

Eiropas Savienības Erasmus+ programmas Pamatdarbības Nr.2 (KA2) Sadarbības partnerības projekts skolu izglītības sektorā “STEM ekosistēmas dažādošana”
Nr. 2024-1-LV01-KA220-SCH-000250755

„Metodika nodarbībām „Moderno tehnoloģiju studija” Ozolnieku vidusskolai”



Līdzfinansē
Eiropas Savienība



Latvijas
Biozinātņu un
tehnoloģiju
universitāte

Finansē Eiropas Savienība. Tomēr paustie uzskati un viedokļi ir tikai autora(-u) uzskati un viedokļi un ne vienmēr atspoguļo Eiropas Savienības vai Valsts izglītības attīstības aģentūras (VIAA) uzskatus un viedokļus. Ne Eiropas Savienību, ne VIAA nevar saukt pie atbildības par tiem.

Saturs

Ievads	4
1. modulis. Dažādi attēlu rediģēšanas rīki pārlūkprogrammā	5
1.1. modulis. Attēlu rediģēšana ar iPiccy	5
Darba lapa „Attēlu rediģēšana ar iPiccy”	7
1.2. modulis. Attēlu rediģēšana ar Pixlr Editor	11
Darba lapa „Attēlu rediģēšana ar Pixlr”	13
2. modulis. 3D modelēšana ar 3D drukāšanu	18
2.1. modulis. Atslēgu piekariņš ar tavu vārdu	18
Darba lapa „Atslēgu gredzens (atsēgu piekariņš) ar tavu vārdu”	19
2.2. modulis. Vāzes izgatavošana	22
Darba lapa „Vāzes veidošana”	23
2.3. modulis. Uzglabāšanas kastes izveide	27
Darba lapa „Uzglabāšanas kastes izgatavošana”	28
2.4. modulis. 3D mājas izveide ar animāciju	30
Darba lapa „3D mājas izveide ar animāciju”	30
3. modulis. Video veidošana pārlūkprogrammā	35
3.1. modulis. Video veidošana ar Magisto	35
Darba lapa „Video veidošana ar Magisto”	36
4. modulis. Ievads Python un Arduino	37
4.1. modulis. Programmēšanas vides konfigurēšana	37
Darba lapa „Processing saskarnes elementu apguve”	40
Darba lapa „Koordinātu apstrāde”	41
4.2. modulis. Pirmie soļi programmēšanā ar grafiku	43
Darba lapa „Grafikas programmēšana”	45
4.3. modulis. Ievads mainīgo lielumu lietošanā	48
Darba lapa „Mainīgie darbībā”	50
Darba lapa „Setup vs Draw Challenge”	51
4.4. modulis. Radošais projekts: no koda līdz spēlei	53
Darba lapa „Radošais programmēšanas uzdevums: lidojoša bumba”	55
4.5. modulis. Ievads Arduino komponentos	57
Darba lapa „Arduino CTC 101 komponentu identificēšana”	59
5. modulis. Vadības plātnes programmēšanas pamati	68
5.1. modulis. Plates uzstādīšana un pieslēgšana IDE (integrētā attīstības vidē)	68
Darba lapa „IDE iepazīšana”	69
5.2. modulis. LED mirgošanas un skaņas radīšanas programmēšana, izmantojot Piezo Speaker	72
Darba lapa „LED izpēte”	75
Darba lapa „Runātāja iepazīšana”	76
Darba lapa „EducationShield bibliotēkas izpēte”	77

5.3. modulis. Projekta izstrāde: nelielas elektroniskas spēles, kas simulē sporta spēles, izveide un spēlēšana	79
Darba lapa „Projekta veidne”	81
6. modulis. Analogo signālu un seriālās komunikācijas pamati	85
6.1. modulis. Analogo signālu nolasīšana un rakstīšana, kā arī analogu sensoru izmantošana robotu programmēšanā	85
6.2. modulis. Sērijveida datu pārraide starp vadības plati un datoru datu nosūtīšanai un saņemšanai	87
Darba lapa „Sērijas komunikācijas izpēte”	88
6.3. modulis. Projekta izstrāde: elementa izveide, kas simulē kādu procesu (mūzikas radīšana, ziņojuma parādīšana, ritma atskaņošana)	89
7. modulis. Darbs ar motoriem	92
7.1. modulis. Dzinēju veidi un standarta servodzinēji	92
Darba lapa “Servo motoru izpēte”	94
7.2. modulis. Nepārtraukta rotācija un ieejas vadība	96
Darba lapa “Ievades vadības servodzinēju izaicinājums”	98
7.3. modulis. Rāpojošo radību konkurss	100
Darba lapa „Rāpojošo radību izaicinājums”	102

Ievads

Šī metodika ir izstrādāta Erasmus+ programmas projekta „STEM ekosistēmas dažādošana“ (Nr. 2024-1-LV01-KA220-SCH-000250755) ietvaros, kas veltīts sadarbības partnerībām skolu izglītībā. Programma tika izstrādāta sadarbībā starp Ozolnieku vidusskolu un Latvijas Lauksaimniecības un tehnoloģiju universitātes (LBTU) akadēmisko personālu. Metodikas autores ir Tatjana Rubina, Natalja Vronska, Jekaterina Smirnova, Baiba Briede un Natalija Sergejeva.

Metodikas „Modern Technology Studio” mērķis ir atbalstīt skolēnu digitālo radošumu un tehnisko kompetenci, izmantojot modernus pārlūkprogrammu rīkus attēlu apstrādei, 3D modelēšanai un multimediju satura veidošanai. Programma apvieno praktiskas, uz darbību vērstas aktivitātes ar digitālo dizaina domāšanu, lai palīdzētu skolēniem attīstīt kompetences vizuālajā komunikācijā, programmēšanas loģikā un inovācijās. Turklāt metodika iepazīstina skolēnus ar robotikas, programmēšanas un vadības sistēmu pamatiem.

Šī metodika apvieno viegli pieejamus tiešsaistes rīkus un fizisko datoriku, lai veicinātu skolēnu radošumu, problēmu risināšanas prasmes un tehnoloģisko kompetenci. Mūsdienīgi pārlūkprogrammu rīki, piemēram, iPiccy, Pixlr un Ezgif, iepazīstina skolēnus ar attēlu rediģēšanas, dizaina kompozīcijas un animācijas pamatiem. Skolēni apgūst, kā uzlabot fotogrāfijas, radīt vizuālos efektus, veidot kolāžas un izveidot GIF animācijas, tādējādi iegūstot būtiskas digitālās prasmes, kas savieno mākslu un tehnoloģijas.

Papildus radošajiem digitālajiem medijiem šī metodika aptver arī robotiku un programmēšanu, izmantojot Python un Arduino. Skolēni apgūst programmēšanas pamatus, iepazīst mainīgos un cilpas, kā arī izpēta, kā programmatūra kontrolē fiziskās sastāvdaļas. Strādājot ar Arduino CTC 101 sistēmu, skolēni iepazīst sensorus, gaismas diodes, skaļruņus, motorus un vadības plātnes, lai izveidotu interaktīvus prototipus un vienkāršus robotus. Moduļi attīstās no ievada eksperimentiem ar digitālajiem signāliem līdz projektveida mācībām, kurās skolēni projektē nelielus robotizētus mehānismus un interaktīvas elektroniskas spēles.

Apvienojot digitālo radošumu ar robotiku un programmēšanu, šī metodika piedāvā visaptverošu, starpdisciplināru pieeju STEM izglītībai. Skolēni apgūst ne tikai tehniskās prasmes, bet arī pārliecību par savām spējām risināt problēmas, būt inovatīviem un strādāt komandā – tās ir galvenās prasmes, kas nepieciešamas, lai iekļautos mūsdienu tehnoloģiju ekosistēmā

1. modulis. Dažādi attēlu rediģēšanas rīki pārlūkprogrammā

Mērķis: Attīstīt prasmes attēlu rediģēšanā, retušēšanā un uzlabošanā.

Pārskats:

1.1. modulis. Attēlu rediģēšana ar iPiccy

1.2. modulis. Attēlu rediģēšana ar Pixlr

1.1. modulis. Attēlu rediģēšana ar iPiccy

Laiks/stundas: Kopā 10 stundas.

4 stundas 1.1. modulim.

6 stundas 1.2. Modulim.

Nodarbību plāns:

1. praktiskā nodarbība

Ievadfāze (10 min.)

- Uzmanības pievēršana: skolotājs ar projektoru rāda fotogrāfijas ar atšķirīgu kvalitāti, spilgtumu un kontrastu.
- Par mērķi un sasniedzamajiem rezultātiem: skolotājs mudina skolēnus formulēt nodarbības mērķi.

Iesaistīšanās fāze (30 min.)

- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursu iPiccy un pamata darba rīkiem. Skolēni patstāvīgi izpēta tiešsaistes resursu iPiccy.
- Informācijas apkopošana un apspriešana ar skolēniem.

2. praktiskā nodarbība

Iesaistīšanās fāze (25 min.)

- Jaunās informācijas izmantošana: skolēni individuāli strādā ar darba lapu "Attēlu rediģēšana ar iPiccy", uzdevumi 1–2.

Refleksijas fāze (15 min.)

- Atsauksmes: Ko jūs šodien uzzinājāt par tiešsaistes resursu iPiccy? Kas izdevās? Kas neizdevās? Ko vēl plānojat apgūt?

3. praktiskā nodarbība

Ievadfāze (40 min.)

- Iepriekšējo zināšanu atkārtošana par dažādiem iPiccy rīkiem.
- Diskusija ar skolēniem par spilgtumu un kontrastu. Kā to var uzlabot? Kāpēc tas ir nepieciešams?

- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursa iPiccy rīku „Foto efekti”. Jaunas informācijas pielietošana: skolēni individuāli strādā ar uzdevumu „Attēla rediģēšana ar iPiccy” 3.–4. uzdevumu.

4. praktiskā nodarbība

Iesaistīšanās fāze (25 min.)

- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursa iPiccy rīku „Retušēšana”. Skolotājs iepazīstina skolēnus ar meitenes attēla video veidošanu, izmantojot mākslīgo intelektu. Jaunās informācijas izmantošana: skolēni individuāli strādā ar uzdevumu „Attēlu rediģēšana ar iPiccy” 5.–6. uzdevumu.

Refleksijas fāze (15 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās? Vai vēlētos izmantot šo tiešsaistes resursu nākotnē? Kāpēc vai kāpēc ne?

•

Darba lapa „Attēlu rediģēšana ar iPiccy”

1. uzdevums. Fotogrāfijai nepieciešamas dažādas korekcijas

Piemēram



sākumā



pabeigta

- Iztaisnoji fotoattēlu (rīks „Rotate & Flip”), mainot „Straighten” vērtību uz 3, pēc tam nospiediet „Apply”.
- Brīvi (pēc saviem ieskatiem) apgrieziet nevajadzīgās fotoattēla malas (rīks „Crop Attēls”), pēc tam nospiediet „Apply”.
- Jāmaina fotoattēla spilgtums un kontrasts (rīks Light & Contrast), mainot Brightness: 39 un Contrast: 4, pēc tam nospiediet Apply.
- Jāuzlabo krāsu intensitāte (rīks Colors), mainot Temperature: 34.
- Jāizpludina fons (rīks Blur Image), mainot Strength: 60, tad izvēlieties otu (Effect mask/ Original) un pārkrāsojiet stiklu, atjaunojot stikla redzamību. Ja jums atkal jāizpludina fona fragments, jums jāmaina ota: Effect mask/ Effect.
- Nepieciešams izveidot melnbaltu akcentu izvēlētajam (pēc jūsu patikas) objektam (rīks Black & White), pēc tam izvēlieties otu un aizkrāsot glāzi, atjaunojot glāzes krāsu.

2. Uzdevums. No fotogrāfijas ir jāizņem nevajadzīgais objekts (jebkurš attēls)

Piemēram



sākumā

pabeigta

- Atrodiet un lejupielādējiet jebkuru attēlu no bezmaksas attēlu krātuves, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.
- Izvēlieties rīku „Clone“ (Klons), mainiet otas izmēru, pēc tam ar zaļo apli noklikšķiniet uz nevajadzīgā objekta un ar balto apli – uz vietas, kurā tiks izveidots klons.

3. uzdevums. Svarīgas informācijas paslēpšana fotogrāfijā (jebkura automašīna)

Piemēram



sākumā



pabeigta

- Atrodiet un lejupielādējiet jebkuru attēlu no bezmaksas attēlu meklēšanas platformas, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.
- Kreisajā pusē pārslēdziet rīkus uz Foto efektiem
- Jāpievieno pikseļi, piemēram, uz BMW numura (rīks Focal Pixelate), mainot fokusa izmēru: 10, fokusa cietību: 50, izbalēšanu: 0, apgrieztu efektu. Atkārtojiet to 3–4 reizes.

4. uzdevums. Kolāžas veidošana (4 attēli pēc izvēles)

Piemēram



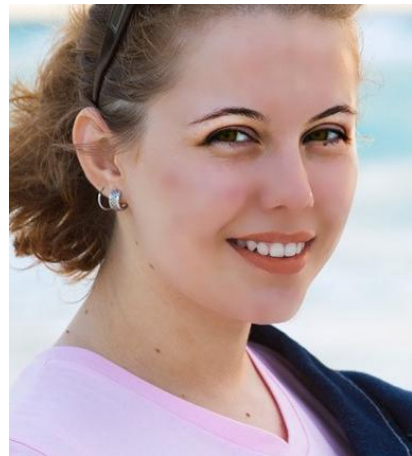
- Atrodiet un lejupielādējiet 4 attēlus no bezmaksas attēlu krātuves, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.
- Kreisajā pusē pārslēdzieties uz rīku „Foto efekti“.
- Pirmajam attēlam: izmantojiet rīku „Orton Effect“, mainot parametrus: Bloom: 50, Brightness: 80, Fade: 0. Saglabājiēt datorā.
- Otrajam attēlam: jāizmanto rīks „Retro Comic“, mainot „Dot size”: 5, „Angle”: 35, „Brightness”, „Contrast”, „Fade”: 0, „Blend Mode”: „Color”. Saglabājiēt datorā.
- Trešajam attēlam: jāizmanto rīks „Rainbow”, mainot „Fade”: 0, „Blend Mode”: „Multiply”. Un rīku „Scanlines”, mainot iestatījumus: līnijas izmērs: 11, izbalēšana: 50. Saglabājiēt datorā.
- Ceturtajam attēlam: jāizmanto rīks „Duotone”, mainot pirmo: zils, otro: zaļš, Contrast: 20, „Fade”,.: 0. Un rīku „Gritty”, mainot Darkness: 50, Noise: 70, Fade: 0. Saglabājiēt datorā.
- Pāriet uz kolāžu veidotāju (Editor/ Collage) un izvēlieties kolāžas izkārtojumu, piemēram, 4 attēlus (1 attēls augšā, 3 attēli apakšā). Pievienojiet nesen uzlabotos attēlus. Var izveidot jebkura attēla horizontālo spoguļattēlu.
- Kolāžas iestatījumos mainiet Spacing: 20, Roundness: 50. Mainiet fona krāsu uz tādu, kas jums vislabāk patīk.

