

LATVIJAS UNIVERSITĀTE DATORIKAS FAKULTĀTE

**KRĀSU IETEKME MEKLĒŠANAS UZDEVUMU
IZPILDĒ VIRTUĀLĀ VIDĒ**

BAKALaura DARBS

Autors: **Georgijs Pliska**

Studenta apliecības Nr.: gp17023

Darba vadītājā: Asoc. prof., MSc. Math., Dr. Geol. Līga Zariņa

RĪGA 2021

ANOTĀCIJA

Pēdējo 10 gadu laikā, virtuālas realitātes tehnoloģijas kļuvas pietīkami lētas un viegli ražojamas, lai būtu pieejami lielai publikai. Tas piesaistījis gan privāto uzņēmumu uzmanību, gan pētnieku uzmanību. Pētījumu skaits, kura tiek izmantota virtuālā realitāte ir palielinājies un turpina augt. Pētījumi raisa jautājumu, vai tiešām virtuālā realitāte stimulē mūsu smadzenes, tādā pašā veidā kā mūsu realitāte?

Bakalaura darba ietvaros tiek pētīts, ka krāsas, pieņemot ka piesātinājums un apgaismojums ir vienāds, ietekmē meklēšanas uzdevumu izpildi virtuālajā vidē. Darba ietvaros tikusi izveidota lietotne eksperimenta veikšanai un novadīts eksperiments klātienē.

Darba nobeigumā tika secināts, ka zaļās un zilās krāsas fona ietekmē reakcijas laiks uzdevuma izpildē ir visilgākais. Otrās ātrākais reakcijas laiks ir violetai fona krāsai un ātrākais laiks ir dzeltenai fona krāsai. Tika noskaidrots, ka stimulus iekļautais papildobjektu skaits uzdevumā meklējamam objektam nozīmīgi ietekmē reakcijas laiku katras fona krāsas ietvaros. Rezultāti liecina par tendenci, ka sarkanai krāsai tuvākās fona krāsas atbilstoši krāsu aplim atbilst mazākam reakcijas laikam, bet fona krāsu, kas krāsu aplī atrodas attālāk no sarkanā, ietekmē reakcijas laiks ir ilgāks.

Atslēgvārdi: krāsas, virtuālā vide, meklēšanās uzdevums, reakcijas laiks, virtuālā realitātē

ABSTRACT

Over the past 10 years, virtual reality technologies have become cheap and easy enough to make available to the public. This has attracted the attention of both private companies and researchers. The number of studies that use virtual reality has increased and continues to grow. Research raises the question of whether virtual reality really stimulates our brains in the same way as our reality?

Within the framework of the bachelor's thesis, it is studied those colours, assuming that the saturation and lighting are equal, affect the performance of search tasks in the virtual environment. Within the framework of the work, an application for the experiment was created and the experiment was conducted in person.

At the end of the work, it was concluded that under the influence of green and blue background the reaction time in the task is the longest. The second fastest response time is for the purple background colour and the fastest time is for the yellow background colour. It was found that the number of additional objects included in the stimuli for the object sought in the task significantly affects the reaction time within each background colour. The results show a trend that the background colours closest to red correspond to a shorter response time according to the colour wheel, while the background colour located farther from the red in the colour wheel has a longer response time.

The effect of color on the performance of search tasks in a virtual environment

Keywords: colours, virtual environment, search task, reaction time, virtual reality

SATURA RĀDĪTĀJS

ANOTACĪJA	2
ABSTRACT.....	3
SATURA RADĪTĀJS	4
IEVADS	5
1. VIRTUĀLAS VIDES DEFINICIJA.....	7
2. EKSPERIMENTĀLĀS DAĻAS PLĀNOŠANA	8
2.1 Nepieciešamā aparatūras iegāde	8
2.2 Virtuālas testu telpas plānošana	8
2.3 Meklēšanas uzdevumu plānošana	9
2.4 Krāsu izvēlē priekš eksperimentālās daļas	10
3. EKSPERIMENTĀLĀS DAĻAS IZSTRĀDE	12
3.1 Testu telpas būvēšanā.....	12
3.2 Kodēšana	13
3.2.1 Sienas krāsas maiņa programmēšanā	13
3.2.2. Meklēšanas uzdevumu programmēšanā.....	14
4. EKSPERIMENTĀLĀ DAĻA	17
5. REZULTĀTI.....	19
SECINĀJUMI.....	24
IZMANTOTĀ LITERATŪRA.....	25
PIELIKUMI	26

IEVADS

1965.gadā, Ivans Sutherlands, savā darbā “Galvenais Ekrāns” uzrakstīja - “Galvenais ekrāns, protams, būtu telpa, kurā dators var kontrolēt matērijas esamību. Šādā telpā parādīts krēsls būtu pietiekami labs, lai tajā varētu sēdēt. Šādā telpā parādīti roku dzelži būtu pietiekami ierobežojoši, un šādā telpā parādīta lode būtu liktenīga. Ar atbilstošu programmēšanu šāds ekrāns, burtiski, varētu būt Brīnumzemē, kurā Alise gāja.”[1]. Ar tādu ļoti mākslīgu un piesātinātu aprakstu, Ivans aprakstīja to ko mēs tagad zinām kā virtuālās realitātes tehnoloģijas.

Tiek uzskatīts kā pirmais tehniskais koncepts, parādījās 3 gadus ātrāk par augstāk minēto izteicienu - 1962. gadā, kad Mortons Heiligs izveidoja “Sensorama” - simulētu motocikla pieredzi, kas brauc cauri Bruklinai, kam raksturīgi maņu iespaidi, piemēram, audio, ožas un haptiskie stimuli, ieskaitot arī vēju, lai sniegtu reālistisku pieredzi. [5]

1980s gados, parādījās pirmie prototipi, kuri kļuva komerciāli. 1985. gadā kompānija “VPL Research” izveidoja “DataGlove”, cimdu ar sensoriem kura var izmērīt pirka pozīciju, orientāciju un saprast roku zīmes. 1988. gadā, “VPL Research” izveidoja “Eyephone”, ķiveri ar ekrānu, kas ļoti stipri atgādina virtuālas realitātes ķiveri, kādus mēs pieradām redzēt mūsdienas.

Pēdējo 10 gadu laika, tehnoloģija kļuva pietiekami lēta, ar lielāku redzes lauku un mazāku aizturi. Sāka parādīties ķiveres, orientēti uz masveidīgu pārdošanu – “HTC Vive”, “Oculus Rift”, “Valve Index” un citi. Ķiveru pieplūde tirgū piesaistīja sev lielu uzmanību gan no parastiem lietotājiem, gan no uzņēmumiem. Industrijas, kuras izmanto virtuālo realitāti vai saka mēģināt to integrēt, vienmēr papildinās - videospēlēs, armijā, arhitektūrā, izglītībā, medicīnā, sociālā aprūpē, tūrismā un citur. Arī stipri palielinājās pētījumu skaits, kas izmanto virtuālu realitāti priekš stimulu izveidošanas. Lielākais pētījumu skaits, kur ir izmantots VR ir medicīnā un izglītības.

Bet vai eksistē pārliecinošas zināšanas, ka stimulu ietekme iekš virtuālās vides sakrīt ar stimulu ietekme reālajā vide? Pētījumi, kuri pārbauda šo jautājumu nav veikti lielā apjomā un autors, ar savu bakalaura darbu, vēlas papildināt šīs zināšanas, veicot eksperimentu, kurš pētītu kādu vienu stimulu un tā iedarbību iekš virtuālās vides.

Šī pētījuma mērķis ir izpētīt krāsas ietekmē iekš virtuālās vides. Tiek samazināts mainīgo skaits, lai būtu iespēja pārbaudīt tieši vienu krāsas aspektus, jo visus aspektu kombinācijas bakalaura darbā nav iespējams pārbaudīt. Pētījumā tiek izmantota meklēšanas uzdevumu izpilde kā nemainīgais faktors. Darba ietvaros tiks izpētīta iepriekš pieejamā literatūra un veikts eksperiments.

Lai sasniegtu pētījuma mērķi, nepieciešams veikt sekojošas darbības:

1. Izpētīt pieejamo literatūru;
2. Izveidot lietotni priekš eksperimenta veidošanas;
3. Novadīt eksperimentu, lai iegūtu datus;
4. Apstrādāt iegūtos datus;
5. Veikt secinājumus un ieteikumus.

Darbs ir sadalīts divās daļās – teorētiskā un praktiskā. Teorētiskajā daļā tiks pētīta pieejamā literatūra. Praktiskajā daļā tika veikts eksperiments klātienē, kurā tika noteikts, ka krāsas ietekmē meklēšanas uzdevumus. Pētījumu noslēdz eksperimenta rezultāti, diskusija un izdarītie secinājumi.

1. VIRTUĀLAS VIDES DEFINICIJA

Saksim ar jautājumu – kas ir virtuāla vide? Trīs galvenās virtuālās vides īpašības – klātbūtne, mijiedarbība un iegremdēšana.

