ja ir, tad kāda veida – no mājas, vai no krusta lauciņiem.
gajiena_izvele()	Meklē labāko gājienu spēlētājam, kurš ir dators, pēc šāda algoritma: 1) nokauj pretinieka kauliņu un nosūta to uz māju, 2) Ja uzkritis 1-nieks, iziet no mājas, ja iespējams; ja uzkritis 6-nieks, no mājas iziet tikai tad, ja cita gājiena nav, 3) ieiet krusta drošajā (savas krāsas) lauciņā, 4) iet ar mazākā numura kauliņu nedrošajā (pelēkajā) krusta daļā.
gajiens()	Dators veic gājienu pēc algoritma. Atceļ cilvēka nepareizus gājienus. Pārbauda, vai gājiens nebeidzas uz pretinieka kauliņa un vai ar pēdējo gājienu kāds nav uzvarējis. Nodod gājiena kārtu nākamajam spēlētājam.
paiet_par_vienu()	Atjauno aplīša veco krāsu un nākošo aplīti nokrāso spēlētāja kārtas numura krāsā.
nosit_citu()	Ja gājiens beidzas uz pretinieka kauliņa, tas tiek "aizsūtīts" uz māju
klikskis()	Izsauc peles klikšķa notikumu spēlētājam – datoram.
stop()	Apstādina spēli pēc uzvaras vai lietotāja izvēles.
koordinatas()	Peles klikšķa ekrāna koordinātas pikseļos pārrēķina aplīšu koordinātās veselos skaitļos attiecībā pret krusta centru.
iedarbina()	Izvada ekrānā gājienu statistikas tabulu un nodod pirmā gājiena kārtu sarkanajiem kauliņiem.
speletajs()	Nofiksē izvēles rūtiņu vērtības sadalījumam datori: cilvēki.
metamais()	Izveido SVG[12] formāta metamo kauliņu ar gājiena krāsas acīm

Autors bija iecerējis izveidot gājienu animācijas, taču izrādījās, ka uz zīmēšanas darbībām JavaScript neattiecas iespēja veikt programmas aizturi uz noteiktu milisekunžu skaitu[13]. Tādēļ spilgtāko krāsu aplīši, kas norāda kauliņu pozīciju, uzreiz pārvietojas attālumā, kas vienāds ar uzmesto punktu daudzumu.

Iemesls ir tas, ka Javascript ekrānu pārzīmē 60 reizes sekundē, un nevar vienkāršā veidā panākt, lai tas notiktu, piemēram, ik pēc 0,1 sekundes.

[12]scalable vector graphic (*angl.*) – attēls, kam nemainās kvalitāte, to mērogojot
[13]https://www.w3schools.com/

Secinājumi

1. Ričurača galda spēles pačisi pirmsākumi meklējami senajā Indijā.
2. Ričurača mūsdienu versijas balstās uz simt gadu senās ludo spēles angļu versiju.
3. Ričurača nosaukums aizgūts no krievu valodas, bet dizains no vācu spēles ludo.
4. Ričuračs piemērots spēlēšanai ģimenē, iesaistot jaunākos bērnus.
5. Ar JavaScript ir iespējams izveidot ričurača matemātisko modeli.
6. Ar JavaScript ir grūti realizējama kauliņu pārvietošanās animācija.