5. uzdevums. Fotogrāfiju retušēšana

Piemēram



sākumā



pabeigta

- Kreisajā pusē pārslēdzieties uz rīku „Retouch“
- Jānoņem pūtītes (rīks „Blemish Fixer“).
- Jāizlīdzina sejas āda (rīks „Airbrush“), mainot otas izmēru uz 10 un izplūdināšanu uz 20 %. Zobi, acis, uzacis – nepieskarieties!
- Jāpadara acis izteiksmīgākas (rīks Mascara), mainot otas izmēru: 7 un izplūdināšanu: 35%.
- Mainiet acu krāsu uz tādu, kas jums patīk vislabāk (rīks EyeColor), mainot otas izmēru: 10 un izplūdināšanu: 20%.
- Jābalina zobi (rīks Teeth Whiten), mainot otas izmēru: 7 un izplūdināšanu: 0 %.

- Krāsojiet lūpas, mainot krāsu uz tādu, kas jums patīk vislabāk (rīks Lip Color), mainot otas izmēru: 4 un izplūdināšanu: 40 %.

6. uzdevums. Izveidojiet video no šī attēla, izmantojot mākslīgo intelektu

Izveidojiet video no šī attēla, izmantojot mākslīgo intelektu (www.vidu.com) (nepieciešams Google vai Apple konts).

Izvēlieties „Image to Video“, pēc tam augšupielādējiet meitenes fotogrāfiju JPG formātā, mainiet modeli uz „Vidu 1.5“ un noklikšķiniet uz pogas „Create“.

Turpiniet, izvēlieties “Reference to Video” (Atsauce uz video), pēc tam augšupielādējiet meitenes foto JPG formātā, ievadiet papildu norādījumus zem foto (piemēram, meitenei jāskatās kamerā, jāsmaida un jāvalkā saulesbrilles), pēc tam noklikšķiniet uz pogas “Create” (Izveidot).

1.2. modulis. Attēlu rediģēšana ar Pixlr Editor

Mācību mērķis:

Pēc 1.2. moduļa apguves studenti spēs izprast, kā rediģēt un retušēt attēlus, izmantojot tiešsaistes rīku Pixlr Editor.

Materiāli:

- Darba lapa „Attēlu rediģēšana ar Pixlr Editor”.
- Digitālie attēli.
- Ierīces ar interneta pieslēgumu (portatīvie datori/planšetdatori) tiešsaistes resursa Pixlr Editor izmantošanai.

Mācību metodes:

- demonstrācija;
- problēmu risināšana;
- praktisks uzdevums;
- mācību diskusija.

Avoti: <https://pixlr.com/editor/> /

Nodarbības plāns:

1. praktiskā nodarbība

Ievada fāze (10 min.)

- Uzmanības pievēršana: skolotājs ar projektoru rāda fotogrāfijas ar atšķirīgu kvalitāti, sarežģītību un kontrastu.
- Par mērķi un sasniedzamajiem rezultātiem: skolotājs mudina skolēnus formulēt moduļa mērķi.
- Iesaistīšanās fāze (30 min.)
- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursu Pixlr, kurā ir darba rīki un daudzas paneļi: Slāņi, Vēsture, Korekcija un Filtrs. Skolēni

patstāvīgi izpēta tiešsaistes resursu Pixlr. Informācijas apkopošana un apspriešana ar skolēniem.

2. praktiskā nodarbība

Iesaistīšanās fāze (25 min.)

- Jaunas informācijas izmantošana: skolēni individuāli strādā ar darba lapu “Attēlu rediģēšana ar Pixlr”, uzdevumi 1–2.
- Refleksijas fāze (15 min.)
- Atsauksmes: Ko jūs šodien uzzinājāt par tiešsaistes resursu Pixlr? Kas izdevās? Kas neizdevās? Ko vēl plānojat apgūt?

3. praktiskā nodarbība

Ievadfāze (40 min.)

- Iepriekšējo zināšanu atkārtošana par dažādiem Pixlr rīkiem. Diskusija ar skolēniem par Pixlr tiešsaistes resursa līdzību ar Photoshop. Par slāņu paneli un nepieciešamību to vienmēr redzēt. Kā to var pievienot, ja slāņu panelis nav redzams pēc noklusējuma? Kāpēc tas ir nepieciešams? Kur var mainīt spilgtumu, kontrastu, krāsu līdzsvaru, tonis un piesātinājumu?
- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursa Pixlr sajaukšanas režīmu opcijām un necaurspīdīgumu.
- Jaunās informācijas pielietošana: skolēni individuāli strādā ar uzdevumu „Attēlu rediģēšana ar Pixlr” (3.–4. uzdevums).

4. praktiskā nodarbība

Iesaistīšanās posms (30 min.)

- Jaunās informācijas pielietošana: skolēni individuāli pabeidza uzdevumu „Attēlu rediģēšana ar iPiccy” (4. uzdevums).

Refleksijas posms (10 min.) Atsauksmes: Kas izdevās? Kas neizdevās?

5. praktiskā nodarbība

Iesaistīšanās posms (40 min.)

- Iepriekšējo zināšanu atjaunošana par dažādiem Pixlr rīkiem. Diskusija ar skolēniem par filtriem. Kur atrodams sajaukšanas režīms un necaurspīdīgums? Kā var mainīt slāņa nosaukumu?
- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar bezmaksas tiešsaistes GIF veidotāju. Skolotājs demonstrē izveidoto GIF animāciju.
- Jaunās informācijas izmantošana: skolēni individuāli strādā pie 5. Uzdevuma.

6. praktiskā nodarbība

Iesaistīšanās fāze (30 min.)

- Jaunās informācijas izmantošana: skolēni individuāli pilda 5. uzdevumu.

Refleksijas fāze (10 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās? Vai vēlētos izmantot šo tiešsaistes resursu nākotnē? Kāpēc vai kāpēc ne?

Darba lapa „Attēlu rediģēšana ar Pixlr”

1. uzdevums. Pārveido krāsainu attēlu melnbaltā, saglabājot krāsu efektu (jebkurš attēls) Piemēram



- Atrodiet un lejupielādējiet 4 attēlus no bezmaksas attēlu krātuves, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.
- Atveriet Pixlr redaktoru, pēc tam izvēlieties „Open Image“, lai atvērtu attēlu. Slāņu panelī atbloķējiet fona slāni (noklikšķiniet uz atslēgas ikonas) un to dublējiet (Cmd+D vai Ctrl+D).
- Koriģējiet dublikātu slāni (Adjustment/ Desaturate) un izmantojiet dzēšgumiju (Softness 0%), lai atgrieztu krāsu vairākām jūsu izvēlētajām figūrām.
- Krāsainajam slānim ir jāmaina krāsu piesātinājums: Adjustment/ Hue and saturation, mainot Saturation: 25-30.

2. uzdevums. Vecās fotogrāfijas (jebkura attēla) izveide

Piemēram



- Atrodiet un lejupielādējiet jebkurus 4 attēlus no bezmaksas krājuma attēlu platformām, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.

- Atveriet Pixlr redaktoru, pēc tam izvēlieties Open Image, lai atvērtu šo attēlu. Slāņu panelī (Layers) atslēdziet fona slāni (uzklikšķiniet uz atslēgas ikonas) un dublējiet to (Cmd+D vai Ctrl+D).
- Lai padarītu attēlu dzeltenāku, nepieciešams veikt šādas izmaiņas: Adjustments/ Hue and saturation, jāatzīmē izvēle Colorize, jāmaina Hue: -140, Saturation: -50, Lightness: 0.

Piemēram



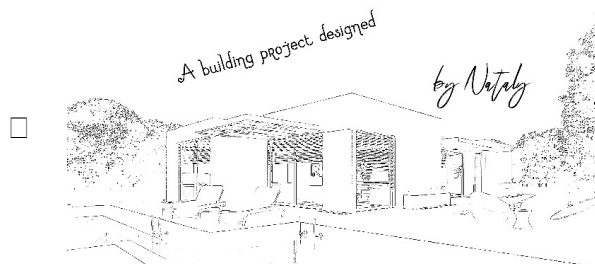
- Slānim Background Copy lietojiet filtru Effect library/ Retro/ Ragwarm. Pēc tam lietojiet filtru Details/ Add Noise, mainot iestatījumu Amount: 20.
- Izmantojot rīku Marquee Select, iezīmējiet attēla kreiso pusi un nospiediet Delete. Lai pārtrauktu iezīmētā apgabala mirgošanu, nospiediet Cmd+D vai Ctrl+D.

3. uzdevums. Zīmējums pēc fotogrāfijas (jebkura fotogrāfija ar labu melnbaltā kontrastu)

Piemēram



sākumā



pabeigts

- Atrodiet un lejupielādējiet jebkuru attēlu no bezmaksas krājuma attēlu platformām, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.
- Dublējiet fona slāni (Duplicate the Background layer), pārvērtiet krāsaino attēlu melnbaltā (izvēlnē Adjustments/Desaturate).
- Palaidiet Adjustments/Levels un mainiet šādus iestatījumus:



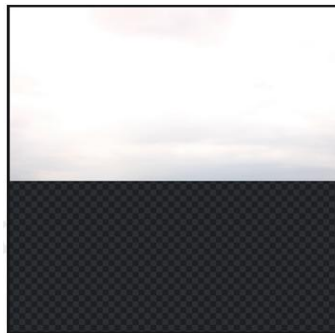
- Dublējiet fona slāni (Duplicate the Background layer) un pārvietojiet šo dublikātu virs visiem esošajiem slāņiem.
- Lietojiet filtru Find Edges, pēc tam veiciet inversiju: Adjustments/Invert.
- Lietojiet filtru Effect library/ Urban/ Sham. Nepieciešams uzlabot krāsu piesātinājumu, kontrastu un gaišumu: Adjustments/Hue & Saturation, mainot Hue: 0, Saturation: 0, Lightness: +15; pēc tam Brightness & Contrast, mainot Brightness: 0, Contrast: +50.
- Nomainiet slāņu sajaukšanas režīmu (blend mode) uz Luminosity.

4. uzdevums. Dzīves grāmatas veidošana

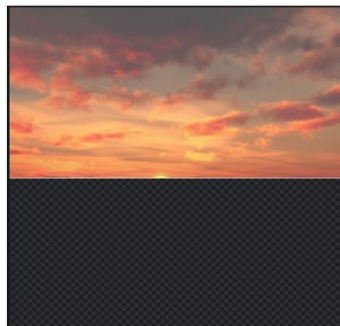
Piemēram



- Atveriet jaunu failu ar šādiem parametriem: Width: 3000 px, Height: 3000 px, Background: none. Atveriet attēlu D&Z (Full HD), atlasiet tikai debesis.
- Pēc atlasē nokopējiet atlasīto fragmentu un pārvietojiet to uz galveno failu (Untitled), mainiet izmērus. Pārdēvējiet slāni par Sky.



- Atveriet attēlu Sunset, atlasiet saulrieta augšējo daļu. Pēc atlasē nokopējiet atlasīto fragmentu un pārvietojiet to uz galveno failu (Untitled), mainiet izmērus. Pārdēvējiet slāni par Sunset. Mainiet caurredzamību (opacity) uz 65%.



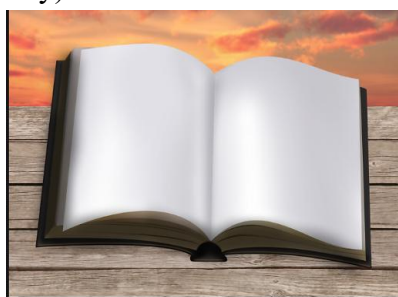
- Atveriet attēlu „Table“, atlasiet visu un pārvietojiet to uz galveno failu („Untitled“), vajadzības gadījumā mainot izmērus. Pārdēvējiet slāni par „Table“.



- Atveriet attēlu „Grāmata“ (Full HD), atlasiet visu saturu un pārvietojiet to uz galveno failu (Untitled), mainiet izmērus un, ja nepieciešams, pagrieziet. Mainiet slāņa nosaukumu uz „Book“.
- Grāmatai nepieciešama ēna. Izveidojiet jaunu slāni (Layer/New layer) un pārdēvējiet to par Shadow. Izvēlieties otu un iestatiet šādus parametrus: Size: 400 px, Softness: 100%, Step: 10%. Mainiet caurredzamību (Opacity) uz 65%.



uzzīmēts ar otu



piemērota necaurspīdība

- Atveriet attēlu Road (Full HD) un izmantojiet Lasso rīku, lai atlasītu tikai ceļu. Pēc atlases nokopējiet atlasīto fragmentu un pārvietojiet to uz galveno failu (Untitled), ja nepieciešams, mainot izmērus.
- Pārdēvējiet slāni par Road. Izvēlieties dzēšgumijas rīku (Softness: 76%, Step: 0%) un izdzēsiet ceļa malas pa perimetru, lai tas izskatītos dabiskāk.



- Atveriet attēlu Fox, izmantojiet Select/Subject, lai atlasītu tikai lapsu. Pēc atlases nokopējiet atlasīto fragmentu un pārvietojiet to uz galveno failu (Untitled), ja nepieciešams, mainot izmērus.
- Pārdēvējiet slāni par Fox. Uzrakstiet savu vārdu grāmatas kreisajā pusē. Pārvērtiet ceļu melnbaltu un izmantojiet bezmaksas krājuma attēlu platformu, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>, lai ievietotu jebkuru objektu.