Mijiedarbības var būt aprakstītas skalā, kas izmēra, cik daudz no virtuālās vides lietotājs var mainīt. [6] Klātbūtne tiek uzskatīta par “subjektīvu pieredzi atrasties vienā vietā vai vide, pat ja cilvēks fiziski atrodas citā”. [7]. Klātbūtne ir sarežģīts psiholoģisks fenomens, kad lietotājam ir sajūta, ka “viņš ir tur” iekš virtuālas vides. Tas ir saistīts ar fiziskās telpas sajūtām un uztveri un arī iespējai iedarboties ar pasauli, tapāt kā ar īsto. [3]

Lai gan pētnieki lielā mērā piekrīt mijiedarbības un klātbūtnes definīcijām, pastāv atšķirīgi viedokļi par iegremdēšanas jēdzienu. Viena pētnieku nozare ierosina, ka iegremdēšana jāuztver kā tehnoloģisks atribūts, ko var objektīvi novērtēt, savukārt citi iegremdēšanu raksturo kā subjektīvu, individuālu pārlicību, t.i. psiholoģisku parādību [7]

Iegremdēšanās ir saistīta ar lietotāja sagaidāmo līmeni kas tam ir par pieredzi. Ja esošie stimuli ir līdzīgi realitātei, tad virtuālās realitātes lietotāja pieredze būs līdzīga ar reālo pieredzi, padziļinot virtuālās realitātes pieredzi. Tas arī atteicas uz to vai iedarbība ir līdzīga reālai pasaulei. Ja tā ir, tad reālisma sajūta palielinās. [4]

Iegremdēšanās attiecas uz stimulēto maņu daudzumu, mijiedarbību un vides simulēšanai izmantoto stimulu realitāti. Šī funkcija var būt atkarīga no tehnoloģiskās sistēmas īpašībām, ko izmanto, lai izolētu lietotāju no realitātes [2]. Augstākās vai zemākās iegremdēšanās pakāpes var būt atkarīgas no divu veidu virtuālas realitātes sistēmām, kuras tiek nodrošinātas lietotājam:

- Iegremdējošās sistēmas ir vienkāršākas un lētākas virtuālas realitātes lietojumprogrammu veids, kas pasaules attēlu reproducēšanai izmanto darbvirsmas.
- Iegremdējošās sistēmas nodrošina pilnīgu simulētu pieredzi, izmantojot vairākas maņu izejas ierīces, piemēram, uz galvas uzstādīti displeji, uzlabotu vides stereoskopisko skatu, kā arī audio un haptiskās ierīces.

2. EKSPERIMENTĀLĀS DAĻAS PLĀNOŠANA

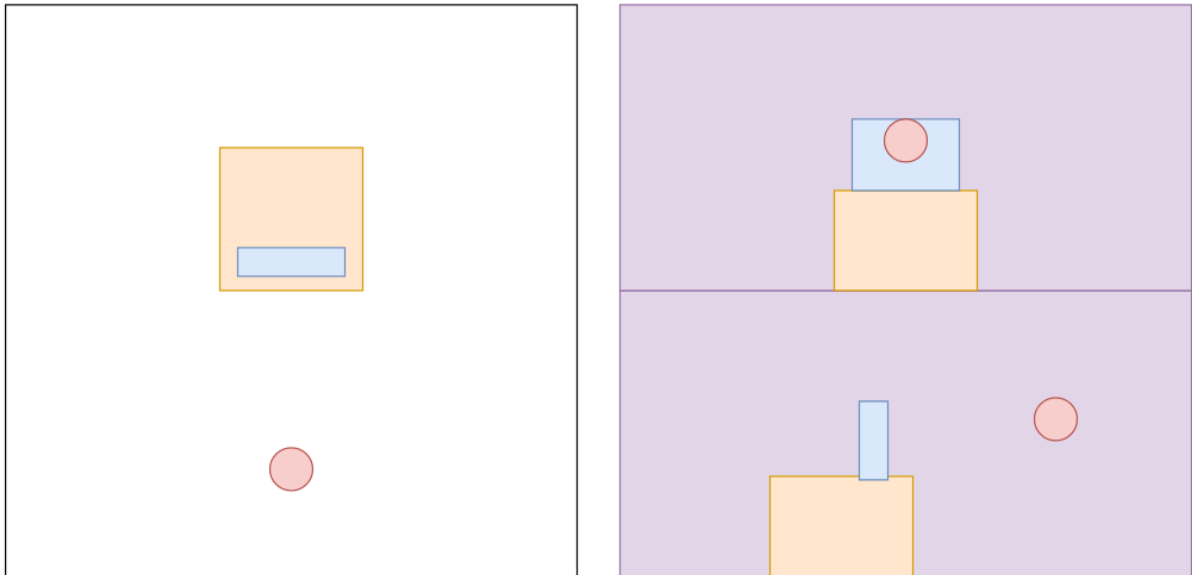
2.1 Nepieciešamā aparatūras iegāde

Saksim ar nepieciešamās aparatūras iegādes t.i. jaunas virtuālas realitātes ķivere un pietīkami jaudīgu datoru, lai izstrādāt un novadīt eksperimentu. Lai nepirktu jaunu aparatūru, speciāli šim eksperimentam, tiek atrasta iespējā izmantot Latvijas Universitātes Datorikas Fakultātes pieejamu virtuālas realitātes ķiveri un datoru, kas ir pieejamas “DF LAB” laboratorija. “DF LAB” ir studentu laboratorijā kas ir atvērta 2017. gadā. Laboratorija dot iespēju katram studentam uzbūvēt tehniskas iekārtas. Ir pieejams lāzera griezējs, 3D printeri un virtuālas realitātes ķivere. Darba autoram tiek dota atļaujā no laboratorijas pārstāvjiem izmantot laboratorijā pieejamu aparatūru priekš eksperimentālās daļās. Tiek izmantota virtuālas realitātes ķivere “HTC Vive” - 110 grādu redzeslauks, izšķirtspējā uz katru aci ir 1080x1200 pikseli un atsvaizdināšanas biežums ir 90Hz[8]. Laboratorijā pieejamam datoram ir grafiskā karte “NVIDIA GeForce GTX 1080 TI” un procesors “Intel Core i7-7700K”. Dators ir pietīkami spēcīgs lai veidot šo eksperimentu.

2.2 Virtuālas testu telpas plānošana

Metodoloģija priekš virtuālas testu telpas izveides netika atrasta, tāpēc telpas dizaina izveide notika balstoties uz autora pieredzi. Testu telpas projektējumu var redzēt attēlā 2.1. Pēc konsultācijas ar darba vadītāju, tiek sastādīti šādas telpas īpašības:

- Testu telpai vajag būt aizvērtai bez logiem un durvīm, lai samazinātu stimulu skaitu
- Testu telpai vajag būt vienmērīgi apgaismotai
- Pētāmai personai nevajag būt iespējai pārvietoties - personai vajag sēdēt uz vietas
- Meklēšanas uzdevumam vajag būt pilnīgi vizuālam, un tiek radītam uz kaut kādas virsmas
- Telpas sienu krāsas tiek mainītas



2.1. att. Testu telpas projektējums. Ar sarkanu punktu ir parādītā gaismas avota pozīcijā un pētāmas personas galvas pozīcijā. Ar baltu ir parādītā grīda. Ar violetu ir parādītas sienas. Ar oranžu ir parādītā priekšmets uz kuras būs objekts ar meklēšanas uzdevumu. Ar zilu ir parādīta virsma priekš meklēšanā uzdevuma.

2.3 Meklēšanas uzdevumu plānošana

Vizuālā meklēšana (vai vienkārši – meklēšana) ir definētu stimulu meklēšana pārblīvētā vizuālajā vide. [11] Vienumu, kurš tiek meklēts, sauc par mērķi, bet vienumus, kas nav mērķis, sauc par traucētājiem.

Tiek pieņemts lēmums izmantot raksturiezīmes meklēšanu. Raksturiezīmes meklēšanā (saukta arī par "disjunktīvu" vai "efektīvu" meklēšanu) ir meklēšana, kur koncentrēšana notiek uz mērķa identificēšanu starp traucētājiem, kas atšķiras no mērķa ar vienu vizuālo elementu - krāsu, formu, orientāciju, vai izmēru. Piemēram, pētāmai personai ir nepieciešams atrast zaļo kvadrātu, kuram apkārt ir zili kvadrāti. Raksturiezīmes meklēšanā efektivitāte tiek mērīta ar reakcijas laiku un precizitāti ir atkarīga no "izlēcienu" efekta. [11]

Raksturiezīmes meklēšana tiek izvēlēta ar mērķi samazināt iespējamo mainīgu skaitu. Raksturiezīmes meklēšanā ir minimālais iespējamais mainīgu skaits t.i. viens mainīgais, bet darba autors uzraucas ka "izlēcienu" efekts var padarīt uzdevumu pārāk vieglu, ja būs tikai divas krāsas – viena priekš mērķa un viena priekš traucētājiem. Lai izvairīties no pārāk vieglā uzdevuma, darba autors pievienoja vēl četras krāsas pie traucētāju krasam, bet tas krāsas aizņems ne vairāk ka pusi no traucētājiem. Pievienotas traucētāju krāsas būs pietuvināti mērķa krasai, kamēr traucētāju krāsa būs pēc iespējas pretējā no mērķa krāsas, lai atvieglot meklēšanu.