Literatūras avotu saraksts

1. didacts.ru/termin/richi-rache.html

2. E. Glonegers, V. Dims (1982). *Lielā Spēļu Grāmata*. Rīga: Zvaigzne ABC

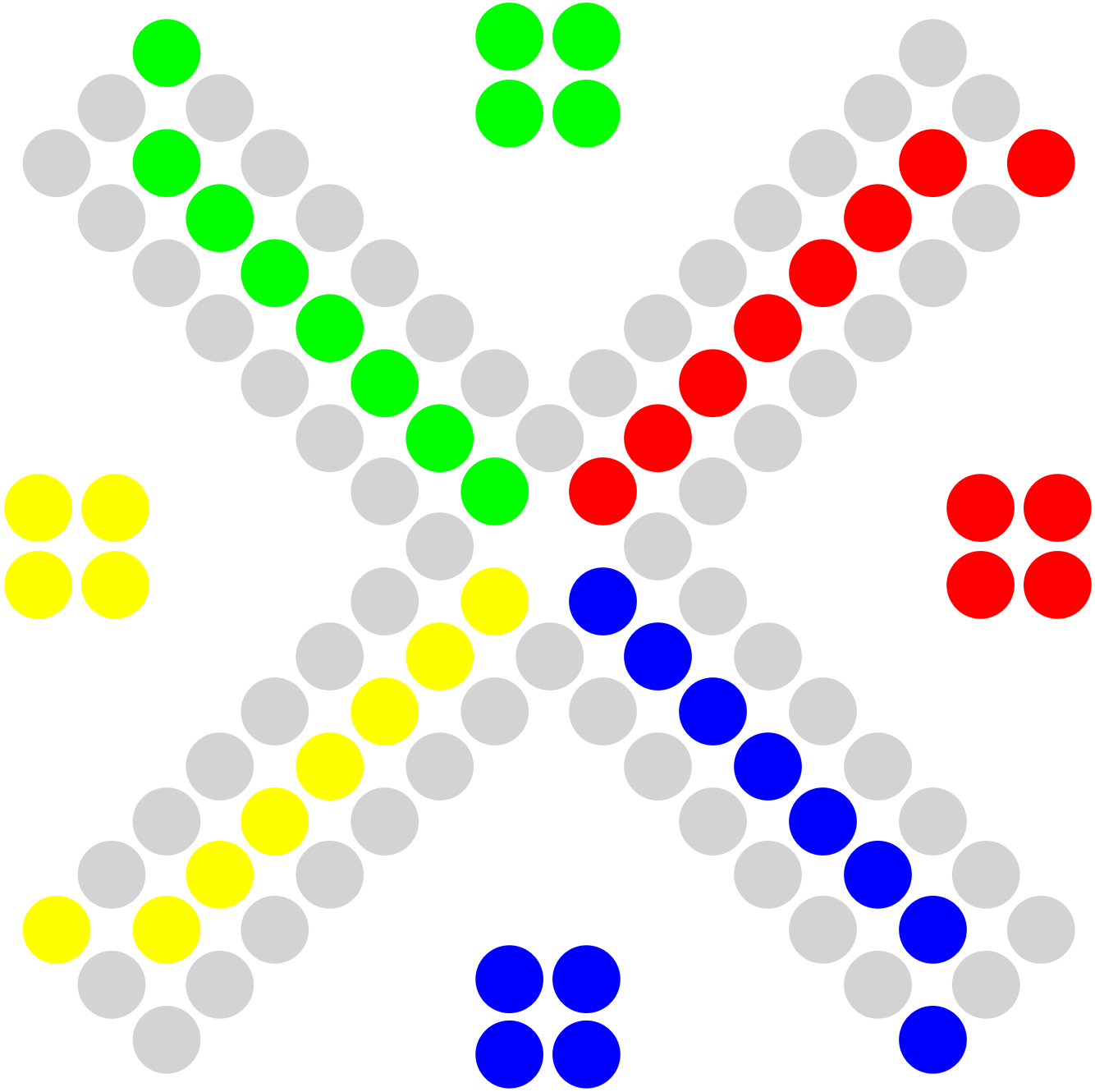
3. History of Ludo Game. Pieejams: ludoleague.in/blog/Take-a-Look-on-The-History-of-Ludo-Game/34