5. uzdevums. GIF animācijas izveide (no jebkuriem vairākiem secīgiem attēliem)

- Atveriet jaunu failu. Izvēlieties opciju teksta ievadīšanai un ievadiet animācijas nosaukumu. Izmantojot rīku „Snipping Tool“, saglabāiet to ar nosaukumu „1-frame“.
- Atveriet pirmo attēlu („File“/„Open Image“), atlasiet to („Marquee Select“) un kopējiet, pēc tam ielīmējiet fona slānī. Pievienojiet paskaidrojošu tekstu zem šī attēla.

Piemēram



- Atveriet otro attēlu un atkārtojiet iepriekšējo darbību. To izdariat ar visām savām fotogrāfijām.
- Atveriet animāciju veidotāju <https://ezgif.com/>
- Izvēlieties „GIF Maker“ un augšupielādējiet visus sagatavotos kadrus. Nospiediet pogu „Upload files!“. Ja jūsu kadri nav vienāda izmēra, jums ir jāpiekrīt, ka visi kadri tiks automātiski apgriezti/pārveidoti, lai tie atbilstu mazākajam.
- Iestatiet kadru aizkaves laiku animācijai: 700. Nospiediet pogu Make a GIF!
- Jūs varat piemērot efektus jaunizveidotajai GIF animācijai.
- Izmantojot izveidoto GIF animāciju, jums vajadzētu izveidot MP4 video. Nospiediet Video to GIF, pēc tam augšupielādējiet GIF animāciju un augšupielādējiet video.

2. modulis. 3D modelēšana ar 3D drukāšanu

Mērķis: Attīstīt telpisko domāšanu, radošumu un digitālās prasmes satura veidošanā. Apgūt, kā izveidot modeli bezmaksas tiešsaistes programmā „SketchUp“, kā veidot animācijas no vairākām ainām un kā izdrukāt izveidoto modeli ar 3D printeri.

Pārskats:

2.1. modulis. Atslēgu piekariņš ar savu vārdu

2.2. modulis. Vāzes izveide

2.3. modulis. Uzglabāšanas kastes izveide

2.4. modulis. 3D mājas izveide ar animāciju

Laiks/stundas: Kopā 18 stundas.

4 stundas 2.1. modulim.

2 stundas 2.2. modulim.

4 stundas 2.3. modulim.

8 stundas 2.4. modulim.

2.1. modulis. Atslēgu piekariņš ar savu vārdu

Mācību mērķis:

- Līdz nodarbības beigām skolēni spēs izveidot atslēgu gredzena (atslēgu piekariņa) modeli tiešsaistes programmā „SketchUp“ un izdrukāt atslēgu gredzenu (atslēgu piekariņu) ar 3D printeri.

Materiāli:

- projektoris;
- ierīces ar interneta pieslēgumu (portatīvie datori/planšetdatori) tiešsaistes resursa SketchUp izmantošanai;
- praktisks uzdevums „Atslēgu gredzens (atslēgu piekariņš) ar savu vārdu”;
- 3D printeris.

Mācību metodes:

- demonstrācija;
- praktisks uzdevums;
- problēmu risināšana;
- mācīšanās, diskutējot.

Avoti: <https://sketchup.trimble.com/en/try-sketchup>

1. praktiskā nodarbība

Ievada posms (10 min.)

- Uzmanības pievēršana: skolotājs parāda dažādus modeļus, kas izdrukāti ar 3D printeri.
- Par mērķi un sasniedzamajiem rezultātiem: skolotājs mudina skolēnus formulēt moduļa mērķi.

Iesaistīšanās posms (30 min.)

- Jaunas informācijas apguve: skolotājs parāda skolēniem, kā pieslēgties tiešsaistes resursam SketchUp. Izskaidro, kā pieslēgties bezmaksas versijai, izmantojot e-pasta, Google, Apple vai Microsoft kontu. Skolotājs sāk iepazīstināt skolēnus ar tiešsaistes resursa SketchUp darba rīkiem (Line, Rectangle, Push/Pull, Rotate, Orbit, Pan, Paint, Scale) un lenti Model Info. Informācijas apkopošana un apspriešana ar skolēniem.

2. praktiskā nodarbība

Ievadfāze (25 min.)

- Jaunas informācijas apguve: skolotājs parāda skolēniem, kā pieslēgties tiešsaistes resursam „SketchUp“. Izskaidro, kā pieslēgties bezmaksas versijai, izmantojot e-pasta, Google, Apple vai Microsoft kontu. Skolotājs turpina iepazīstināt skolēnus ar tiešsaistes resursa „SketchUp“ darba rīkiem (mērlente, pārvietošana, divpunktu loks, izmēri, 3D teksts) un lenti „Display“.
- Jaunās informācijas izmantošana: skolēni patstāvīgi iepazīst tiešsaistes resursu SketchUp.

Pārdomu posms (15 min.)

- Atsauksmes: Ko jūs šodien uzzinājāt par tiešsaistes rīku SketchUp? Kas izdevās? Kas neizdevās? Ko vēl plānojat apgūt?

3. praktiskā nodarbība

Iesaistīšanās posms (40 min.)

- Iepriekšējo zināšanu atkārtošana par dažādiem SketchUp rīkiem un joslām. Diskusija ar skolēniem par pārvietošanas rīkiem. Kā ar tiem strādāt? Kā pareizi norādīt modeļa izmērus? Kā darbojas rīki „Push/Pull“ un „2 Point Arc“?
- Jaunas informācijas apguve: skolotājs parāda skolēniem, kā veidot noapaļotus stūrus un telpisko tekstu. Jaunās informācijas pielietošana: skolēni individuāli strādā pie uzdevuma „Atslēgu gredzens ar savu vārdu“.

4. praktiskā nodarbība

Iesaistīšanās posms (30 min.)

- Jaunās informācijas pielietošana: skolēni turpina individuāli strādāt pie uzdevuma „Atslēgu piekariņš ar savu vārdu” un izdrukā atslēgu piekariņu ar 3D printeri.

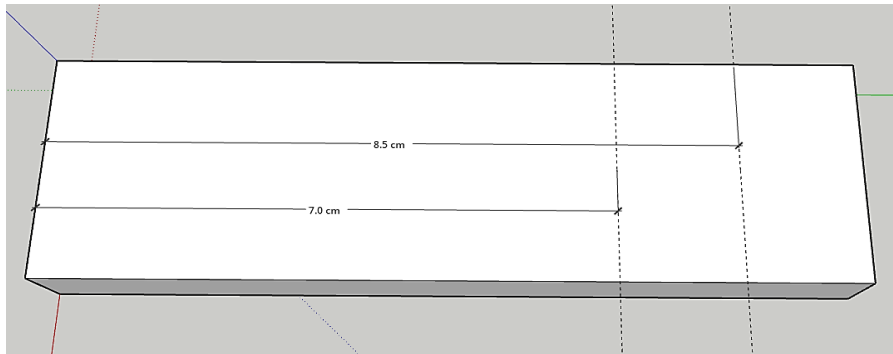
Refleksijas posms (10 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās? Vai bija interesanti izveidot 3D modeli un to izdrukāt? Kāpēc vai kāpēc ne?

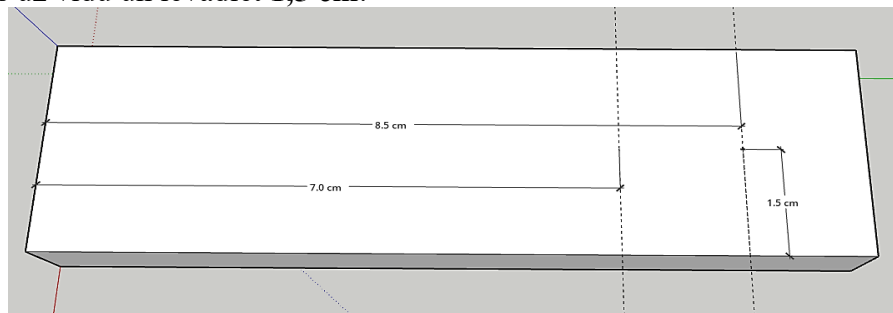
Darba lapa „Atslēgu gredzens (atsēgu piekariņš) ar savu vārdu”

- Atveriet programmu SketchUp.

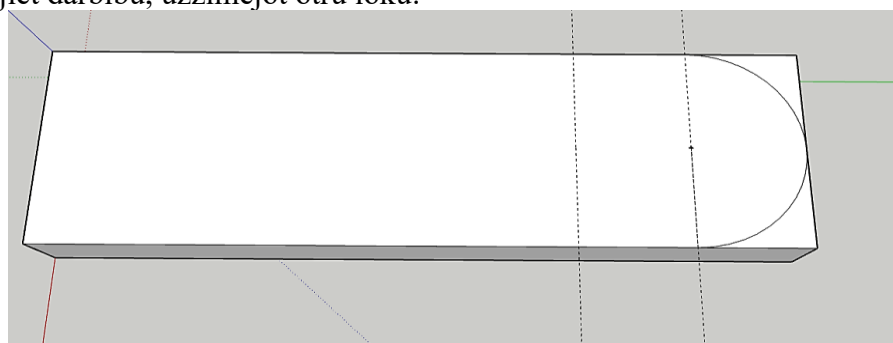
- Sadaļā „Model Info“ mainiet formātu uz „Centimeter“, attēlošanas precizitāti uz „0,0 cm“ un garuma pieskaņošanu uz „0,1 cm“.
- Izvēlieties rīku „Rectangle“, noklikšķiniet uz sākuma punkta un ievadiet: 3, 10 cm.
- Izvēlieties rīku „Push/Pull“ un paceliet modeli 1 cm augstumā.
- Izvēlieties rīku „Tape Measure“, noklikšķiniet uz viduspunktu, pārvietojiet peli no kreisās puses uz modeļa centru un ievadiet: 7 cm.
- Atkārtoti izvēlieties rīku „Tape Measure“, noklikšķiniet uz viduspunktu, pārvietojiet peli no kreisās puses uz modeļa centru un ievadiet 8,5 cm:



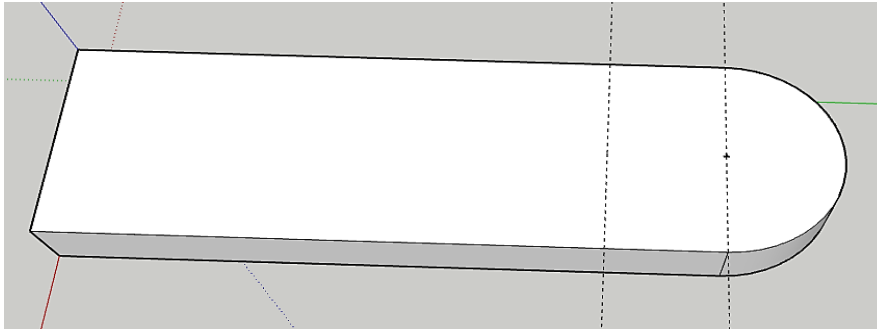
- Atkārtoti izvēlieties rīku **Tape Measure**, noklikšķiniet un pārvietojiet peli no modeļa apakšas uz vidu un ievadiet **1,5 cm**:



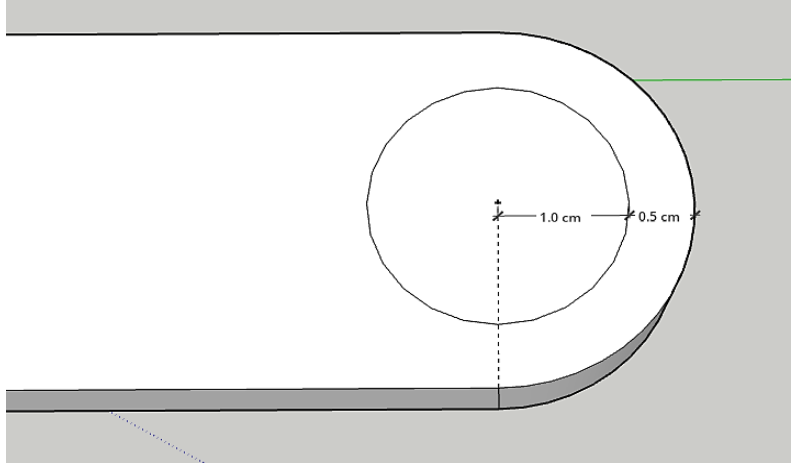
- Atkārtoti izvēlieties rīku **2 Point Arc**, uzzīmējiet loku līdz modeļa viduspunktam. Tad atkārtojiet darbību, uzzīmējot otru loku:



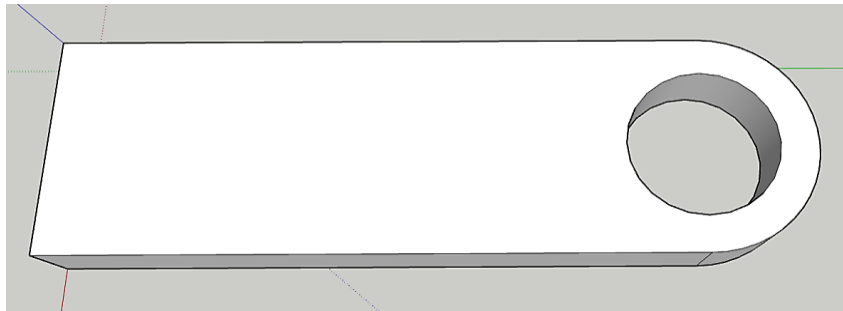
- Stumiet nevajadzīgo materiālu ar rīku **Push/Pull**:



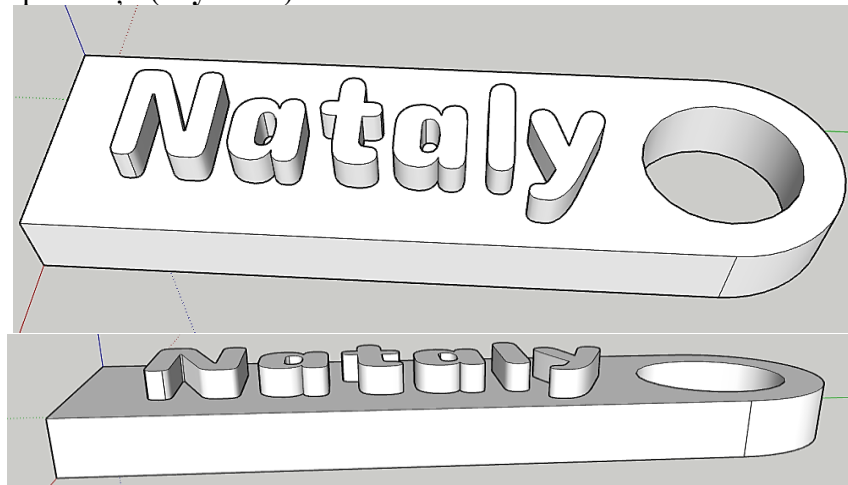
- Izmantojot rīku **Circle** un punktu uz modeļa, uzzīmējiet apli ar rādiusu **1 cm**:



- Izstumiet caurumu ar rīku **Push/Pull**:



- Izvēlieties rīku **3D Text**, ierakstiet savu vārdu un mainiet parametrus, piemēram: fonts **Concert One**, **Height: 1.5 cm**, **Text extrusion: 0.5 cm**. Pēc tam ievietojiet savu 3D vārdu uz atslēgu piekariņa (**keychain**):



- Saglabājiet savu darbu kā SketchUp failu (**Download/ SKP**).