Meklēšanas uzdevums tiek radīts uz virtuālā virsmas iekš virtuālas testu telpas.

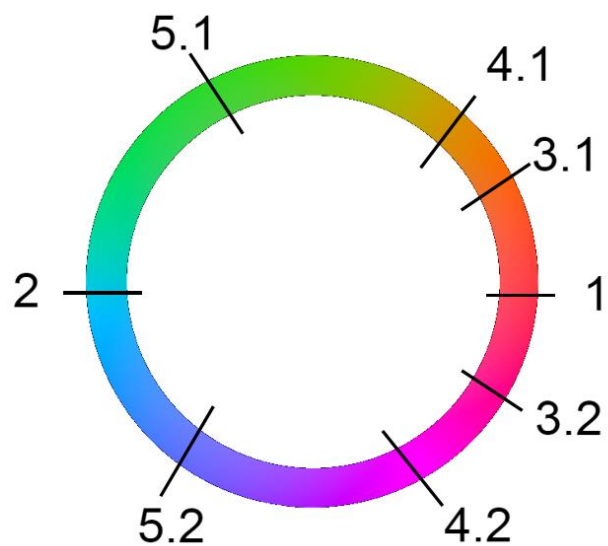
Pieņemti sekojoši meklēšanas uzdevuma raksturīpašības, kā uzdevums:

- Sastāvēs no 40 mēģinājumiem
- 20 mēģinājumiem tiek parādīts mērķis
- 20 mēģinājumiem netiek parādīts mērķis
- 10 mēģinājumiem būs 5 pievienoti traucētāji ar citu krāsu, un pārējie ar galvenā krāsu, puse no mēģinājumiem ar mērķi un puse bez mērķa.
- 10 no mēģinājumiem būs 10 pievienoti traucētāji ar citu krāsu, un pārējie ar galvenā krāsu, puse no mēģinājumiem ar mērķi un puse bez mērķa.
- 10 no mēģinājumiem būs 15 pievienoti traucētāji ar citu krāsu, un pārējie ar galvenā krāsu, puse no mēģinājumiem ar mērķi un puse bez mērķa.
- 10 no mēģinājumiem būs 20 pievienoti traucētāji ar citu krāsu, un pārējie ar galvenā krāsu, puse no mēģinājumiem ar mērķi un puse bez mērķa.

2.4 Krāsu izvēlē priekš eksperimentālās daļas

Pēc konsultācijas ar darba vadītāju, meklēšanas uzdevuma mērķa krāsa tiek izvēlēta sarkana krāsa. Sarkana krāsa rāda ātrāku reakcijas un atpazīšanas laiku [12] kas atvieglos mērķa meklēšanu. Izvēlēto krāsu kopsavilkumu var redzēt attēlā 2.2[13]:

- Traucētāju krāsa ir paņemta ciāna krāsa, jo ciāna krāsa ir krāsu apli pretējā sarkanu krāsai.
- Pievienotas traucētāju krāsas ir 30 grādi (oranža) un 60 grādi (dzeltena) uz vienu pusi un -30 grādi (rozā) un -60 grādi (violela) pretējā puse no sarkanas krāsas, ar mērķi izvēlēties pietuvinātas sarkanām krāsas lai samazinātu “izlēciena” efektu.
- Sienas krāsa ir 60 grādi (dzeltenā) un 120 grādi (zaļa) uz vienu pusi un -60 grādi (violela) un -120 grādi (zila) pretējā puse no sarkanas krāsas, lai aizņemtu pēc iespējas plašas riņķa vietas.
- Meklēšanas uzdevumā radīšanas virsma tiek izvēlēta melna krāsa kontrasta izveidošanai ar mērķi un traucētājiem.



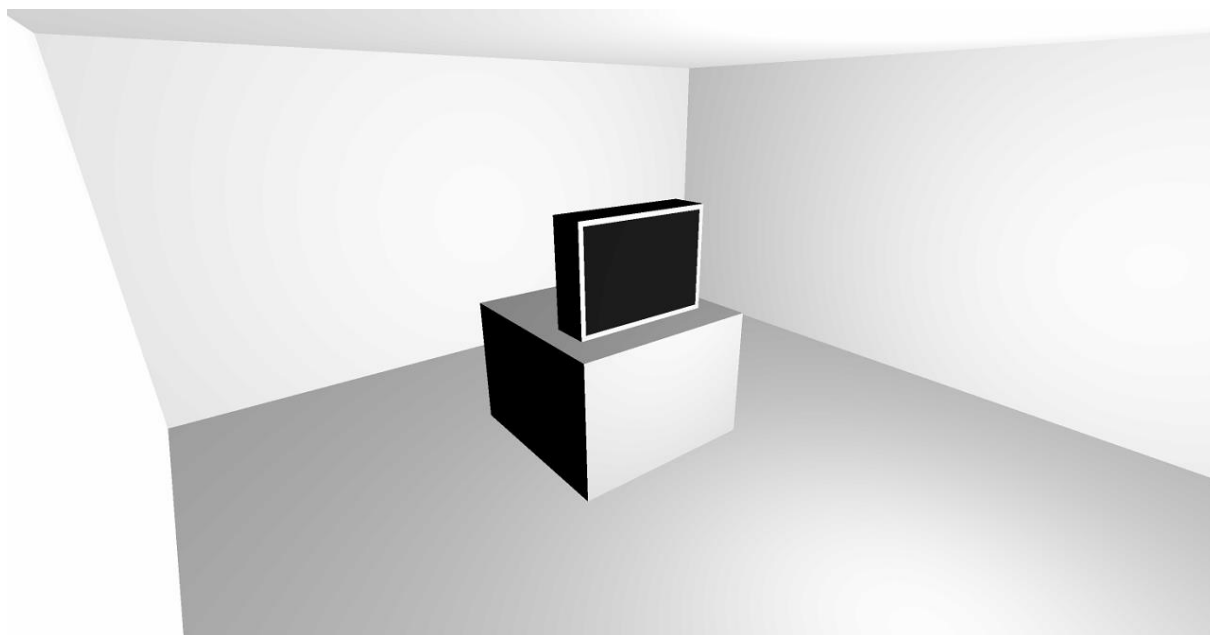
2.2. att. Krāsu riņķus ar atzīmētām krāsām – Mērķus krāsa (1. sarkana), Pamata traucētāju krāsa (2. ciāna), Pievienotas traucētāju krāsas (3.1. oranža, 3.2. rozā, 4.1. dzeltena, 4.2. violeta) un Sienas fonu krāsas (4.1. dzeltena, 4.2. violeta, 5.1. zaļa, 5.2. zila)

3. EKSPERIMENTĀLĀS DAĻAS IZSTRĀDE

Priekš eksperimentālās daļas izpildes, ir nepieciešams izveidot specializētu lietotni, dizains priekš kuras jau ir apspriests iepriekšējā nodaļa. Priekš lietotnes izstrādes ir izmantots spēļu dzinis “Unity”, versija 2019.4.21.f. “Unity” ir bezmaksas spēļu dzinis, kurš ir populārākais dzinis tirgū[9]. “Unity” ir pieejamā liela datubāzē ar gataviem programmēšana risinājumiem, kas atvieglos darbu. Darba autoram ir pieredze ar izstrādi izmantojot “Unity”.