4. JavaScript tutorial. Pieejams: w3schools.com/js/

5. Mensch ärgere Dich nicht. Pieejams: de.wikipedia.org/wiki/Mensch_%C3%A4rgere_Dich_nicht

6. Top 10 historical board games. Pieejams: blog.britishmuseum.org/top-10-historical-board-games/

1. Pielikums



```
<!DOCTYPE html>
<html>
<meta charset="UTF-8">
<head>
	<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
	<style>
		img {padding: 20px 0px 0px 10px;}
		p {margin: 10px 0px 20px 10px;}
		input {margin: 6px 0px 0px 10px; padding: 2px 4px;}
		#datori {margin: 6px 0px 10px 20px;}
		body {font-size: 14px; font-family: arial;}
		table {width: 180px; margin: 20px 0px 0px 15px; padding: 4px;}
		th {padding: 0px 0px 10px 0px; font-size: 16px;}
		td {text-align: center;}
	</style>
</head>
<body>
<div id="datori" style="font-size: 16px;"></div>
<input type="checkbox" id="ch0" checked onClick="speletajs(0)">
<label for="ch0">Sarkanie</label><br>
<input type="checkbox" id="ch1" checked onClick="speletajs(1)">
<label for="ch1">Zaļie</label><br>
<input type="checkbox" id="ch2" checked onClick="speletajs(2)">
<label for="ch2">Dzeltenie</label><br>
<input type="checkbox" id="ch3" checked onClick="speletajs(3)">
<label for="ch3">Zilie</label><br>
<img src="d.gif" name="metamais"><br>
<p id="seciba"></p>
<input id="sac" type="button" value="Spēlēt" onClick="iedarbina()">
<input id="met" type="button" value="Gājiens" onClick="gajiens()" disabled>
<input id="beidz" type="button" value="Pabeigt" onClick="stop()">
<canvas id="kanva"
  width="600" height="600"
  style="border: 1px solid #d3d3d3; background-color: seashell;
  left: 300px; top: 2px; position: absolute;">
 </canvas>
<body onload="zime_laukumu()">
<script>
document.onclick = koordinatas;
const mg=48;
const stura_radiuss=10;
const x0=80;
const y0=20;
const acs_radiuss=5;
const acs_vieta=[[-2,-2],[2,-2],[-2,0],[0,0],[2,0],[-2,2],[2,2]];
const cipars=[[0,0,0,1,0,0,0],[0,1,0,0,0,1,0],[0,1,0,1,0,1,0],[1,1,0,0,0,1,1],[1,1,0,1,0,1,1],[1,1,1,0,1,1,1]];
var svgns = "http://www.w3.org/2000/svg";
var svg = document.createElementNS(svgns, "svg");
svg.setAttributeNS(null,"id","svgDoc");
svg.setAttributeNS(null,"height",y0+mg+1);
svg.setAttributeNS(null,"width",x0+mg+1);
document.getElementsByTagName("body")[0].appendChild(svg);
var svgDoc = document.getElementById("svgDoc");
var taisnsturis = document.createElementNS(svgns, "rect");
taisnsturis.setAttributeNS(null, "x", x0);
taisnsturis.setAttributeNS(null, "y", y0);
taisnsturis.setAttributeNS(null, "width", mg);
taisnsturis.setAttributeNS(null, "height", mg);
taisnsturis.setAttributeNS(null, "rx", stura_radiuss);
```

```
taisnsturis.setAttributeNS(null, "ry", stura_radiuss);
taisnsturis.setAttributeNS(null, "fill", "white");
taisnsturis.setAttributeNS(null, "stroke", "black");
taisnsturis.setAttributeNS(null, "stroke-width", "1");
const pauze=1500; //milisekundes
const spilgtums=0.4;
const krasa=['red','green','gold','blue','gray','seashell'];
const krasaLV=['sarkanajiem','zaļajiem','dzeltenajiem','zilajiem'];
const krusts = [[8,-8],[7,-9],[6,-8],[5,-7],[4,-6],[3,-5],[2,-4],[1,-3],[0,-2],
        [-1,-3],[-2,-4],[-3,-5],[-4,-6],[-5,-7],[-6,-8],[-8,-8],[-9,-7],