- Saglabājiet savu darbu kā 3D modeli drukāšanai ar 3D printeri (**Download/ STL**).
- Izdrukājiet atslēgu piekariņu (**keychain**) ar 3D printeri.

2.2. modulis. Vāzes izgatavošana

Mācību mērķis:

Līdz nodarbības beigām skolēni spēs izveidot vāzes modeli tiešsaistes programmā „SketchUp” un izdrukāt to ar 3D printeri.

Nepieciešamais aprīkojums:

- projektors;
- ierīces ar interneta pieslēgumu (portatīvie datori/planšetdatori) tiešsaistes resursa „SketchUp” izmantošanai;
- praktiskais uzdevums „Vāzes izveide”;
- 3D printeris.

Mācību metodes:

- demonstrācija;
- praktisks uzdevums;
- problēmu risināšana;
- mācīšanās, diskutējot.

Avoti: <https://sketchup.trimble.com/en/try-sketchup>

1. praktiskā nodarbība

Ievada posms (10 min.)

- Iepriekšējo zināšanu atkārtošana par dažādiem SketchUp rīkiem. Diskusija ar studentiem par rīku pārvietošanu. Kur var atrast papildu rīkus?

Ievadposms (30 min.)

- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursu „SketchUp” darba rīkiem (līkne, aplis, brīvroka, nobīde, krāsošana) un joslu „Materiāli”.
- Jaunās informācijas pielietošana: skolēni individuāli strādā pie praktiskā uzdevuma „Vāzes izveide”.

2. praktiskā nodarbība

Ievadposms (25 min.)

- Studenti individuāli pabeidza praktisko darbu „Vāzes veidošana”.

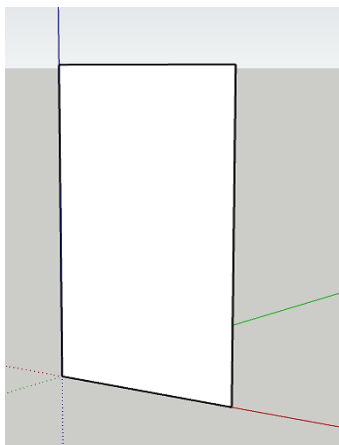
Pārdomu posms (15 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās? Vai vēlētos izmantot šo tiešsaistes resursu arī turpmāk? Kāpēc vai kāpēc ne?

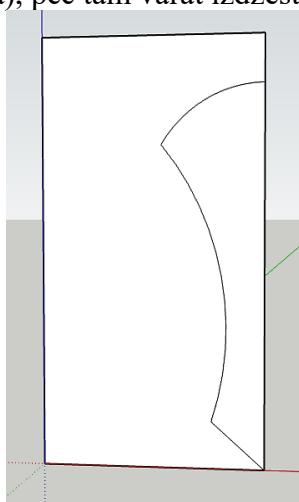
Darba lapa „Vāzes veidošana”

- Atveriet programmu SketchUp.
- Modeli informācijā mainiet formātu uz centimetriem, attēlošanas precizitāti uz 0,0 cm un garuma pieskaņošanu uz 0,1 cm.

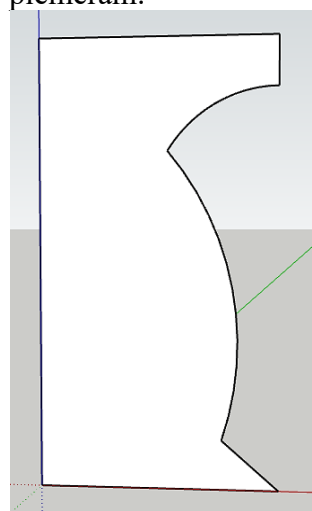
- Izvēlieties taisnstūra rīku, noklikšķiniet uz sākuma punkta un ievadiet, piemēram, 10, 18 cm.



- Izmantojot līknes un līnijas rīkus, uzzīmējiet vāzes kontūru (pēc saviem ieskatiem/brīvā stilā), pēc tam varat izdzēst liekās taisnstūra daļas, piemēram:

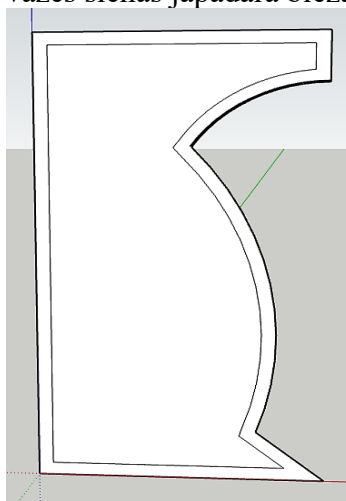


Vāzes profils

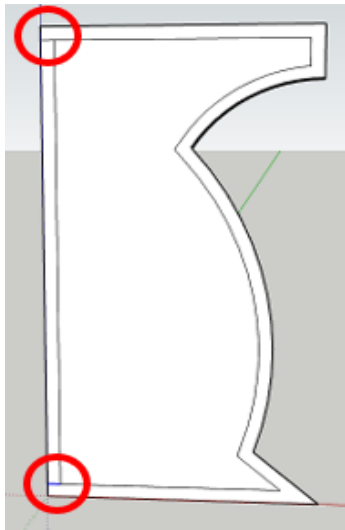


Dzēst nevajadzīgās daļas

- Izmantojot rīku „Offset”, vāzes sienas jāpadara biezākas: 0,5 cm

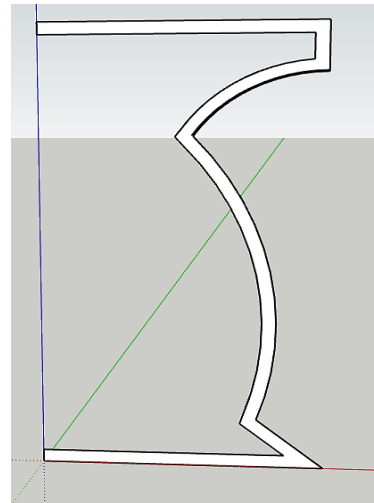


- Jums ir jāizveido kontūra kontūrošanas rīkam, tāpēc ir jāuzvelk divas papildu līnijas. Pēc tam varat izdzēst nevajadzīgās vāzes daļas.



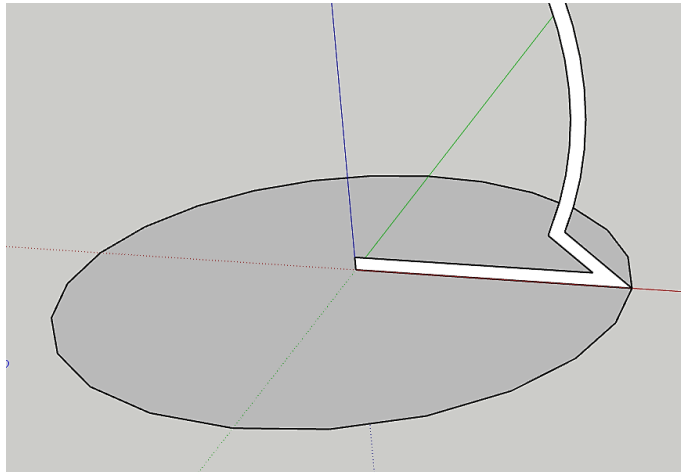
Papildu rindas

□

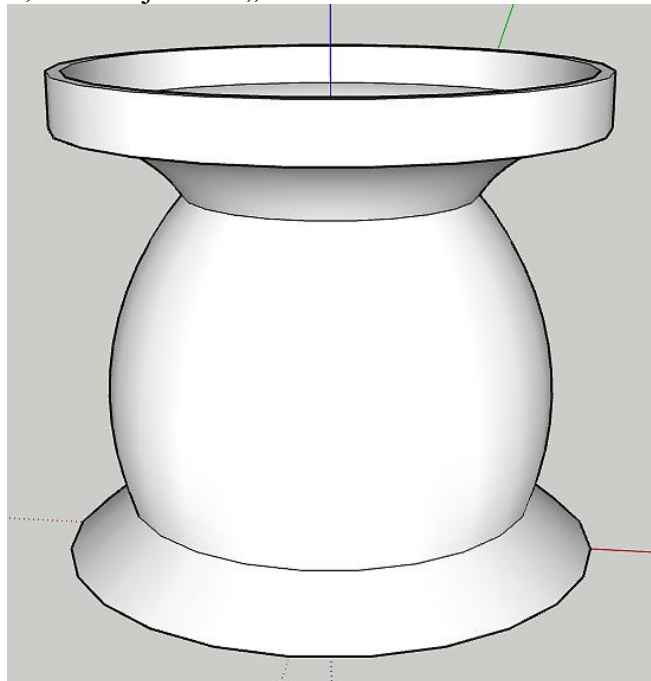


Dzēst nevajadzīgās daļas

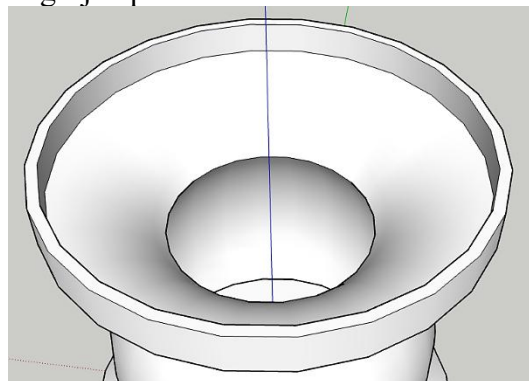
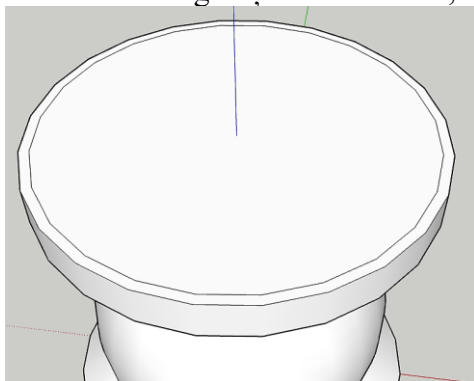
- Izmanto apla rīku, lai uzzīmētu apli. Pārliecinies, ka aplis ir uzzīmēts pareizajā plaknē:



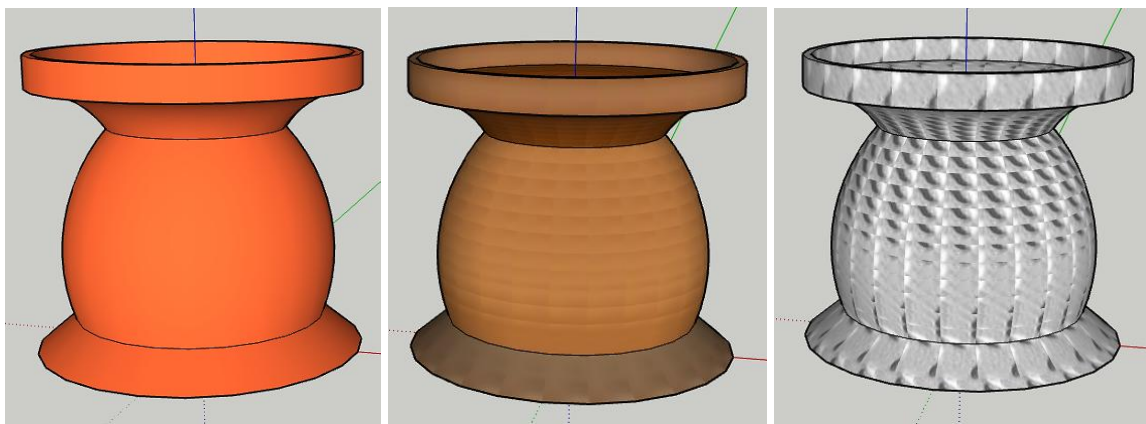
- Izveidojiet vāzi, izmantojot rīku „Follow me”:



- Ja vāzes augšdaļā nav cauruma, atzīmējiet augšējo apli un izdzēsiet to:



- Izmantojot izvēlni Materials/ Browse, piešķiriet vāzei jebkuru materiālu pēc savas izvēles, piemēram:



- Saglabājiet savu darbu kā SketchUp failu (Lejupielādēt/ SKP).
- Saglabājiet savu darbu kā 3D modeli, lai to izdrukātu ar 3D printeri (Lejupielādēt/ STL).
- Izdrukājiet vāzi ar 3D printeri.

2.3. modulis. Uzglabāšanas kastes izveide

Mācību mērķis:

Līdz nodarbības beigām skolēni spēs izveidot uzglabāšanas kastes modeli tiešsaistes programmā „SketchUp“ un izdrukāt to ar 3D printeri.