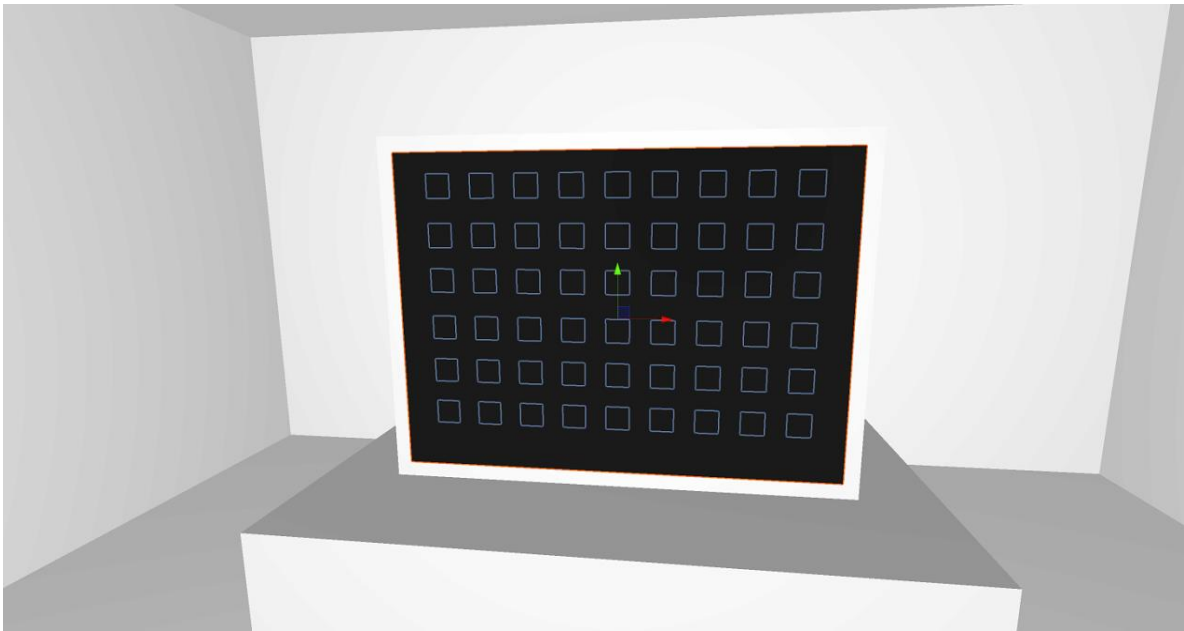
3.1 Testu telpas būvēšanā

Testu telpa izmērs ir 4x4x2m. Katra siena, grīdā un griesti ir atsevišķs objekts. Telpas centra, ar pārbīdi 0.5 metri pretējā virziena no pētāmas personas, stāv galds ar izmēriem 1x1x0.7m. Uz galda stāv monitors. Monitora izmēri ir 0.55x0.75x0.2m. Iekš monitora ir melns ekrāns ar izmēru 0.7x0.5m. Attālumā 1,5 metri no monitora ekrānā, augstuma 1m no grīdās ir ielikts gaismas avots. Gaismas avots ir punktu tipa gaisma, kas nozīmē ka visa gaisma iziet no viena punkta. Gaismas krāsa ir balta (HSL - 0, 100%,100%). Galīgo telpas izskatu var redzēt attēlā 3.1.



3.1. att. Testu telpas iekš “Unity”

Iekš monitora izdevās ielikt 54 kvadrāti, kas ir izvietoti 6x9 režģi, katrs izmērā 0.05x0.07m. Autors uzskata ka tas ir optimālais skaits priekš iespējamu mērķu un traucētāju skaitu. Kvadrāti izkārtojumu var redzēt attēlā 3.2.



3.2. att. Kvadrātu izvietojums iekš monitora

Priekš grīdās, galda un monitora ir izveidoti materiāli bez tekstūras, kuri vienmēr ir balti un nemainās. Priekš sienam un griežtiem ir izveidots otrais materiāls, kurš ir balts pēc noklusējuma un kuram var izmantoti krāsu. Priekš monitora un kvadrātiem iekš tajā ir izveidots trešais materiāls, kurš ir bez tekstūras, ir vienmēr melns, un kuram var mainīt krāsu.

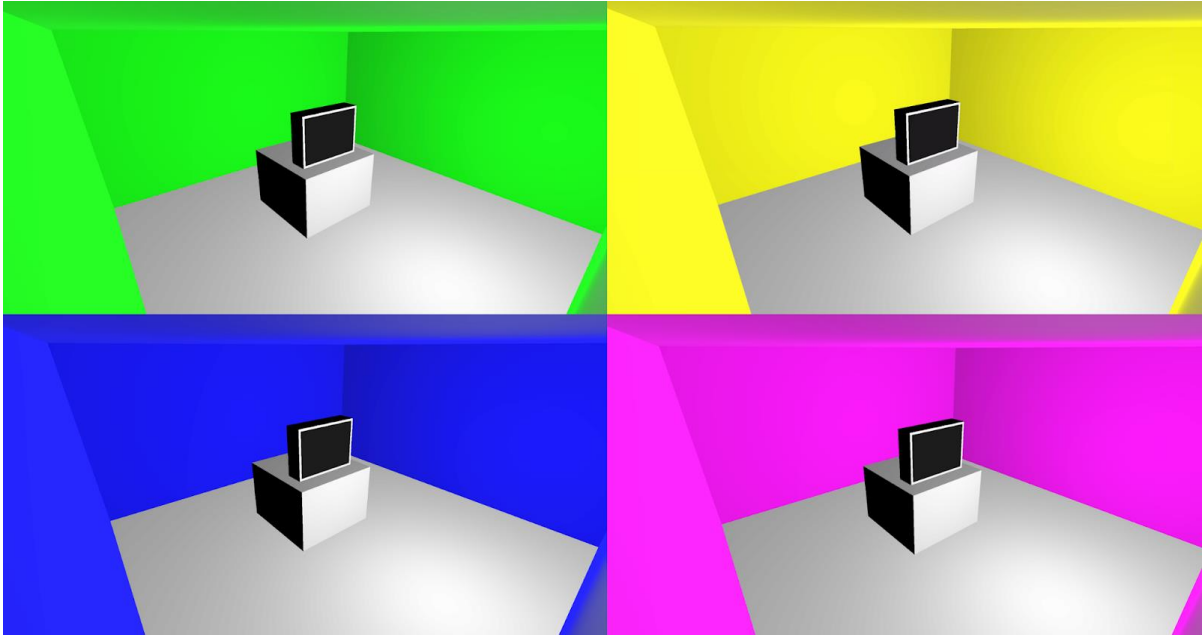
Lai izveidotu iespēju izmantot “HTC Vive” ķiveri iekš projekta, tiek izmantots gatavs programmēšanas risinājums kas ir brīvs un pieejams internetā[10]. To izveidoja pats ķiveres izstrādātājs.

3.2 Kodēšana

Darba vajadzībām nepieciešams izveidot divas funkcijas – viena maina testu telpas sienas krāsa un otra kontrole meklēšanas uzdevuma izpildi.

3.2.1 Sienas krāsas maiņa programmēšanā

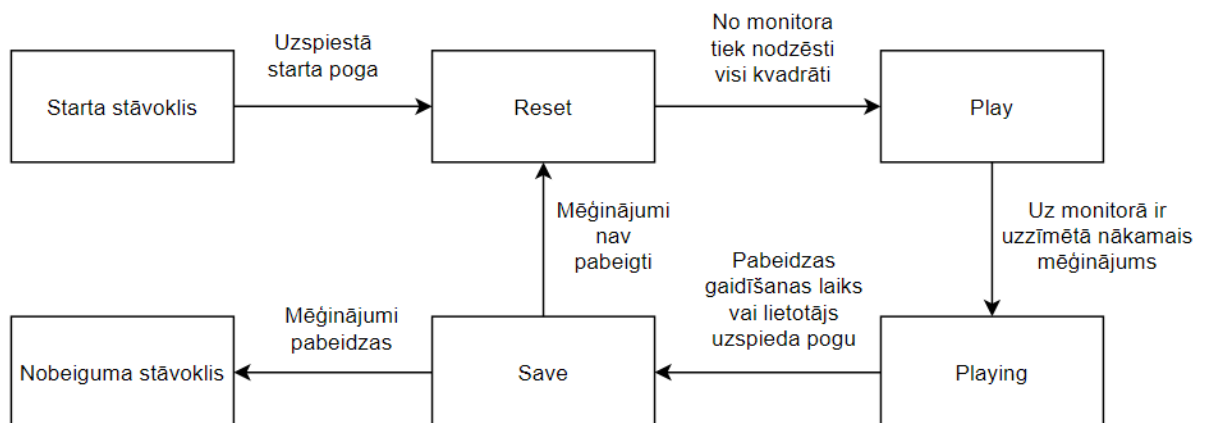
Tiek izveidots tukšais objekts iekš projektā, kur tiek apvienotas visas sienas un griežti. Tukšam objektam pievienots c# skripts. Skripts tiek izmantots lai mainīt sienu krāsu, pēc ieplānotas pogas nospiešanas. Izmantosim ciparu pogas no 0 līdz 3. Skripts nav grūts - tiek pārbaudīts vai ir nospiesta vajadzīgā poga, un ja ir, tad tiek mainīta krāsa. Krāsu maiņu var redzēt attēlā 3.3.



3.3. att. Četras krāsas, ar ko var nokrāsot sienas iekš testu telpas

3.2.2. Meklēšanas uzdevumu programmēšanā

Meklēšanas uzdevuma koda darbība balstās uz “Unity” divu galveno funkciju darbību - “update” funkcijas kura tiek izsaukta katru mīli sekundi un “start” funkcijas, kura tiek izsaukta vienu reizi kad sakas kodu izpildīšana. Uzdevums tiek programmēts ar stāvokļu sistēmu, kuru diagrammu var redzēt attēlā 3.4. Kopuma ir 4 stāvoklī kuros var atrastiem lietotne kad izpilda uzdevumu - Reset, Play, Save, Playing.



3.4. att. Programmas stāvokļu diagramma

3.2.2.1 Stāvoklis “Reset”

Pēc starta pogas uzapešanās, programma pārceļas uz “Reset” stāvoklī. “Reset” stāvokļa mērķis ir monitoru tīrīšanā no jau iepriekš paradītas kombinācijas. Monitors kļūst melns uz vienu sekundi un pēc tam visi kvadrāti aizdegas ar ciāna krāsu uz vienu sekundi. Darbību soļus var redzēt attēlā 3.5.