        [-8,-6],[-7,-5],[-6,-4],[-5,-3],[-4,-2],[-3,-1],[-2,0],[-3,1],
        [-4,2],[-5,3],[-6,4],[-7,5],[-8,6],[-8,8],[-7,9],
        [-6,8],[-5,7],[-4,6],[-3,5],[-2,4],[-1,3],[0,2],[1,3],
        [2,4],[3,5],[4,6],[5,7],[6,8],[8,8],[9,7],[8,6],[7,5],
        [6,4],[5,3],[4,2],[3,1],[2,0],[3,-1],[4,-2],[5,-3],[6,-4],[7,-5],[8,-6],
        [9,-7],[7,-7],[6,-6],[5,-5],[4,-4],[3,-3],[2,-2],[1,-1],
        [-7,-9],[-7,-7],[-6,-6],[-5,-5],[-4,-4],[-3,-3],[-2,-2],[-1,-1],
        [-9,7],[-7,7],[-6,6],[-5,5],[-4,4],[-3,3],[-2,2],[-1,1],
        [7,9],[7,7],[6,6],[5,5],[4,4],[3,3],[2,2],[1,1]];
const maja = [[9.3,-0.7],[7.9,-0.7],[7.9,0.7],[9.3,0.7],
        [0.7,-9.3],[-0.7,-9.3],[-0.7,-7.9],[0.7,-7.9],
        [-7.9,-0.7],[-9.3,-0.7],[-9.3,0.7],[-7.9,0.7],
        [0.7,7.9],[-0.7,7.9],[-0.7,9.3],[0.7,9.3]];
const t1="<i class='fa fa-laptop' style='font-size:20px;'></i>";
const t2="<i class='fa fa-user-o' style='font-size:18px;'></i>";
var poz = [[-1,-1,-1,-1],[-1,-1,-1,-1],[-1,-1,-1,-1],[-1,-1,-1,-1]];
var iespejams = [[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]];
var nosita = [[0,0],[0,0],[0,0],[0,0]];
var dators = [true,true,true,true];
var c = document.getElementById('kanva');
var centerX=c.width/2;
var centerY=c.height/2;
var x=0;
var y=0;
var npk=0;
var acis=0;
var cilveki=0;
var datori=4;
var knr=-1;
var klik=-1;
var palaists=false;
var ok=false;
var beigt=false;
var irgajiens=false;
var vieta=[];
for (var i=0; i<4; i++){
        vieta.push([]);
        vieta[i].push(new Array(68));
        for (var j=0; j<68; j++){
                if (j<60){
                        var n=15*i+j;
                        if (n>59) {n=n-60};
                        }
                else
                        {n=8*i+j}
                vieta[i][j]=n;
        }
}
function zime_laukumu()
{
        var k;
```

```
	for (var i = 0; i < krusts.length; i++)
		{
			if (i<60)
		  {k=4}
			else if (i<68)
		  {k=0}
			else if (i<76)
		  {k=1}
			else if (i<84)
		  {k=2}
			else
			{k=3};
			zime_apliti(i,k,spilgtums,"krustam")
		}
		for (var i=0; i<4; i++)
			for (var j=0; j<4; j++)
				{zime_apliti(i*4+j,i,1,"majai")}
	document.getElementById("datori").innerHTML = t1 + " 4 " + t2 + " 0";
	document.getElementById("seciba").innerHTML = "Izvēlies pretiniekus"
}
function zime_apliti(nr,kr,sp,kur)
{
   var ctx = c.getContext('2d');
	ctx.globalAlpha = 1;
   ctx.beginPath();
	ctx.fillStyle=krasa[5];
	if (kur=="krustam")
		{ctx.fillRect(centerX-18+krusts[nr][0] *30, centerY-18+krusts[nr][1] *30, 36, 36);
	 ctx.arc(centerX+krusts[nr][0] *30, centerY+krusts[nr][1] *30, 18, 0, Math.PI * 2, true)}
	if (kur=="majai")
		{ctx.fillRect(centerX-18+maja[nr][0] *30, centerY-18+maja[nr][1] *30, 36, 36);
		ctx.arc(centerX+maja[nr][0] *30, centerY+maja[nr][1] *30, 18, 0, Math.PI * 2, true)}
	ctx.globalAlpha = sp;
	ctx.fillStyle = krasa[kr];
	ctx.fill()
}
function gajiena_info()
{
	if (!beigt)
	{
		if (dators[npk] || !dators[npk] && ok || !dators[npk] && !irgajiens)
			{acis=Math.floor(Math.random()*6)}
		metamais();
		document.getElementById("seciba").innerHTML = "Gājiens "+krasaLV[npk];