Nepieciešamais aprīkojums:

- projektoris;
- ierīces ar interneta pieslēgumu (portatīvie datori/planšetdatori) tiešsaistes resursa „SketchUp“ izmantošanai;
- praktiskais uzdevums „Uzglabāšanas kastes izgatavošana“;
- 3D printeris.

Mācību metodes:

- demonstrācija;
- praktisks uzdevums;
- problēmu risināšana;
- mācīšanās, apspriežot.

Avoti: <https://sketchup.trimble.com/en/try-sketchup>

1.–2. praktiskā nodarbība

Ievada posms (10 min.)

- Iepriekšējo zināšanu atkārtošana par dažādiem SketchUp rīkiem. Diskusija ar studentiem par rīku pārvietošanu. Kur var atrast papildu rīkus?

Iesaistīšanās posms (30 min.)

- Jaunas informācijas apguve: skolotājs iepazīstina skolēnus ar tiešsaistes resursu „SketchUp“ darba rīkiem (2 Point Arc, Move) un iespēju dublēt objektu.
- Jaunās informācijas pielietošana: skolēni individuāli strādā pie uzdevuma „Uzglabāšanas kastes izveide” kā praktiskā darba ietvaros.

3.–4. praktiskā nodarbība

Ievadposms (25 min.)

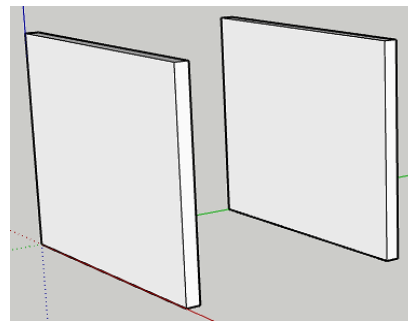
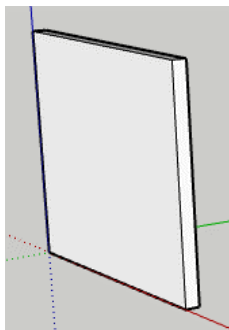
- Jaunās informācijas pielietošana: skolēni individuāli pabeidza praktisko darbu „Uzglabāšanas kastes izgatavošana”.

Pārdomu posms (15 min.)

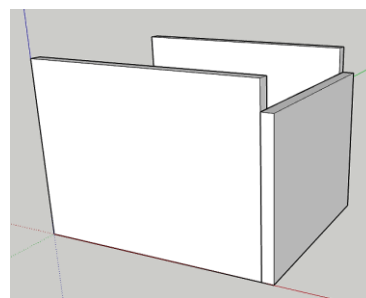
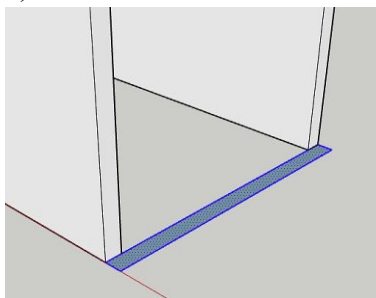
- Atsauksmes: Kas izdevās? Kas neizdevās? Vai vēlētos izmantot šo tiešsaistes resursu nākotnē? Kāpēc vai kāpēc ne?

Darba lapa „Uzglabāšanas kastes izgatavošana“

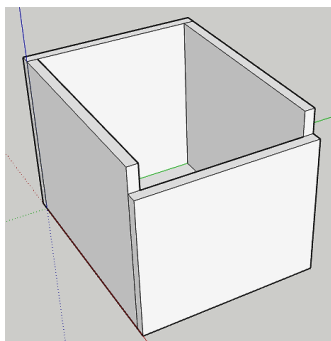
- Atveriet programmu SketchUp.
- Mainiet modeļa informācijas formātu: centimetri, attēlošanas precizitāte: 0,0 mm, garuma pieskaņošana: 0,1 mm.
- Izvēlieties taisnstūra rīku un ievadiet, piemēram, 500, 25 mm.
- Izvēlieties rīku „Push/Pull” un paceliet modeli līdz 400 mm augstumam. Duplicējiet šo objektu pa labi (Ctrl + Move) 400 mm attālumā:



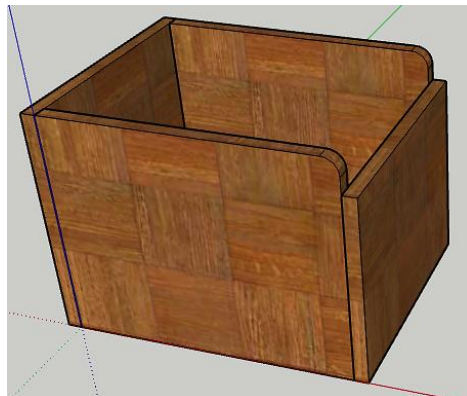
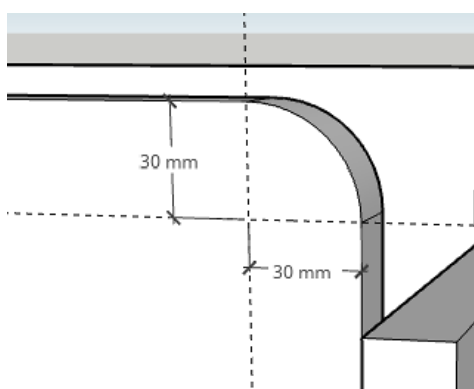
- Savienojiet abus iezīmētos objektus, uz zilas plaknes uzzīmējot taisnstūri (425 × 25 mm) un izvirzot modeli 340 mm augstumā:



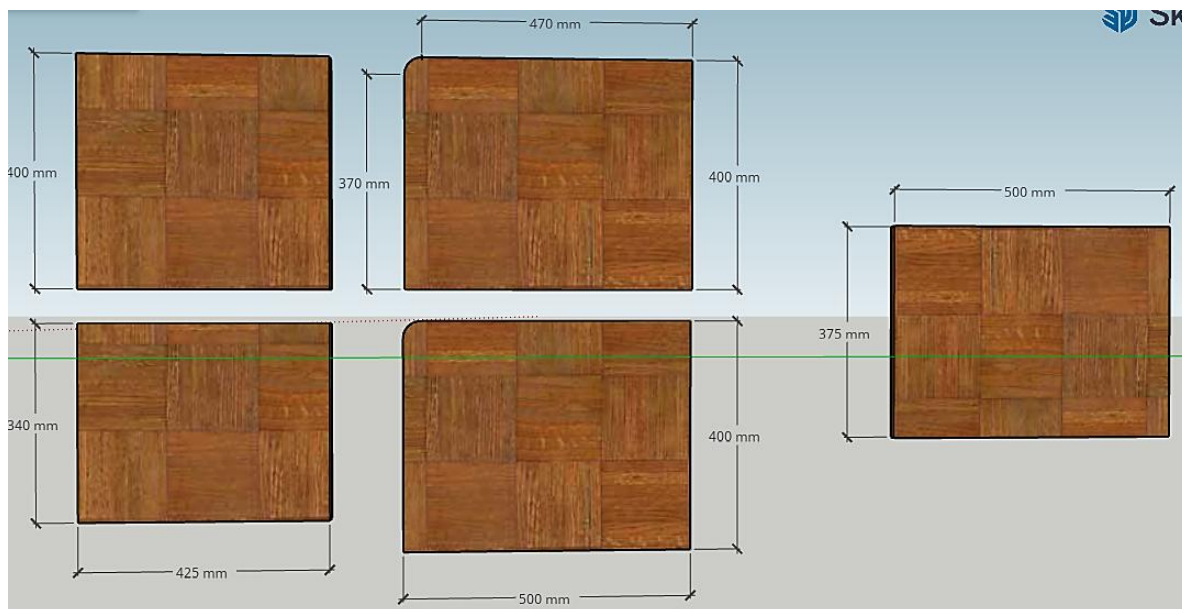
- Atkārtojiet šo darbību, uzzīmējot uzglabāšanas kastes sienu no otras puses (augstums 400 mm):



- Izmantojot rīku „2 punktu loks” (2 Point Arc), noapaļojiet stūrus ar rādiusu **30 mm** un noņemiet lieko daļu. Izmantojot paneli „Materiāli/Pārlūkot” (Materials/Browse), uzklājiet uz uzglabāšanas kastes jebkuru materiālu pēc savas izvēles, piemēram:



- Izveidojiet uzglabāšanas kastes rasējumu ar visām detaļām un nepieciešamajiem izmēriem:



2.4. modulis. 3D mājas izveide ar animāciju

Mācību mērķis:

Līdz moduļa beigām studenti spēs izveidot 3D māju tiešsaistes programmā SketchUp un izveidot video animāciju.

Nepieciešamais aprīkojums:

- projektors;
- ierīces ar interneta pieslēgumu (portatīvie datori/planšetdatori) tiešsaistes resursa „SketchUp“ izmantošanai;
- praktiskais uzdevums „3D mājas izveide ar animāciju“.

Mācību metodes:

- demonstrācija;
- praktisks uzdevums;
- problēmu risināšana;
- mācīšanās, apspriežot

Avoti: <https://sketchup.trimble.com/en/try-sketchup>

1.–4. praktiskā nodarbība

Ievadfāze (10 min.)

- Iepriekšējo zināšanu atkārtošana par dažādiem SketchUp rīkiem. Diskusija ar studentiem par dažādiem rīkiem un joslām. Kur var atrast papildu rīkus?

Iesaistīšanās posms (25 min.)

- Jaunās informācijas pielietošana: studenti individuāli strādā pie praktiskā uzdevuma „3D mājas izveide ar animāciju“.

Refleksijas fāze (5 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās?

5.–8. praktiskā nodarbība

Iesaistīšanās fāze (35 min.)

- Iepriekšējo zināšanu atjaunošana par dažādiem SketchUp rīkiem. Diskusija ar studentiem par dažādiem rīkiem un joslām. Kā varu izveidot ainas? Kā varu lejupielādēt vairākus objektus no 3D krātuves?
- Jaunās informācijas izmantošana: studenti turpina individuāli strādāt pie uzdevuma „3D mājas izveide ar animāciju” – praktiskais darbs.

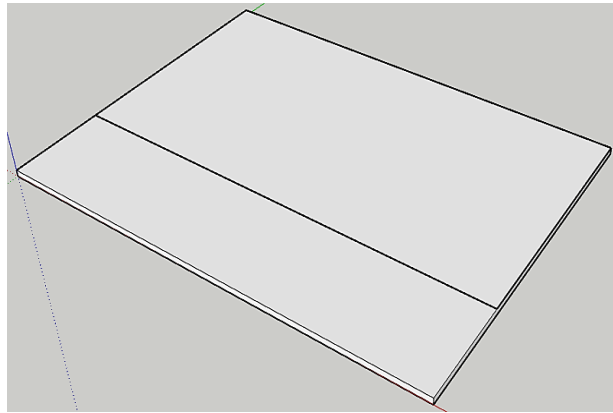
Refleksijas fāze (5 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās?

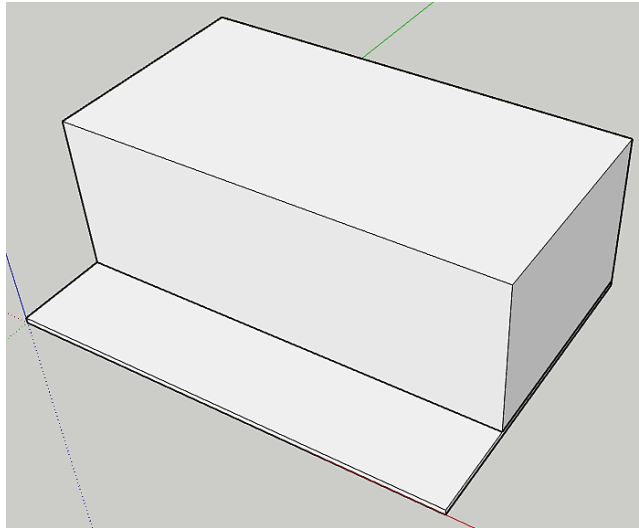
Darba lapa „3D mājas izveide ar animāciju”

- Atveriet programmu SketchUp.
- Sadaļā „Model Info” mainiet formātu uz „Millimeter”, attēlošanas precizitāti uz „0,0 mm” un garuma pieskaņošanu uz „0,1 mm”.
- Izvēlieties rīku „Rectangle” un ievadiet, piemēram, 8400, 6100 mm.
- Izvēlieties rīku „Push/Pull” un paceliet modeli līdz 110 mm augstumam.

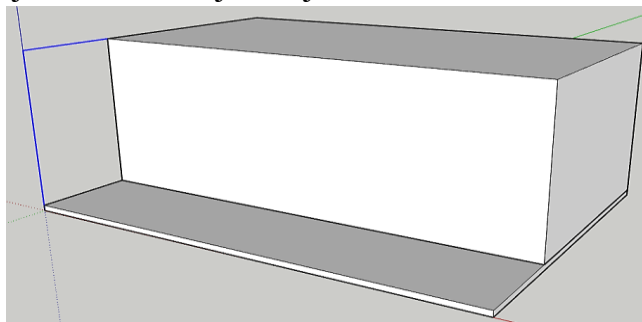
- Izveidotais modelis ir jāgrupē. Izceliet modeli, tad noklikšķiniet ar peles labo pogu un izvēlieties „Make Group”.
- Izvēlieties rīku „Rectangle” un ievadiet, piemēram, 4300, 8400 mm:



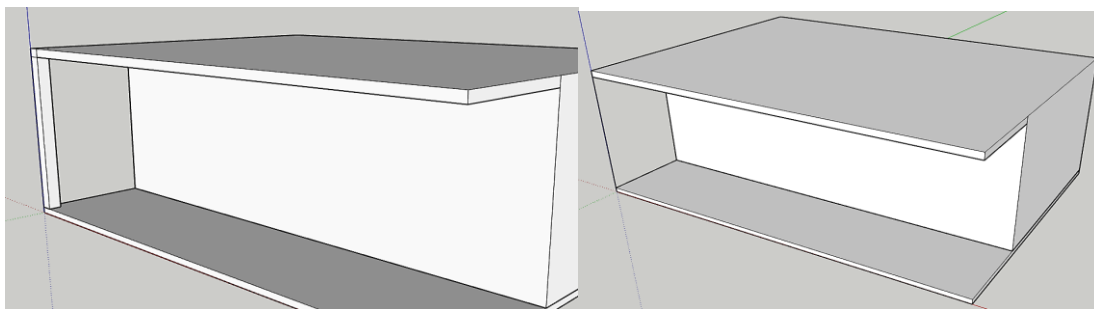
- Izvēlieties rīku „Push/Pull” un paceliet modeli līdz 3000 mm augstumam.