3.5. att. “Reset” stāvoklī darbības soļi – 1. monitors kļūst melns 2. visi kvadrāti aizdegas ar ciāna krāsu

3.2.2.2 Stāvoklis “Play”

Kad pabeidzas stāvoklis “Reset”, programma pārceļas uz stāvokli “Play”. Stāvokļa “Play” mērķis ir jauna mēģinājumā ģenerēšana. Tiek definēts liels masīvs kurš sastāv no mazajiem masīviem. Katrs viens mazais masīvs atbilst vienam mēģinājumam. Mazais masīvs sastāv no diviem cipariem, kas ir pievienotas traucētāju kvadrātu skaits konkrētā mēģinājumā un vai šajā mēģinājumā vajag vai nevajag aizkrāsot mērķi. Lielais masīvs ir sakārtots pēc pieaugšanas secības - no sakuma mazie masīvi ar 5 pievienoties traucētājiem un bez mērķa, pēc tam masīvi ar 5 pievienotiem traucētājiem un ar mērķi, pēc tam masīvi ar 10 pievienotiem traucētājiem un bez mērķa u.t.t. Iekš “start” funkcijas liels masīvs tiek nejauši pārkārtots lai nebūtu iespēja zināt uz priekšu kāds mēģināmus tūlīt notiks. Dažādu mēģinājumā piemērus var redzēt attēlā 3.6. Algoritms priekš jaunas mēģinājumā veidošanas ir sekojošs:

- Nolasām no liela masīvā nākamo mazo masīvu.
- Pārbaudām vai vajag aizkrāsot mērķi - ja vajag tad izvēlamies nejaušu kvadrātu no 54 un aizkrāsojam to.
- Izvēlamies nejaušu ciparu no 0 līdz 54 un pārbaudām vai kvadrāts jau ir aizkrāsots.
- Ja konkrētais kvadrāts jau ir aizkrāsots, tad mēs vēlreiz izvēlamies nejaušu ciparu, kamēr neatradīsim neaizkrāsotu kvadrātu.

- Ja kvadrāts nav aizkrāsots, tad izvēlamies nejaušu ciparu no 0 līdz 3 priekš pievienotas krāsu izvēlēs un aizkrāsojam konkrētu kvadrātu.
- Kad nepieciešamu kvadrātu skaists ir aizkrāsots, tad tiek palaists laika hronometrs, un programma pārceļas uz stāvoklī “Playing”.

Autors vērtē, ka algoritms skan ļoti neefektīvi no programmēšanas perspektīvas, bet pieejams dators ir pietiekami spējīgs lai ātri izpildīt to funkcijā, ka pētāma persona neredz nekādā veida aiztures.



3.6. att. Mēģinājuma piemēri - bez mērķi un 10 traucētajiem ar pievienotu krāsu un ar mērķi un 15 traucētajiem ar pievienotu krāsu

3.2.2.3 Stāvoklis “Playing”

Kad programmas atrodas stāvoklī ”Playing”, tad pētāmais personai parādās iespējā nospiegt uz “HTC Vive” kontroles pogu lai apstiprinātu ievadu. Stāvoklis ir pieejams tikai 5 sekundes. Pēc pogas nospiešanās reģistrēšanas vai piecas sekundes laikā, notiek pārceļšana uz “Save” stāvoklī.

3.2.2.4 Stāvoklis “Save”

“Save” stāvokļa mērķis ir reakcijas laika saglabāšanā. Pēc saglabāšanas notiek pārbaudē vai visi mēģinājumi tiek parādīti - ja atbilde ir pozitīvā, tad programma pārceļas uz nobeiguma stāvoklī un rāda rezultātus, kuri tiek saglabāti. Ja tas nebija pēdējais mēģinājums tad notiek pārceļšana uz “Reset” stāvoklī.

4. EKSPERIMENTĀLĀ DAĻA

Eksperimentālā daļa notika klātienē, Raiņa Bulvārī 19, 3.stāva, 302. kabinetā. Galvenais iemesls – izmantotu virtuālas realitātes ķiveri nav iespējas palaist bez datora, tāpēc eksperimentu vajag veikt datorā atrašanas vieta. Studenti tiek aicināti piedalīties eksperimenta no 24 līdz 28.05. Uz uzaicinājumu atsaucas 12 studenti, ar kuriem tiek personīgi sarunāti tikšanas laiki, lai izvērtētos no lielu cilvēku skaitu vienā telpa. Nekādi personīgi dati (dzimums, vecums, u.t.t.) par šiem cilvēkiem netika vakti, jo tas nav svarīgs priekš eksperimenta būtības. Studentiem tikai bija prasīts redzes problēmas esamības un visi atbildēja negatīvi.

Paša virtuālas realitātes ķivere bija vienmēr dezinficētā, kabinetā vienmēr tiek atvērts logs, students un autors vienmēr bija maskas un 2m distance tika ievērotā. Studentu pozīcija ir apmēram 1,5m no virtuālā monitora, kura tiek nofiksētā ar nekustīgu krēslu un krēslā pozīcijā ir atzīmētā uz grīdas.

Eksperimentālās daļas norise ir sekojoša:

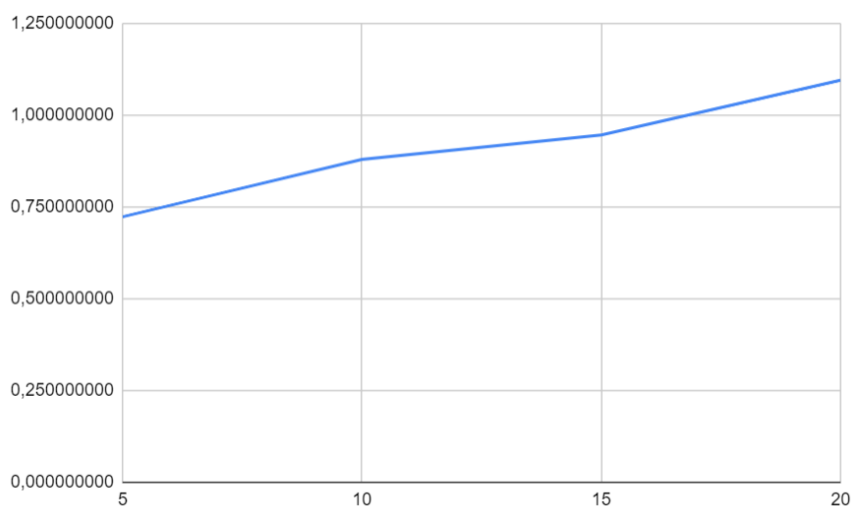
- Kabinets ir izvēdināts, virsmas un virtuāls ķivere ir dezinficētā.
- Autors, izmantojot random.org, veido jaunu nejaušu secību priekš fona krāsām.
- Sarunātā laika student atnāk uz 302. kabinetu un ir aicināts apsēties krēslā.
- Studentam tiek iedota ķiverē, kuru students apģērbs sev pats, un kontrolierīce.
- Student sēž uz krēslā un gaida komandu no autora.
- Autors palaida lietotni, un prasa vai students labi orientēja un vai students uztver virtuālo testu telpu.
- Pēc pozitīvās atbildes saņemšanas, tiek uzsākts meklēšanas test ar mērķi paskaidrot testa būtībā, - ka test strādā, ko vajag sagaidīt un ko vajag spiest.
- Tad kad pozitīvā atbilde par testa būtību saprašanu ir saņemta, autors saka testu no sakuma, ar pirmo no sienu krāsām.
- Students izpilda testu.
- Autors saglabā rezultātus
- Tad autors prasa kādas ir studentu sajūtas, lai pārbaudīt studentu stāvoklī.
- Kad pozitīvā atbilde no studenta ir saņemta, autos palaidu testu vēlreiz ar nākamo krāsu.
- Kad 4 krāsas tiek izietas, autos prasa studentu noņemt ķiveri un kontrolieri un padot autoram
- Students pamet telpu.

Tas bija oriģinālais eksperimentālās daļas plāns, bet testējos lietotni, autors pamanīja dīvainu efektu, ka pat zinot cik daudz ir mēģinājumu, autora sajūtas lika domāt ka mēģinājumu ir daudz mazāk. Ar velmi izanalizēt to efektu precīzāk, autors pievienoja pie eksperimentālas daļas nobeigumā divus jautājumus, kurus autors uzdot studentam pēc eksperimentāls daļas nobeigšanas, pirms students pametīs telpu:

- “Jūsaprāt, cik bija mēģinājumu?”
- ”Jūsaprāt, cik procenti no visiem mēģinājumiem bija ar mērķī?”