		var info=document.getElementById("seciba");
		info.style.color = krasa[npk];
		iespejami_gajieni();
		if (dators[npk] || !dators[npk] && !irgajiens) {klikskis()}
	}
}
function iespejami_gajieni()
{
	for (var i=0; i<4; i++)
		for (var j=0; j<4; j++)
			{iespejams[i][j]=0};
	var starts=Boolean(poz[npk][0]!==0 && poz[npk][1]!==0 && poz[npk][2]!==0 && poz[npk][3]!==0);
	starts=Boolean(starts && poz[npk][0]!==61 && poz[npk][1]!==61 && poz[npk][2]!==61 && poz[npk][3]!==61);
	var savs=[0,0,0,0];
	var gals=68+8*npk;
	for (var i=0; i<4; i++)
		for (var j=0; j<4; j++)
```

```
		{if (i!==j && poz[npk][i]>-1 && poz[npk][j]>-1)
			{
			if (poz[npk][i]+acis+1==poz[npk][j] || poz[npk][i]+acis==60 && poz[npk][j]==0)
				{savs[i]=1}
			}
		}
	for (var i=0; i<4; i++)
		{
			var p=poz[npk][i];
			if (acis==0 || acis==5)
			{
				if (starts && p<0) {iespejams[npk][i]=1}
				else if (vieta[npk][p]+acis<gals && savs[i]==0) {iespejams[npk][i]=2}
			}
			else if (vieta[npk][p]+acis<gals && savs[i]==0) {iespejams[npk][i]=2}
		}
	irgajiens=Boolean(iespejams[npk][0]>0 || iespejams[npk][1]>0 || iespejams[npk][2]>0 || iespejams[npk][3]>0)
}
function gajiena_izvele()
{
	var v;
	var minus;
	var pirmais=-1;
	var maxpoz=-1;
	var maja=-1;
	var kauj=-1;
	if (irgajiens)
	{
		for (var i=0; i<4; i++)
		{
			if (iespejams[npk][i]==2 && poz[npk][i]>maxpoz)
			{
				maxpoz=poz[npk][i];
				pirmais=i
			}
		}
		for (var i=0; i<4; i++)
		{
			if (iespejams[npk][i]==2)
			{
				v=vieta[npk][poz[npk][i]]+acis+1;
				for (var j=0; j<4; j++)
					for (var k=0; k<4; k++)
						if (j!==npk)
						{
							if (poz[j][k]==60-15*j) {minus=0} else {minus=1}
							if (v-1==vieta[j][poz[j][k]-minus]) {kauj=i}
						}
			}
		}
		for (var i=0; i<4; i++)
		{
			if(iespejams[npk][i]==1 && maja<0) {maja=i}
		}
		if (kauj>-1)
			{klik=kauj}
			else if (maja<0 && pirmais>0) {klik=pirmais}
				else if (maja>-1 && pirmais<0) {klik=maja}
					else if (acis==0 && maja>-1) {klik=maja}
						else {klik=pirmais}
	}
```

```
	else {klik=-1}
}
function gajiens()
{
	if (beigt) {return}
	setTimeout(gajiena_info,pauze);
	if (dators[npk]) {gajiena_izvele()};
	if (iespejams[npk][0]>0 && klik==0) {knr=0}
		else if (iespejams[npk][1]>0 && klik==1) {knr=1}
			else if (iespejams[npk][2]>0 && klik==2) {knr=2}
				else if (iespejams[npk][3]>0 && klik==3) {knr=3}
					else {knr=-1}
	if (knr>-1)
	{
		if (iespejams[npk][knr]==1)
			{zime_apliti(4*npk+knr,5,1,"majai");
			var p=vieta[npk][60];
			zime_apliti(p,npk,1,"krustam");
			poz[npk][knr]=0;
			ok=true}
		else
			{ok=true;
			for (var i=0; i<acis+1; i++)
				{paiet_par_vienu()};
				var rinda;
				var attalums=0;
				for (var a=0; a<4; a++)
					if (poz[npk][a]>0) {attalums=attalums+poz[npk][a]}
				rinda = document.getElementById("stat").rows[npk+1].cells;
				rinda[0].innerHTML = attalums;
				if (attalums==266)
					{
						stop();
						document.getElementById("seciba").innerHTML = "Uzvara
"+krasaLV[npk]
					}
				if (vieta[npk][poz[npk][knr]]<61) {nosit_citu()}
			}
		klik=-1
	}
	else
		{ok=false}
	if (acis<5 && dators[npk] || acis<5 && !dators[npk] && irgajiens && ok || !dators[npk] && !irgajiens)
	{