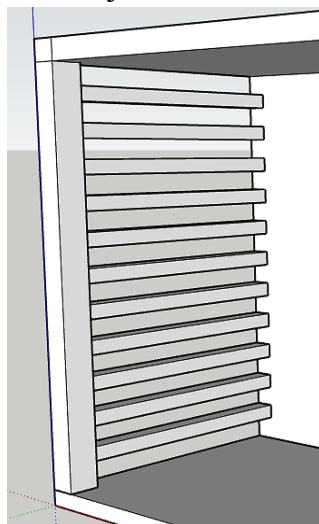
- Izmantojot līnijas rīku, uzzīmējiet mājas sānu malu:



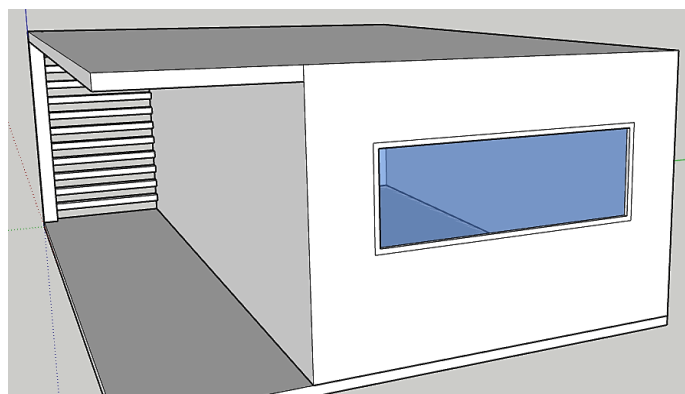
- Izmantojot rīku „Taisnstūris”, uzzīmējiet nojumes jumtu. Pēc tam varat dzēst liekās līnijas uz jumta. Izmantojot rīku „Push/Pull”, paceliet nojumes jumtu līdz 150 mm augstumam:



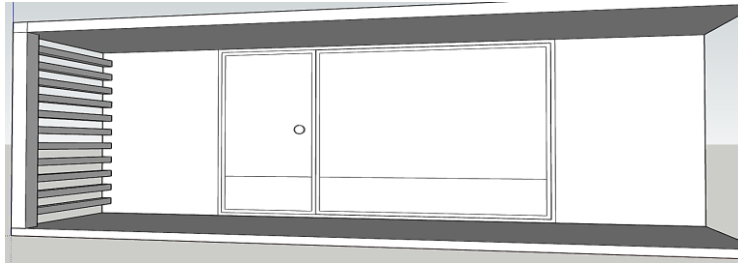
- Izmantojot taisnstūra rīku (200 mm, 200 mm) un stumšanas/velkšanas rīku, izveidojiet stabu:
- Dublikējot izveidoto stabu, izveidojiet sānu sienu:



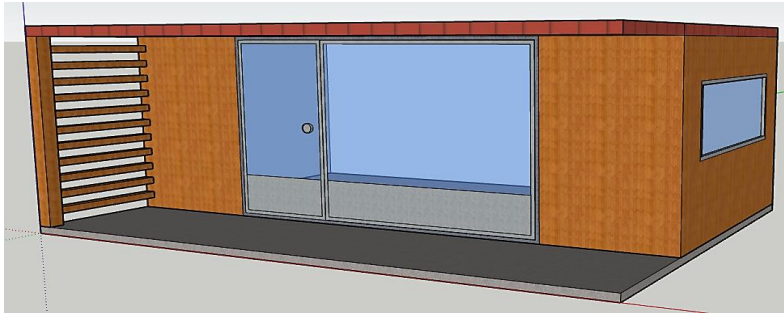
- Izmantojot rīkus **Rectangle**, **Offset** un **Push/Pull**, izveidojiet logu. Pēc tam, izmantojot paneli **Materials / Browse / Glass and Mirrors**, uzklājiet izvēlēto loga stikla materiālu, piemēram:



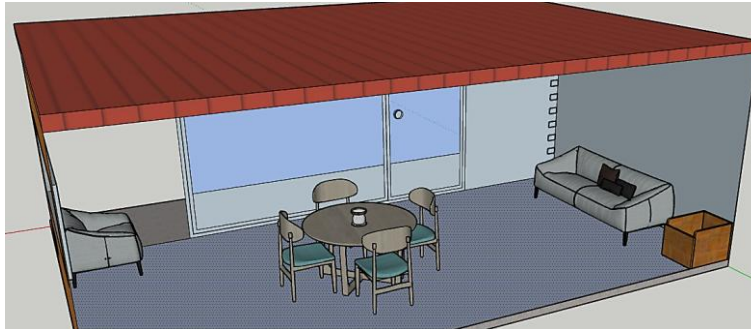
- Izmantojot taisnstūra, nobīdes, apla un līnijas rīkus, izveidojiet durvis un panorāmas logu:



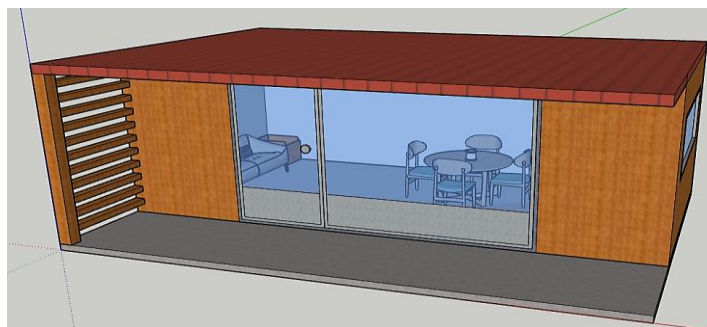
- Izmantojot lenti Material/Browse, piemērojiet mājai jebkuru vēlamo materiālu, piemēram:



- Izmantojot 3D Warehouse, izvēlieties mēbeles, traukus un citus priekšmetus, kas jums patīk, un lejupielādējiet tos. Novietojiet mājā savu izveidoto vāzi un uzglabāšanas kasti.



- Izvēlieties paneli **Scenes** un izveidojiet klipus (**Add scene**), pagriežot mājas modeli jebkurā veidā, kā vēlaties.



3. modulis. Video veidošana pārlūkprogrammā

Mērķis: Attīstīt telpisko domāšanu, radošumu un digitālās prasmes satura veidošanā. Apgūt, kā izveidot video bezmaksas tiešsaistes programmā „Magisto“.

Pārskats:

3.1. modulis. Video veidošana ar „Magisto“

Ilgums: 4 stundas.

3.1. modulis. Video veidošana ar Magisto

Mācību mērķis:

Līdz 3.1. moduļa beigām studenti spēs saprast, kā izveidot video bezmaksas tiešsaistes programmā „Magisto“.

Materiāli:

- Darba lapa „Video veidošana ar Magisto”.
- Ierīces ar interneta pieslēgumu (portatīvie datori/planšetdatori) tiešsaistes Magisto lietošanai.

Mācību metodes:

- demonstrācija;
- problēmu risināšana;
- praktisks uzdevums;
- mācīšanās, apspriežot.

Avoti: <https://www.magisto.com/online-video-marketing>

1. praktiskā nodarbība

Ievada posms (10 min.)

- Uzmanības pievēršana: skolotājs parāda dažādus videoklipus.
- Par mērķi un sasniedzamajiem rezultātiem: skolotājs mudina skolēnus formulēt nodarbības mērķi.

Iesaistīšanās posms (30 min.)

- Jaunas informācijas apguve: skolotājs parāda skolēniem, kā pieslēgties tiešsaistes resursam Magisto. Izskaidro, kā pieslēgties bezmaksas versijai, izmantojot e-pastu vai Google, Facebook vai Apple kontu. Skolotājs iepazīstina skolēnus ar tiešsaistes resursu Magisto. Skolēni patstāvīgi izpēta tiešsaistes resursu Magisto. Informācijas apkopošana un apspriešana ar skolēniem.

2. praktiskā nodarbība

Iesaistīšanās fāze (25 min.)

- Jaunās informācijas izmantošana: skolēni individuāli strādā ar darba lapu “Video veidošana ar Magisto”.

Refleksijas fāze (10 min.)

- Atsauksmes: Ko jūs šodien uzzinājāt par tiešsaistes resursu Magisto? Kas izdevās? Kas neizdevās? Ko vēl plānojat apgūt?

3.–4. praktiskā nodarbība

Ievadfāze (25 min.)

- Iepriekšējo zināšanu atkārtošana par Magisto. Diskusija ar skolēniem par dažādiem videoklipu veidošanas rīkiem un mākslīgo intelektu.
- Jaunās informācijas pielietošana: skolēni individuāli turpina darbu pie praktiskā uzdevuma „Videoklipa veidošana ar Magisto”.

Refleksijas fāze (5 min.)

- Atsauksmes: Kas izdevās? Kas neizdevās?

Darba lapa „Video veidošana ar Magisto”

Augšupielādējiet savus fotoattēlus vai atrodiet un lejupielādējiet jebkuru attēlu no bezmaksas attēlu krātuves, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.

Pēc tam pievienojiet tekstu, ja vēlaties, un noklikšķiniet uz „Tālāk”.

Izvēlieties vienu no unikālajiem rediģēšanas stiliem bibliotēkā.

Logu apakšā mainiet iepriekš iestatītās krāsas.

Izvēlieties vienu no bibliotēkas skaņu celiņiem.

Noklikšķiniet uz „Priekšskatīt”, un AI balstītais profesionālais video veidotājs izveidos jūsu video.

Izveidoto video varat rediģēt vai saglabāt. Jūs varat dalīties ar savu video (to var skatīt ikviens, to var skatīt, izmantojot saiti, vai to var skatīt tikai jūs).

Jums ir jālejupielādē šis fails.

Augšupielādējiet savus foto failus vai atrodiet un lejupielādējiet jebkuru attēlu no bezmaksas attēlu meklēšanas platformas, piemēram, <http://www.unsplash.com> vai <http://www.pixabay.com>.

Tad pievienojiet tekstu.

Izvēlieties vienu no unikālajiem rediģēšanas stiliem bibliotēkā.

Logu apakšā mainiet iepriekš iestatītās krāsas.

Augšupielādējiet savu mūziku.

Noklikšķiniet uz „Priekšskatīt”. Dalieties ar savu video kā „Neiekļauts sarakstā” (to var skatīt ikviens, kam ir saite) un dalieties, nosūtot to skolotājam pa e-pastu.

Augšupielādējiet savus video failus vai atrodiet un lejupielādējiet jebkuru video no bezmaksas video krājuma platformas, piemēram, <http://www.pixabay.com>.

Tad pievienojiet tekstu.

Izvēlieties vienu no unikālajiem rediģēšanas stiliem bibliotēkā.

Logu apakšā mainiet iepriekš iestatītās krāsas.

Augšupielādējiet savu mūziku.

Noklikšķiniet uz „Priekšskatīt”. Dalieties ar savu video kā „Neiekļauts sarakstā” (to var apskatīt ikviens, kam ir saite) un nosūtiet to skolotājam pa e-pastu.

4. modulis. Ievads Python un Arduino

Mērķis: Izprast un pielietot Arduino CTC 101 sistēmas pamatus, iepazīstoties ar tās komponentiem, izveidojot programmēšanas vidi un izstrādājot pirmās programmas.

Programmas saturs:

Iepazīties ar Arduino CTC 101 komplekta detaļām un funkcijām.

Sagatavot un konfigurēt programmēšanas vidi (Processing IDE).

Iemācīties izveidot, saglabāt un pārvaldīt projektus datorā.

Rakstīt, palaist un paplašināt vienkāršas programmas.

Laiks/stundas: 10 (5 dubultstundas – 2x40 min)

4.1. modulis. Programmēšanas vides konfigurēšana

Mērķis: Ļaut studentiem iepazīties ar „Processing” un Python bibliotēku konfigurāciju, izprast galvenos saskarnes elementus un iepazīstināt ar koordinātu asu jēdzienu programmā „Processing”.

Mācību mērķis:

Līdz nodarbības beigām studenti spēs:

- Izprast Arduino projektu programmēšanas vides mērķi.
- Skolotāja vadībā instalēt programmu „Processing”.
- Pievienot „Processing” Python bibliotēku.
- Orientēties „Processing” saskarnē un atpazīt tās galvenos elementus.
- Iemācīties, kā „Processing” vidē izveidot, saglabāt un atvērt projektus.
- Izprast „Processing” koordinātu sistēmu un to, kā tā atšķiras no matemātiskās sistēmas.
- Sakārtot projektus, strādājot īpaši tam paredzētā personīgajā mapē.

Materiāli:

Datori ar interneta pieslēgumu.

Oficiālā „Processing” instalēšanas rokasgrāmata:

(<https://ctc101.arduino.cc/resources?id=31800447>)

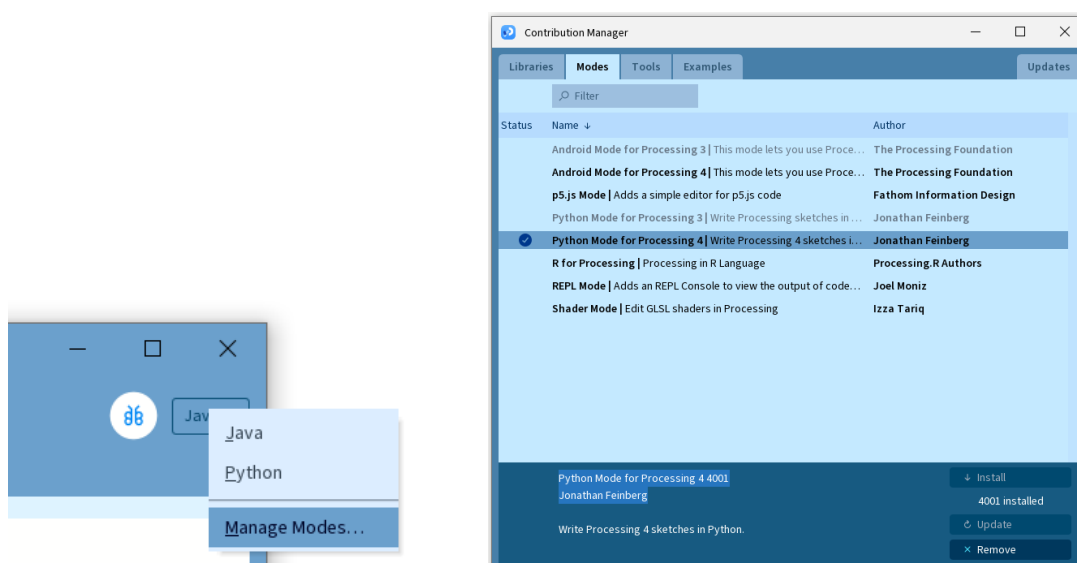
Darba lapa „Processing interfeisa identifikācija”.