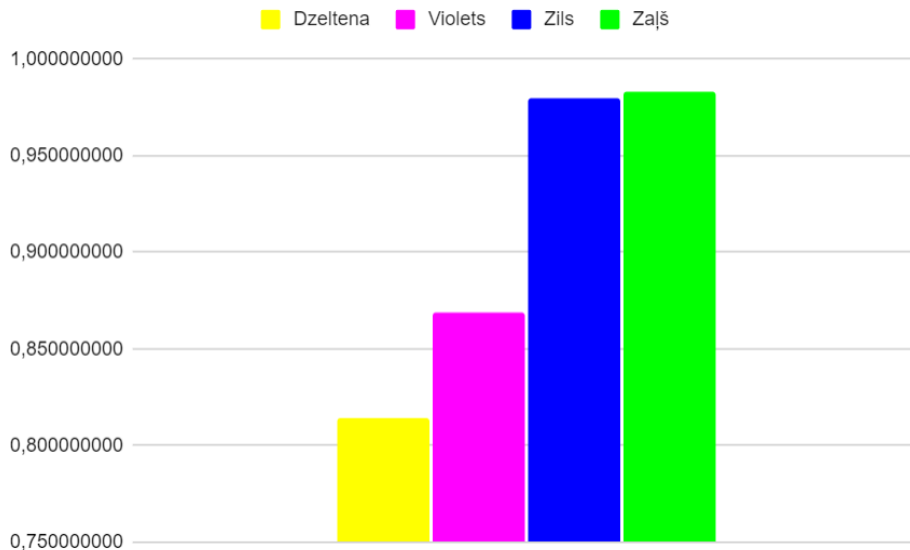
5. REZULTĀTI

Pēc veiksmīgi novadītas eksperimentālās daļas, dati tiek apkopoti un izdarīta analīze. Pirmo apskatīsim vidējo reakcijas laiku atkarīgi no pievienotu traucētāju skaitu, ko var redzēt attēlā 5.1. Progresija ir gandrīz lineārā, kas ir nesagaidāmi un nav raksturīgi priekš raksturiezīmes meklēšanas. Visdrīzāk tas ir saistīts ar pievienotu traucētāju esamību.



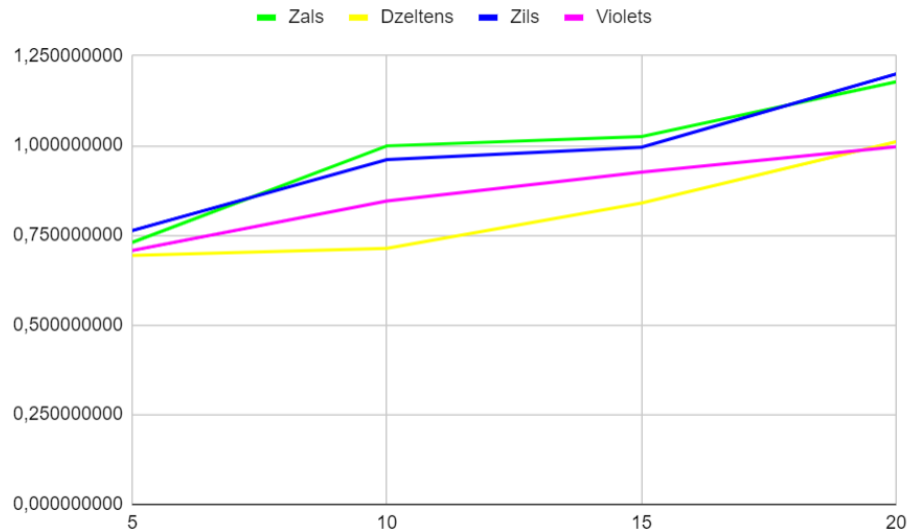
5.1. att. Vidējo reakcijas laiku atkarīgi no pievienotu traucētāju skaitu

Tālāk apskatīsim vidējo reakcijas laiku priekš katras sienu krāsas ko var redzēt attēlā 5.2. Var redzēt, ka dzeltena krāsa rada viss ātrāku laiku, violeta krāsas rada otra ātrāku laiku, un zila un zaļa rada gandrīz identiskas laiku, kuru var uzskatīt pat trešo ātrāku laiku. Šī secībā ir negaidītā. Zaļai un zilai krāsai tuvībā ir paskaidrojama, jo pieminētas krāsas ir vienādā attālumā no sarkana, bet pēc tādā secinājumā arī dzeltenai un violetai vajag būt vienādas laiks, bet tā nav.



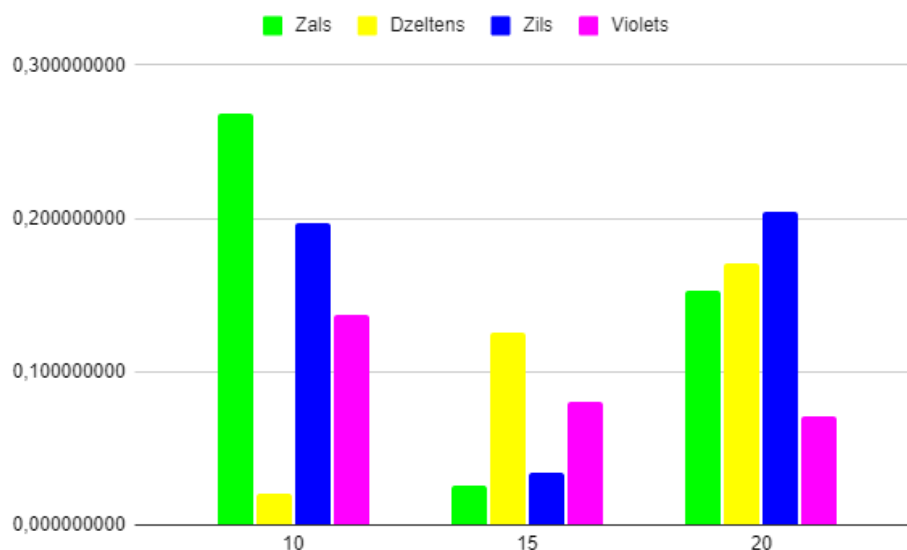
5.2. att Vidējais reakcijas laiks priekš katras sienu krāsas

Tagad apskatīsim vidējo reakcijas laiku, priekš katras sienu krāsas atkarīgi no pievienoto traucētāju skaitu ko var redzēt attēlā 5.3. Var redzēt dažādu uzvedīnu katrai krāsai. Pie pieciem pievienotiem traucētājiem, reakcijas laiks ir ļoti tuvs viens otram. Palielinot pievienotu traucētāju skaitu līdz 10, zaļa un zila krāsa reakcijas laiks ātri palielinās. Ir zināms ka citi pētījumi liecina, ka zilai krāsai ir saistība ar laika uztveri un ka zila krāsa tiek uztverta, kā ilgāka laika [14], tāpēc šāda uzvedībā ir sagaidāmā. Palielinot pievienotu traucētāju skaitu līdz 10 violets laiks palielinās lēnāk un dzeltena laiks gandrīz nepalielinās. Pie 15 pievienotiem traucētājiem, zils un zaļš gandrīz neizmainījās, violets palielinājās nedaudz un dzeltens palielinājās visstraujāk. Pie 20 pievienotiem traucētājiem, zila un zaļa krāsa atkal palielinājās uz lielu laika gabalu, violets un dzeltens palielinājās nedaudz.



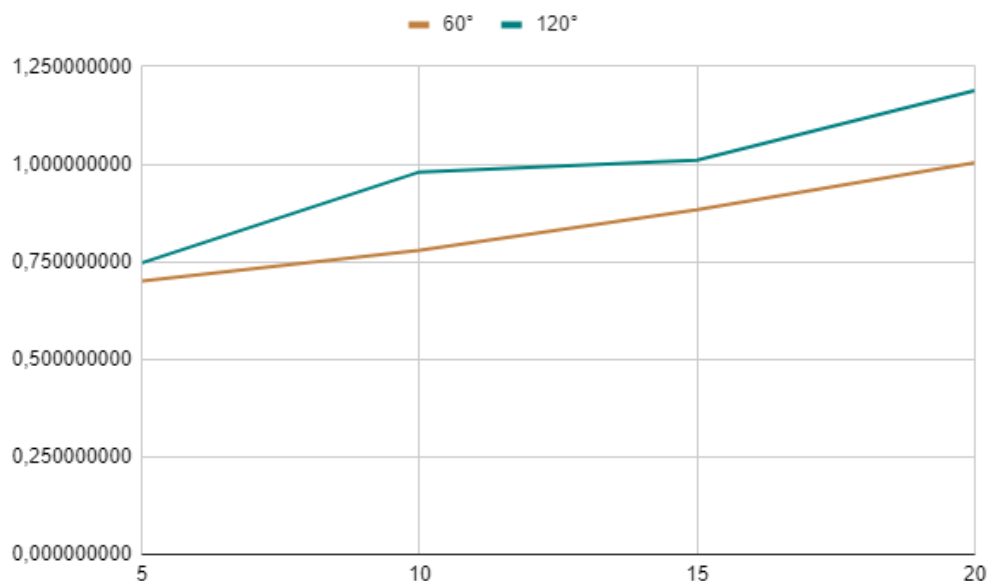
5.3. att Vidējais reakcijas laiks priekš katras sienu krāsas atkarīgi no pievienota traucētāju skaitu

Tālāk apskatīsim detalizētāk sienu krāsu un pievienotu traucētāju skaitu sakarību izmantojot citu perspektīvu. Pievienotu traucētāju skaits palielināšanās ietekme uz sienas krāsas vidējo reakcijas laiku starpību pret iepriekšējo traucētāju skaitu var redzēt attēla 5.4. Varam redzēt diezgan negaidītu uzvedību. Kad pievienotu traucētāju skaits palielinās no 5 uz 10, tad 3 krāsām bija liels pieaugums, bet dzeltenai gandrīz nebija. Kad pievienotu traucētāju skaits palielinājās no 10 uz 15, tad dzeltenai un violetai krāsai bija lielākais pieaugums, bet citam krāsam relatīvi mazs. Kad pievienotu traucētāju skaits palielinājās no 15 uz 20 tad 3 krāsam ir liels pieaugums, izņemto violetu. Tas ir dīvaina uzvedina, kura nebija sagaidītā, un nevar saskatīt darbības principus.