			if (npk<3) {npk++} else {npk=0}
		}
}
function paiet_par_vienu()
{
	var minus;
	var fons=4;
	var sp=spilgtums;
	var p=poz[npk][knr];
	for (i=0; i<4; i++)
	{
		for (j=0; j<4; j++)
		{
			if ([poz[i][j]]>-1)
			{
				if (poz[i][j]==60-15*i) {minus=0} else {minus=1}
				if (vieta[npk][p-1]==vieta[i][poz[i][j]-minus] && i!==npk)
```

```
                                        {
                                                sp=1;
                                                fons=i
                                        }
                                        else if ((vieta[npk][p-1]==vieta[i][poz[i][j]-minus] || p==61 && poz[i][j]==0) && j!==knr)
                                        {
                                                sp=1;
                                                fons=npk
                                        }
                                }
                        }
                }
                if (p==0)
                        {zime_apliti(vieta[npk][60],npk,spilgtums,"krustam")}
                else if (p>60)
                        {zime_apliti(vieta[npk][p-1],npk,sp,"krustam")}
                else
                        {zime_apliti(vieta[npk][p-1],fons,sp,"krustam")}
                zime_apliti(vieta[npk][p],npk,1,"krustam");
                poz[npk][knr]++
}
function nosit_citu()
{
        var rinda;
        var minus;
        var v=vieta[npk][poz[npk][knr]];
        for (var i=0; i<4; i++)
                {if (i!==npk)
                        {for (var j=0; j<4; j++)
                                {if (poz[i][j]>0)
                                        {
                                        if (poz[i][j]==60-15*i) {minus=0} else {minus=1}
                                        if (v-1==vieta[i][poz[i][j]-minus])
                                                {
                                                        zime_apliti(i*4+j,i,1,"majai");
                                                        poz[i][j]=-1;
                                                        nosita[npk][0]++;
                                                        nosita[i][1]++
                                                        var att=0;
                                                        for (var a=0; a<4; a++)
                                                                if (poz[i][a]>0) {att=att + poz[i][a]};
                                                        rinda = document.getElementById("stat").rows[npk+1].cells;
                                                        rinda[1].innerHTML = nosita[npk][0];
                                                        rinda = document.getElementById("stat").rows[i+1].cells;
                                                        rinda[0].innerHTML = att;
                                                        rinda[2].innerHTML = nosita[i][1]
                                                }
                                        }
                                }
                        }
                }
}
function klikskis()
{
        if (!beigt && palaists) {document.getElementById("met").click()}