Darba lapa „Processing koordinātas”.

Nodarbības plāns:

1. Izskaidro, kāpēc Arduino projektiem ir nepieciešama programmēšanas vide, uzsverot, ka tā nodrošina precīzu aparatūras komponentu vadību, ļauj skolēniem īstenot un pārbaudīt savas idejas, kā arī darbojas kā tilts starp lietotāju un Arduino fiziskajām sastāvdaļām, pārvēršot instrukcijas konkrētās darbībās. (5 minūtes)
2. Skolēni skolotāja uzraudzībā instalē Processing, sekojot oficiālajam ceļvedim. [1] (5 minūtes)

3. Skolotāja vadībā skolēni pievieno Python režīmu (Python Mode for Processing 4 4001 (Jonathan Feinberg)) programmai Processing, lai nodrošinātu programmēšanu Python valodā. (5 minūtes)



4. Katrs students (vai studentu pāri) savā datorā izveido atsevišķu mapīti, kurā tiks saglabāti visi „Processing“ projekti, tādējādi nodrošinot strukturētu darba gaitu. (5 minūtes)
5. Skolotājs parāda, kā: (5 minūtes)
 - Izveidot jaunu projektu.
 - Saglabāt to skolēna projektu mapē.
 - Atvērt jau esošu projektu.
 - Pēc tam skolēni atkārtoti šo procesu savos datoros.
6. Skolotājs izskaidro programmas „Processing“ saskarni: izvēlni, redaktoru, palaidies pogu un konsoli [2]. (10 minūtes)
7. Studenti aizpilda darba lapu „Lietotāja saskarnes elementu identificēšana”, norādot galvenās izvēlnes pozīcijas un saskarnes elementus. (10 minūtes)
8. Skolēni pāros vai grupās pārskata savas aizpildītās darba lapas, apspriež neskaidrības un skolotāja vadībā noskaidro katra saskarnes elementa mērķi un funkciju. (5 minūtes)
 - Skolotājs iepazīstina ar programmā „Processing“ [3] izmantotajām mērvienībām: (5 minūtes)
 - Pikseļi – ekrāna koordinātu pamatvienība, kas nosaka pozīcijas un izmērus.
 - Radiāni – tiek izmantoti leņķu norādīšanai.

Pēc tam skolēni apspriež, kāpēc šīs mērvienības ir svarīgas un kā tās atšķiras no ikdienā lietotajiem mērijumiem.

9. Skolotājs iepazīstina ar „Processing“ koordinātu sistēmu [3]: (10 minūtes)
 - Sākumpunkts (0,0) atrodas augšējā kreisajā stūrī, nevis apakšējā kreisajā stūrī, kā tas ir matemātikā.
 - x palielinās pa labi, bet y — uz leju.
 - Salīdziniet to ar standarta matemātiskajām asīm, kur y palielinās uz augšu.

10. Skolēni aizpilda darba lapu „Koordinātas programmā Processing”, kurā iekļauts ekrānuzņēmums ar vairākiem karikatūru stilā attēlotiem objektiem. Skolēni aprēķina un pieraksta katra objekta centra aptuvenās koordinātas (x, y), it kā tie būtu zīmēti programmā Processing. Pēc tam viņi mazās grupās salīdzina un apspriež savas atbildes, pievēršot uzmanību jebkādam atšķirībām aprēķinos. Šis uzdevums palīdz viņiem saistošā un konkrētā veidā nostiprināt izpratni par koordinātu sistēmu programmā Processing. (10 minūtes)

Pārdomas (5 minūtes):

Pirms nodarbības beigām katrs students nosauc vienu „Processing” saskarnes elementu un vienā teikumā paskaidro, kādam nolūkam tas tiek izmantots. Studentam vajadzētu censties neatkārtot citu atbildes, lai gan neliela atkārtošā var notikt, ja elementu skaits ir mazāks nekā studentu skaits.

Papildus uzdevumi:

- Papildus interfeisa elementu apzīmēšanai darba lapā „Interfeisa identifikācija”, lūdziet skolēniem ar saviem vārdiem aprakstīt katra elementa funkciju blakus tā nosaukumam.
- Skolēni var strādāt pāros, lai salīdzinātu savus aprakstus un apspriestu atšķirības izpratnē.

Avoti:

[1] <https://ctc101.arduino.cc/resources?id=31800447>

[2] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/processing>

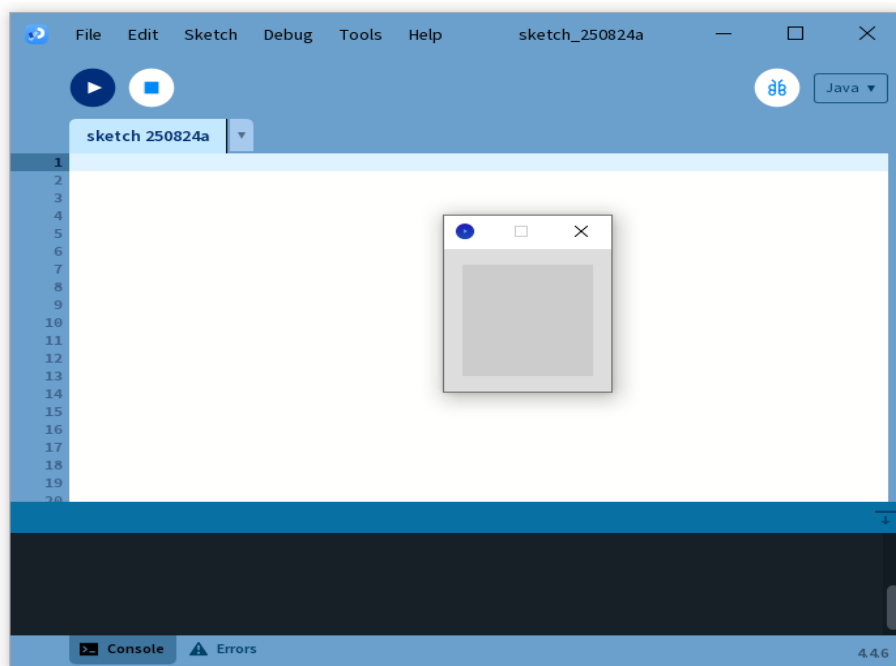
[3] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/screens-and-pixels>

Darba lapa „Processing saskarnes elementu apguve”

Iepazīstieties ar Processing saskarni. Izpētiet pievienoto Processing ekrānuzņēmumu un apzīmējiet galvenos elementus. Kad esat identificējis katru elementu, atzīmējiet to sarakstā.

Zemāk redzamajā „Processing” ekrānuzņēmumā atrodi un apzīmējiet šādus elementus:

- **Toolbar** – Rīkjosla
- **Editor Window** – Redaktora logs (koda rakstīšanas vieta)
- **Console** – Konsole (izvades logs apakšā)
- **Run Button** – Palaišanas poga (ar trijstūra simbolu)
- **Stop Button** – Apturēšanas poga (ar kvadrāta simbolu)
- **Tabs (for multiple files)** – Cilnes (vairākiem failiem)
- **Errors (Error Area)** – Kļūdu paziņojumu apgabals
- **Programming Language Selector (dropdown menu)** – Programmēšanas valodas izvēlne (parasti rāda "Java")
- **Output Window** – Rezultāta (izvades) logs (logs, kas atveras pēc "Run")
- **Project Name** – Projekta (skices) nosaukums



Darba lapa „Koordinātu apstrāde”

Svarīgākā informācija

Programmā „Processing” koordinātu sistēma atšķiras no parastās matemātiskās sistēmas.

- Sākumpunkts (0,0) atrodas izvades loga augšējā kreisajā stūrī.
- X ass stiepjas pa labi. →
- Y ass palielinās uz leju. ↓

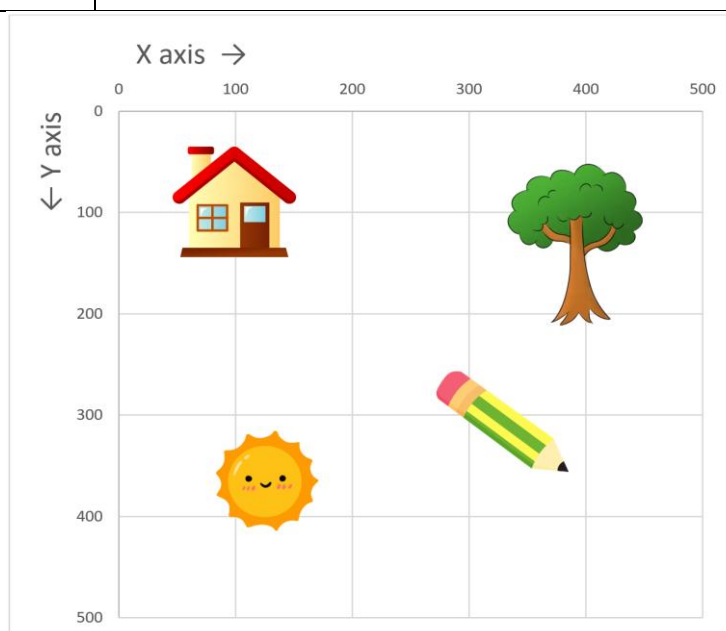
Piemērs:

- punkts(0, 0) — augšējā kreisajā stūrī.
- punkts(100, 50) — 100 pikseļus pa labi un 50 pikseļus uz leju no augšējā kreisā stūra.

Uzdevumi

Apskatiet pievienoto ekrānuzņēmumu ar vienkāršiem objektiem. Katram objektam pierakstiet tā centra aptuvenās koordinātas (x, y).

Objekts	Jūsu aptuvenās koordinātes (x, y)
Saule	
Koks	
Māja	
Zīmulis	



Kā šī koordinātu sistēma atšķiras no tās, ko esat redzējuši matemātikas stundās? (Uzrakstiet īsu atbildi.)

Ja pabeidzi ātrāk, uzzīmē vēl vienu objektu pēc savas izvēles (mākonī, automašīnu, robotu, sirdi vai jebko citu). Atzīmē uzzīmētā objekta centra koordinātas (x, y). Tad lūdz partnerim aprēķināt tava objekta centra koordinātas un salīdziniet rezultātus.

4.2. modulis. Pirmie soļi programmēšanā ar grafiku

Mērķis: Iepazīstināt skolēnus ar programmēšanu, izmantojot Python valodu vidē „Processing“, veidojot vienkāršas grafikas. Skolēni uzzinās, kā kods tiek pārvērsts vizuālā attēlā, un apgūs programmēšanas pamatjēdzienus, piemēram, zīmēšanas komandas, secību un komentārus.

Mācību mērķis:

Līdz nodarbības beigām skolēni spēs:

- Saprast, kā Python kods programmā „Processing“ veido grafikas.
- Izmantot funkcijas `size()`, `line()`, `rect()` un `ellipse()`.
- Izmantot `fill(R, G, B)`, lai noteiktu figūru pildījuma krāsu, un `stroke(R, G, B)`, lai noteiktu kontūru krāsu.
- Saprast un izmantot komentārus kodā, lai pievienotu piezīmes programmām.
- Piemērot secību, lai radītu zīmējumus un vienkāršus rakstus.

Nepieciešamais aprīkojums:

Datori, kuros ir instalētas bibliotēkas „Processing“ un „Python“.

Projektors vai tāfele skolotāja demonstrācijām.

Darba lapa „Grafikas programmēšana“.

- 1) Nodarbības plāns:
- 2) Skolotājs izskaidro zīmēšanas komandas un komentārus (`#`) Python kodā, rādot piemērus uz tāfeles vai projektorā. (5 minūtes)
- 3) Iepazīstina ar sintaksi: `size(width, height)` un sniedz vienkāršu piemēru, piemēram, (300, 300). Izskaidro, kā `size()` nosaka „gleznošanas laukumu“ (izvades logu) visiem zīmējumiem. [1] [2] (5 minūtes)
- 4) Skolotājs izskaidro, kā iestatīt krāsas, izmantojot `fill(R, G, B)` figūras iekšpusei un `stroke(R, G, B)` tās kontūrai. Īsumā piemin, ka R, G, B vērtības ir diapazonā no 0 līdz 255. Parāda izvades loga (`size()`) iestatīšanu, izmantojot līnijas (`line()`), taisnstūra (`rect()`) un elipses (`ellipse()`) funkcijas, izmantojot Python valodu programmā Processing, pievienojot komentārus katrai funkcijai. Izceļ parametru (x, y, width, height) lomu figūru īpašību definēšanā. [3][4][5][6] (15 minūtes)
- 5) Skolēni atkārto skolotāja piemērus savos datoros un eksperimentē ar skaitļu maiņu funkcijās, lai novērotu rezultātus. Skolotājs apstaigā klasi un sniedz palīdzību pēc nepieciešamības. (15 minūtes)
- 6) Skolēni pilda uzdevumus no darba lapas „Grafikas programmēšana“. (25 minūtes)
- 7) Izvēlēti skolēni prezentē savus risinājumus, izmantojot projektoru, vai dalās ar tiem ar klasesbiedriem. Skolotājs izceļ radošas pieejas un bieži sastopamas kļūdas. (10 minūtes)
- 8) Pārdomās (5 minūtes):
- 9) Skolēni pārrunā šādus vadības jautājumus, strādājot pāros vai visai klasei kopā:
 - Ko tu pamanīji par to, kā kods pārvēršas attēlos?
 - Kura koda rakstīšanas daļa tev bija viegla?
 - Kura daļa bija grūta un kāpēc?
 - Apspriediet, kā dažādu pildījuma un kontūru krāsu izmantošana mainīja figūru izskatu.