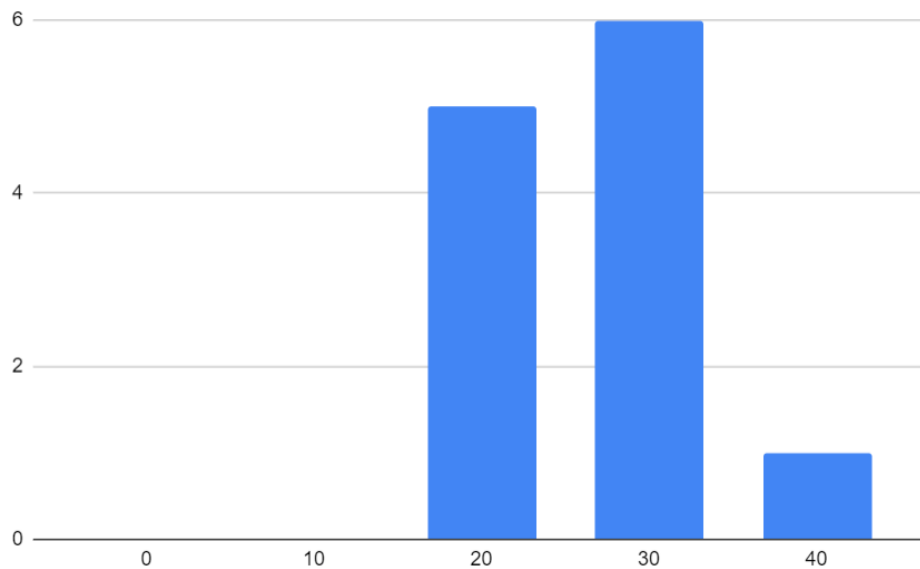
5.4. att Izmaiņas sienas krāsa vidējā reakcijas laiku starpību atkarīgi pret iepriekšējo pievienoto traucējumu skaitu.

Tālāk apskatīsim ka sienas krāsas pozīcijā iekš krāsu riņķi ietekme vidējo reakcijas laiku meklēšanas uzdevuma attiecība pret pievienotu traucētāju skaitu. Krāsas kas ir 120 grādi no sarkanas krāsa uz vienu un otru pusi ir apvienoti un krāsas kas 60 grādi no sarkana arī ir apvienoti un to mēs redzam attēlā 5.5. Varam redzēt ka krāsas kuras ir 120 grādi no sarkana ietekme vairāk uz meklēšana uzdevumu izpildi un rāda lēnāk reakcijas laiku, nekā 60 grādu krāsas kuri rada atraku reakcijas laiku.



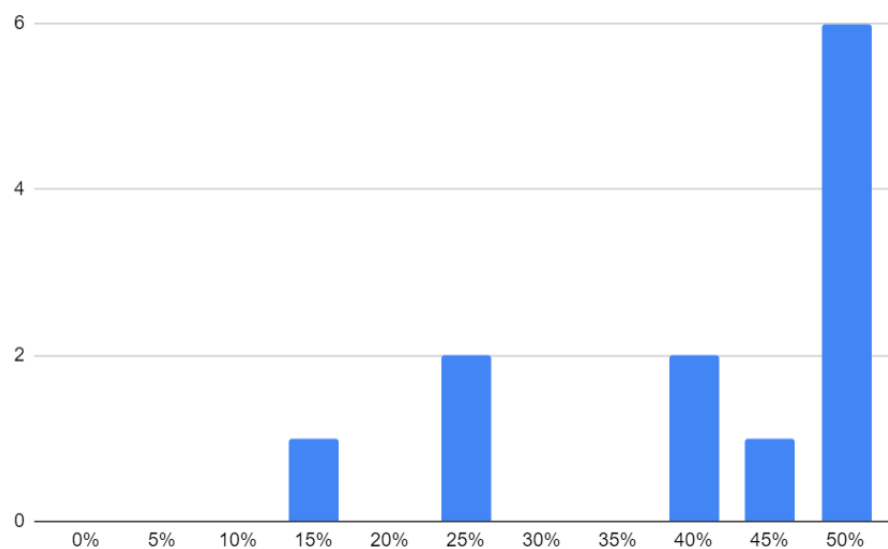
5.5. att. Izmaiņas vidējā reakcijas laika krāsas grupētas pēc pozīcijā krāsu riņķi atkarība no pievienotu traucētāju skaitu

Tupinām analīzi ar nobeigumā jautājumiem. Savilkumu par pirmo jautājumu var redzēt attēlā 5.6. Var redzēt ka lielākā daļa no studentiem atbildēja mazāk vai daudz mazāk nekā īsts mēģinājumu skaits, kas ir negaidītais efekts un darba autors nevar pateikt kas to izraisa.



5.6. att Atbildes uz jautājumu "Jūsaprāt, cik bija mēģinājumu?"

Apkopojumu par otro jautājumā atbildes var redzēt attēla 5.7. Varam redzēt ka gandrīz puse atbildēja pareizi, un otra puse dev atbildes kuri ir mazāk vai daudz mazāk nekā pareizi 50%. Autors var secināt ka mērķu daudzu uztvere strādā sagaidāmi.



5.7. att Atbildes uz jautājumu "Jūsaprāt, cik procenti no visiem mēģinājumiem bija ar mērķi?"

SECINĀJUMI

Bakalaura darbā ir veikts eksperiments lai pārbaudītu dažādu krāsu ietekmi uz meklēšanu uzdevumu izpildi virtuālā vide.

Darba pirmā daļa autors izpētīja materiālus un izveidoja plānu priekš meklēšanas testam iekš virtuālas vides. Darba otra daļa tiek aprakstīta lietotnes izstrādē, ar kuru tiek veikts eksperiments. Darba Trešā daļa tiek aprakstīts pati eksperimentālā daļa – veikšanas secība, un procedūrā.

Izvērtējot iegūtos rezultātus, autors konstatē, krāsas kas ir tuvāk sarkanam iekš krāsu riņķi rada vidēji mazākus reakcijas laikus, nekā krāsa kas ir tālāk no sarkanā. Reakcijas laika izmaiņas atkarība no pievienotu traucētāju skaitu nav vienāds katrai krāsai un nav prognozējams. Pretējas krāsas kas ir tuvāk sarkanam iekš krāsu riņķu un ir vienādā attāluma no sarkanā, rada dažādu reakcijās laiku, kad pretējas krasas kuri atrodas tālāks no sarkanas krasas un ir vienādā attālumā, rada līdzīgus reakcijas laikus.

Tiek konstatēts, ka šādi faktori var tiek uzlaboti nākamās pētījumos:

- Palielināt cilvēku skaitu kas piedalās eksperimentālā daļā
- Veikts eksperimentālu daļu ar vienu krāsu, ar dažādu piesātinājumu.
- Neizmantojot pavienotas traucētājus

IZMANTOTĀ LITERATŪRA

[1] Sutherland, I. E. (1965). "The Ultimate Display. Multimedia: From Wagner to Virtual Reality."

[2] Slater, M. (2009). Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 364, 3549–3557. doi: 10.1098/rstb.2009.0138

[3] Heeter, C. (1992). Being there: the subjective experience of presence. *Presence* 1, 262–271. doi: 10.1162/pres.1992.1.2.262

[4] Baños, R., Botella, C., García-Palacios, A., Villa, H., Perpiñá, C., and Gallardo, M. (2009). Psychological variables and reality judgment in virtual environments: the roles of absorption and dissociation. *Cyberpsychol. Behav.* 2, 143–148. doi: 10.1089/cpb.1999.2.143

[5] Heilig, M. (1962). Sensorama simulator. U.S. Patent No - 3, 870. Virginia: United States Patent and Trade Office.