}
function stop()
{
        beigt=true
        document.getElementById("sac").disabled = true
        document.getElementById("met").disabled = true
```

```
		document.getElementById("beidz").disabled = true
		if (poz[npk][0]>64 && poz[npk][1]>64 && poz[npk][2]>64 && poz[npk][3]>64)
			{
				var uzvara = document.getElementById("stat").rows[npk+1];
				uzvara.style.color = krasa[npk]
			}
		else {document.getElementById("seciba").innerHTML = "Spēle netika pabeigta!"}
}
function koordinatas()
{
 knr=-1;
 var x1 = event.clientX;
 var y1 = event.clientY;
 x = Math.round((x1-600)/30);
 y = Math.round((y1-302)/30);
 for (var i = 0; i < krusts.length; i++)
 {
			if (x==krusts[i][0] && y==krusts[i][1])
			{
				for (var j=0; j<4; j++)
				{
					if (poz[npk][j]==0 && i==60+8*npk || i==vieta[npk][poz[npk][j]-1])
					{
						klik=j;
						klikskis();
						return
					}
				}
			}
 }
 if (acis==0 || acis==5)
 {
			for (var i = 0; i < maja.length; i++)
			{
				if (x==Math.round(maja[i][0]) && y==Math.round(maja[i][1]))
				{
					for (var j=0; j<4; j++)
					{
						if (poz[npk][j]<0 && j==i-4*npk)
						{
							klik=j;
							klikskis();
							return
						}
					}
				}
			}
 }
}
function iedarbina()
{
		var rinda;
		var galva = [];
		galva[0]="<i class='fa fa-arrows-alt'></i>";
		galva[1]="<i class='fa fa-smile-o'></i>";
		galva[2]="<i class='fa fa-frown-o'></i>";
		document.getElementById("stat").style.border = "1px solid gray";
		for (var i=0; i<3; i++)
		{
			rinda = document.getElementById("stat").rows[0].cells;
			rinda[i].innerHTML = galva[i]
```

```
		}
		palaists=true;
		document.getElementById("sac").disabled = true
		document.getElementById("met").disabled = false
		for (var i=0; i<4; i++)
		{
			document.getElementById("ch" + i).disabled = true;
			rinda = document.getElementById("stat").rows[i+1].cells;
			rinda[0].innerHTML = 0;
			rinda[1].innerHTML = 0;
			rinda[2].innerHTML = 0
		}
		gajiena_info()
}
function speletajs(n)
{
 dators[n]=document.getElementById("ch" + n).checked;
 if (dators[n])
			{datori++;
			cilveki--}
		else
			{datori--;
			cilveki++}
 document.getElementById("datori").innerHTML = t1 + " " + datori + " " + t2 + " " + cilveki
}
function metamais()
{
		var aplis=[];
		var s=0;
		svgDoc.appendChild(taisnsturis);
		for (var i=0; i<7; i++)
		{if (cipars[acis][i]==1)
			{aplis[s] = document.createElementNS(svgns, "circle");
			aplis[s].setAttributeNS(null, "cx", acs_vieta[i][0]*mg/7+x0+mg/2);
			aplis[s].setAttributeNS(null, "cy", acs_vieta[i][1]*mg/7+y0+mg/2);
			aplis[s].setAttributeNS(null, "r", acs_radiuss);
			aplis[s].setAttributeNS(null, "fill", krasa[npk]);
			svgDoc.appendChild(aplis[s]);
			s++}
		}
}
</script>
<br>
<table id="stat">
		<tr><th></th><th></th><th></th></tr>
		<tr><td></td><td></td><td></td></tr>
		<tr><td></td><td></td><td></td></tr>
		<tr><td></td><td></td><td></td></tr>
		<tr><td></td><td></td><td></td></tr>
</table>
<div id="k0"></div>
<div id="k1"></div>
<div id="k2"></div>
<div id="k3"></div>
<br>
<div id="p0"></div>
<div id="p1"></div>
<div id="p2"></div>
<div id="p3"></div>
</body>
</html>
```