[6] Steuer J. Defining virtual reality: Dimensions determining presence *Communication in the age of virtual reality*, Lawrence Erlbaum Associates, Hillsdale, NJ (1995), pp. 33-56

[7] Witmer B.G., Singer M.J. Measuring presence in virtual environments: A presence questionnaire *Presence*, 7 (3) (1998), pp. 225-240

[8] HTC Vive [tiešsaiste] – [atsauces 25.05.2021.]. Pieejams: <https://www.vive.com/eu/product/vive/>

[9] Statista [tiešsaiste] – [atsauces 25.05.2021.]. Pieejams: <https://www.statista.com/statistics/321059/game-engines-used-by-video-game-developers-uk/>

[10] Assets Store [tiešsaiste] – [atsauces 25.05.2021.]. Pieejams: <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>

[11] Treisman, A. M.; Gelade, G (1980). "A feature-integration theory of attention". *Cognitive Psychology*. 12 (1): 97–136. doi:10.1016/0010-0285(80)90005-5. PMID 7351125.

[12] Amini Vishteh R, Mirzajani A, Jafarzadehpour E, Darvishpour S. Evaluation of Simple Visual Reaction Time of Different Colored Light Stimuli in Visually Normal Students. *Clin Optim (Auckl)*. 2019;11:167-171 <https://doi.org/10.2147/OPTO.S236328>

[13] Color ring [tiešsaiste] – [atsauce 25.05.2021]. Pieejams: <https://upload.wikimedia.org/wikipedia/commons/9/90/Colors-i54-ring.png>

[14] <https://www.nature.com/articles/s41598-018-19892-z>

PIELIKUMI

Pielikums 1. Sienas krāsu mainīšanas kods

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Walls_Change_Colors : MonoBehaviour
{
    private GameObject wall_left;
    private GameObject wall_rigth;
    private GameObject wall_back;
    private GameObject wall_front;
    private GameObject ciling;
    private GameObject floor;
    private GameObject table;
    void Start()
    {
        wall_back = GameObject.Find("back_wall");
        wall_front = GameObject.Find("front_wall");
        wall_left = GameObject.Find("left_wall");
        wall_rigth = GameObject.Find("righth_wall");
        ciling = GameObject.Find("ciling");
        table = GameObject.Find("Table");
        floor = GameObject.Find("Floor");
    }
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Keypad0))
        {
            wall_back.GetComponent<Renderer>().material.color = Color.green;
            wall_front.GetComponent<Renderer>().material.color = Color.green;
            wall_left.GetComponent<Renderer>().material.color = Color.green;
            wall_rigth.GetComponent<Renderer>().material.color = Color.green;
            ciling.GetComponent<Renderer>().material.color = Color.green;
        }
        if (Input.GetKeyDown(KeyCode.Keypad1))
        {
            wall_back.GetComponent<Renderer>().material.color = new Color(1f,1f,0f,1f);
            wall_front.GetComponent<Renderer>().material.color = new Color(1f,1f,0f,1f);
            wall_left.GetComponent<Renderer>().material.color = new Color(1f,1f,0f,1f);
            wall_rigth.GetComponent<Renderer>().material.color = new Color(1f,1f,0f,1f);
            ciling.GetComponent<Renderer>().material.color = new Color(1f,1f,0f,1f);
        }
        if (Input.GetKeyDown(KeyCode.Keypad2))
        {
            wall_back.GetComponent<Renderer>().material.color = Color.blue;
            wall_front.GetComponent<Renderer>().material.color = Color.blue;
        }
    }
}
```

```
    wall_left.GetComponent<Renderer>().material.color = Color.blue;
    wall_rigth.GetComponent<Renderer>().material.color = Color.blue;
    ciling.GetComponent<Renderer>().material.color = Color.blue;
}
if (Input.GetKeyDown(KeyCode.Keypad3))
{
    wall_back.GetComponent<Renderer>().material.color = Color.magenta;
    wall_front.GetComponent<Renderer>().material.color = Color.magenta;
    wall_left.GetComponent<Renderer>().material.color = Color.magenta;
    wall_rigth.GetComponent<Renderer>().material.color = Color.magenta;
    ciling.GetComponent<Renderer>().material.color = Color.magenta;
}
}
}
```

Pielikums 2. Testa izpildes kods

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Valve.VR;

public class Change_Color : MonoBehaviour
{
    private GameObject[] Pixels = new GameObject[54];
    private int[][] Rounds = new int[40][]{new int[2]{5,0},new int[2]{5,0},new
int[2]{5,0},new int[2]{5,0},new int[2]{5,0},new int[2]{5,1},new int[2]{5,1},new
int[2]{5,1},new int[2]{5,1},new int[2]{5,1},new int[2]{10,0},new int[2]{10,0},new
int[2]{10,0},new int[2]{10,0},new int[2]{10,0},new int[2]{10,1},new int[2]{10,1},new
int[2]{10,1},new int[2]{10,1},new int[2]{10,1},new int[2]{15,0},new int[2]{15,0},new
int[2]{15,0},new int[2]{15,0},new int[2]{15,1},new int[2]{15,1},new
int[2]{15,1},new int[2]{15,1},new int[2]{15,1},new int[2]{20,0},new int[2]{20,0},new
int[2]{20,0},new int[2]{20,0},new int[2]{20,0},new int[2]{20,1},new int[2]{20,1},new
int[2]{20,1},new int[2]{20,1},new int[2]{20,1}};
    private Color[] Colors = {new Color(1f,0.5f,0f,1f), new Color(1f,1f,0f,1f), new
Color(1f,0f,0.5f,1f), new Color(1f,0f,1f,1f)};
    private int Round = 0;
    private float start_time = 0;
    private bool Process = false;
    private bool Save = false;
    private bool Reset = false;
    private bool Play = false;
    private bool Playing = false;
    private string Result = "";
    GameObject Pixel;
    IEnumerator co;
    void Start()
    {
        for(int i=0;i<40;i++)
        {
            int[] temp = Rounds[i];
            int ran = Random.Range(0,40);
            Rounds[i] = Rounds[ran];
            Rounds[ran] = temp;
        }
        for(int i=0;i<54;i++)
        {
            Pixels[i] = GameObject.Find("Quad ("+(i+1)+)");
        }
    }
    void Update()
    {
        if
(SteamVR_Actions._default.Press.GetStateDown(SteamVR_Input_Sources.Any))
        {
            if(Playing == true)
```

```

        {
            Playing = false;
            Save = true;
        }
    }
    if (Input.GetKeyDown(KeyCode.Keypad7))
    {
        Process = true;
        Reset = true;
    }
    if (Process)
    {
        if (Reset)
        {
            Reset = false;
            StartCoroutine(Reset_Routine());
        }
        if(Play)
        {
            Play = false;
            co = Plays_Routine();
            StartCoroutine(co);
        }
        if(Save)
        {
            int[] tmp_rnd = Rounds[Round];
            Result = Result + (tmp_rnd[0] + "," + tmp_rnd[1] + "," + (Time.time -
start_time) + "\n");
            Save=false;
            StopCoroutine(co);
            if (Round == 39){
                Process = false;
                Debug.Log(Result);
                for(int i=0;i<54;i++){
                    Pixels[i].GetComponent<Renderer>().material.color = Color.black;
                }
            }
            else{
                Round = Round + 1;
            }
            Reset = true;
        }
    }
}
IEnumerator Reset_Routine()
{
    for(int i=0;i<54;i++){
        Pixels[i].GetComponent<Renderer>().material.color = Color.black;
    }
    yield return new WaitForSeconds(1);
    for(int i=0;i<54;i++){
        Pixels[i].GetComponent<Renderer>().material.color = Color.cyan;
    }
}

```

```

    }
    yield return new WaitForSeconds(1);
    Play = true;
}
IEnumerator Plays_Routine()
{
    int[] tmp_rnd = Rounds[Round];
    if (tmp_rnd[1] == 1){
        int ran = Random.Range(0,54);
        Pixels[ran].GetComponent<Renderer>().material.color = Color.red;
    }
    for(int i=0;i<tmp_rnd[0];i++)
    {
        int rand = Random.Range(0,54);
        if(Pixels[rand].GetComponent<Renderer>().material.color != Color.cyan)
        {
            do
            {
                rand = Random.Range(0,54);
            } while (Pixels[rand].GetComponent<Renderer>().material.color !=
Color.cyan);
        }
        int cl = Random.Range(0,3);
        Pixels[rand].GetComponent<Renderer>().material.color = Colors[cl];
    }
    start_time = Time.time;
    Playing = true;
    yield return new WaitForSeconds(3);
    Save = true;
}
}

```

Bakalaura darbs „Krāsu ietekme meklēšanas uzdevumu izpildē virtuālā vidē” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti.

Autors: Georgijs Pliska 31.05.2021.

Rekomendēju darbu aizstāvēšanai

Vadītāja: Asociētais profesore, Dr.geol., Līga Zariņa 31.05.2021.

Recenzents: profesors, Dr.habil.dat., Juris Borzovs

Darbs iesniegts Datorikas fakultātē 31.05.2021.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

___.06.2021. prot. Nr. ____.

Komisijas sekretāre: Asociētais profesore, Dr.geol., Līga Zariņa