Papildus idejas:

- Aiciniet skolēnus apvienot figūras, veidojot sarežģītākas figūras (piemēram, automašīnu, koku, robota seju vai ziedu).
- Lūdziet skolēnus pievienot komentārus savam kodam, paskaidrojot katru rindu.
- Ja laiks atļauj, mudiniet skolēnus izpētīt krāsu funkcijas (fill(), stroke()).

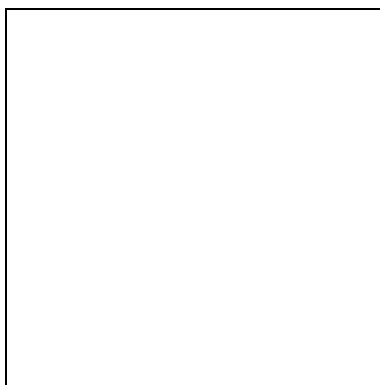
Avoti:

- [1] https://processing.org/reference/size_.html
- [2] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/setup-and-draw>
- [3] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/line>
- [4] https://processing.org/reference/ellipse_.html
- [5] https://processing.org/reference/rect_.html
- [6] https://processing.org/reference/fill_.html

Darba lapa „Grafikas programmēšana”

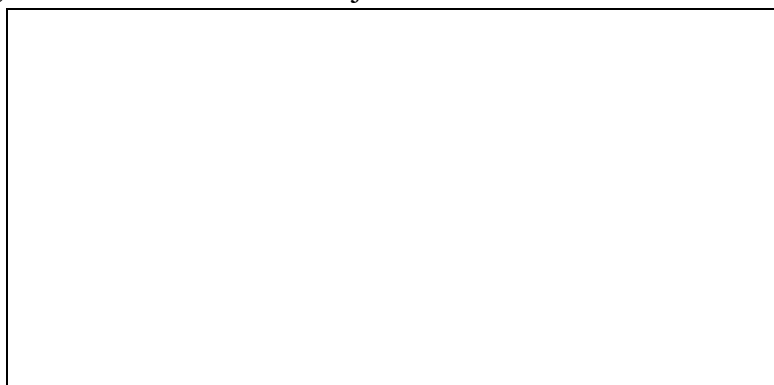
1. uzdevums. Vienkārša zīmēšana

1. Uzraksti programmu, kas atver izvades logu ar izmēriem 500×500 pikseļi (izmanto funkciju `size()`).
2. Uzzīmē taisnstūri loga centrā (izmanto funkciju `rect()`).
3. Uzzīmē apli (elipsi ar vienādu platumu un augstumu) virs taisnstūra (izmanto funkciju `ellipse()`).
4. Uzzīmē diagonālu līniju pāri ekrānam (izmanto funkciju `line()`).
5. Uzzīmējiet savu izvadi



2. uzdevums. Trīsstūra izaicinājums

1. Uzraksti programmu, kas atver izvades logu ar jebkuru izvēlētu izmēru (izmanto funkciju `size()`).
2. Ar trīs `line()` komandām uzzīmē vienādsānu trīsstūri (trīsstūri ar divām vienāda garuma malām) loga centrā. Pārliecinies, ka visas līnijas savienojas, veidojot slēgtu figūru.
3. Uzzīmējiet taisnleņķa trīsstūri (trīsstūri ar vienu 90° leņķi) citā loga daļā, izmantojot trīs `line()` komandas.
4. Pievienojiet komentārus, lai izskaidrotu katru koda rindu.
5. Uzzīmējiet gaidāmo izvadi zemāk esošajā laukā:



3. uzdevums. Iepazīsties ar funkciju `arc()`

Par `arc()`:

Funkcija `arc()` programmā Processing iezīmē elipses daļu (liektu līniju).

- Sintakse:
- `arc(x, y, width, height, start, stop);`

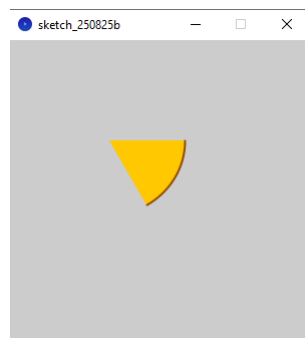
- x, y – elipses centra koordinātas;
- width, height – elipses izmērs;
- start, stop – leņķi radiānos, kas nosaka zīmējamo daļu.

Svarīgi skolēniem:

Leņķi funkcijā `arc()` vienmēr tiek izteikti radiānos, nevis grādos. Lai atvieglotu darbu, varat izmantot funkciju `radians(degrees)`, kas automātiski pārvērš grādus radiānos.

Piemērs:

Uzzīmējiet loku, kas atgādina „picas gabalu“. Rezultāts:



Python kods:

```
size(300, 300)
fill(255, 200, 0)      # Yellow-orange fill for the pizza
stroke(150, 75, 0)     # Brown outline
strokeWeight(2)        # Sets the thickness of the outline (border) of shapes
                        # to 2 pixels
arc(100, 100, 150, 150, radians(0), radians(60)) # Draw a slice (60°)
```

Uzdevums: Pac-Man

- 1.
2. Atver izvades logu ar jebkuru izvēlētu izmēru (izmanto funkciju `size()`).
3. Uzzīmē Pac-Man, izmantojot funkciju `arc()`. Izvēlies sākuma un beigu leņķus tā, lai tas izskatītos kā atvērtas muti.
4. Uzzīmē Pac-Man aci, izmantojot funkciju `ellipse()`.
5. Pievieno komentārus, lai izskaidrotu katru koda rindu.
6. Pēc izvēles: izmantojiet `fill()`, `stroke()` un `strokeWeight()`, lai pielāgotu Pac-Man krāsu un kontūru.

Piemērs:

Loga izmērs: 400 × 400 pikseli

Pac-Man ķermenis: spilgti rozā, sarkana kontūra, kontūras biezums 3.

Acis: melnas.

Rezultāts:



Papildu informāciju par funkciju `arc()` var atrast šeit:

<https://py.processing.org/reference/arc>

Krāsu atgādinājums — RGB formāts

Krāsas programmā „Processing“ tiek norādītas, izmantojot RGB formātu: `fill(R, G, B)` vai `stroke(R, G, B)`

R, G, B nozīmē:

- R = sarkans (krāsas sarkanā komponente);
- G = zaļš (krāsas zaļā komponente);
- B = zils (krāsas zilā komponente).

Katra vērtība var būt no 0 līdz 255:

- 0 = nav šīs krāsas;
- 255 = pilna šīs krāsas intensitāte.

R, G un B kombinācija nosaka galīgo krāsu:

- `fill(255, 0, 0)` – spilgti sarkana;
- `fill(0, 255, 0)` – spilgti zaļa;
- `fill(0, 0, 255)` – spilgti zila;
- `fill(255, 255, 0)` — dzeltena (sarkana + zaļa);
- `fill(0, 255, 255)` — ciāna (zaļa + zila).

Padoms: Mainot R, G un B vērtības, varat radīt miljoniem dažādu krāsu savām figūrām.

4.3. modulis. Ievads mainīgo lielumu lietošanā

Mērķis: Izprast un izmantot mainīgos, izmantojot Python valodu programmā Processing, kā arī apgūt, kā funkcijas `setup()` un `draw()` kontrolē programmas gaitu.

Mācību sasniedzamais rezultāts:

Līdz nodarbības beigām skolēni spēs:

- Saprast, kas ir mainīgais, un kā tas var uzglabāt un mainīt vērtības.
- Iepazīties ar atšķirību starp `setup()` (izpildās vienreiz) un `draw()` (izpildās nepārtraukti).
- Izmantot mainīgos, lai vadītu vizuālos elementus (piemēram, pozīcijas, izmērus vai krāsas).
- Sākt izstrādāt strukturētu kodu.

Nepieciešamie materiāli:

Datori, kuros ir instalētas bibliotēkas „Processing” un „Python”.

Projektors vai tāfele skolotāja demonstrācijām.

Darba lapa „Mainīgie darbībā”.

Darba lapa „Setup vs Draw Challenge”.

Nodarbības plāns:

1. Skolotājs iepazīstina ar mainīgo jēdzienu, salīdzinot tos ar „kastēm, kurās tiek uzglabāta informācija”, un izskaidro to lomu programmēšanā. [1] (10 minūtes)
2. Skolotājs demonstrē vienkāršu piemēru: apli, kura x-pozīciju kontrolē mainīgais. Parāda, kā mainīgā vērtības maiņa liek aplim kustēties. [1] (10 minūtes)
3. Skolēni atkārtoti kustīgā apla piemēru savos datoros. Viņi eksperimentē ar ātrumu un virzienu, mainot mainīgo lielumu vērtības (liekot aplam kustēties). (15 minūtes)
4. Skolotājs izskaidro `setup()` un `draw()` struktūru, uzsverot, ka `setup()` tiek izpildīts vienreiz inicializācijai, bet `draw()` atkārtojas nepārtraukti animācijas radīšanai. [3][4][5][6] (10 minūtes)
5. Skolēni aizpilda darba lapu “Mainīgie darbībā”, ievadot koda fragmentus un atbildot uz īsiem jautājumiem. (15 minūtes)
6. Skolotājs ievada nelielu uzdevumu: ievietot `print(“Hello”)` gan `setup()`, gan `draw()`. Skolēni prognozē atšķirību, pēc tam pārbauda un reģistrē rezultātus darba lapā “Setup vs Draw Challenge”. (15 minūtes)
7. Pārdomās (5 minūtes):

Visi kopā skolēni īsumā pastāsta par vienu piemēru, kā viņi savā zīmējumā izmantoja mainīgo, vai arī par to, ko viņi atklāja par to, kā funkcijas `setup()` un `draw()` darbojas atšķirīgi. Skolotājs mudina skolēnus, izklāstot savus secinājumus, lietot pareizo terminoloģiju (mainīgais, cilpa, inicializācija)

Papildus idejas:

- Darba lapā „Mainīgie darbībā” skolēni var paplašināt savus skices, pievienojot vairāk mainīgo (piemēram, krāsu vai izmēru), lai objekti kustētos vai mainītos dinamiski.

- Darba lapā „Setup vs Draw Challenge” skolēni var modificēt piemēru, iekļaujot mainīgo, kas mainās draw() laikā (piemēram, pakāpeniski pārvietojot apli pa ekrānu), lai novērotu, kā atkārtota izpilde rada animāciju.
- Mudiniet spējīgākos skolēnus izstrādāt īsu animāciju, kas apvieno mainīgos, cilpas un krāsu maiņas.

Avoti:

- [1] <https://processing.org/examples/variables.html>
- [2] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/variables>
- [3] https://processing.org/reference/draw_.html
- [4] <https://processing.org/examples/setupdraw.html>
- [5] https://processing.org/reference/setup_.html
- [6] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/setup-and-draw>

Darba lapa „Mainīgie darbībā”

1. uzdevums. Piemērs

Uzmanīgi izlasiet zemāk sniegto piemēru:

```
x = 50
y = 200
speed = 3

def setup():
    size(400, 400)
    background(255)

def draw():
    global x, speed
    background(255)
    fill(255, 100, 100)
    ellipse(x, y, 50, 50)
    x = x + speed

    if x > 375 or x < 25:
        speed = -speed
```

Atbildiet uz jautājumiem:

1. Kas liek bumbiņai mainīt virzienu?

2. Kas notiktu, ja noņemtu līnijas fonu (255)?

3. Kuras mainīgās lielums nosaka kustību un kuras — izskatu?

2. uzdevums. Izveido savu animāciju

Izveido savu skici, izmantojot vismaz divus mainīgos.

Ideju piemēri:

- Kosmosa kuģis, kas lido pāri debesīm.
- Robots, kas mirgo ar acīm.
- Gaisa balons, kas virzās uz augšu un maina krāsu.

- Tu vari arī izdomāt savu oriģinālu ideju, ja vien tajā tiek izmantoti vismaz divi mainīgie, lai vadītu kustību, krāsu vai citas īpašības.

Pirms sāc programmēt, apraksti savu ideju zemāk:

Mana projekta ideja:

Uzraksti to mainīgo vārdu nosaukumus, kurus plāno izmantot savā programmā (piemēram: x_position, y_position, colour_change), un īsi apraksti, ko katrs mainīgais kontrolē:

Darba lapa „Setup vs Draw Challenge”

Ekperimentē ar setup() un draw(), lai izprastu to izpildes laiku, atkārtosanos un lomu Processing programmās.

1. uzdevums. Izvērtējiet rezultātu

Izpētiet šo kodu un pirms tā izpildes uzrakstiet, kā, jūsuprāt, tas darbosies:

```
def setup():
    print("Setup runs once")

def draw():
    print("Draw runs many times")
```

Your prediction:

2. uzdevums. Palaišana un reģistrēšana

Palaižiet programmu un pārbaudiet konsoles izvadi. Uzrakstiet, cik reizes parādās katrs ziņojums.

Funkcija	Message Printed	Cik reizes?
setup()		
draw()		

3. uzdevums. Maini un izmēģini

Pārveidojiet kodu, iekļaujot tajā savu tekstu:

```
def setup():  
    print("Welcome to Processing!")  
  
def draw():  
    print("Learning is fun!")
```

Palaidiet to vēlreiz un aprakstiet, kas notiek:

4. uzdevums. Izklāstiet, ko esat iemācījušies

Vienā vai divos teikumos aprakstiet, kad jūs izmantotu katru funkciju.

setup():

draw():

4.4. modulis. Radošais projekts: no koda līdz spēlei

Mērķis: Pielietot zināšanas par mainīgajiem un programmas struktūru, lai izstrādātu un īstenotu vienkāršu interaktīvu projektu, izmantojot „Processing” ar Python.

Sasniedzamie rezultāti:

Līdz nodarbības beigām studenti spēs:

- Padziļināt izpratni par mainīgajiem, piemērojot tos kustīgiem objektiem.
- Izmantot `setup()` un `draw()`, lai radītu dinamiskas un nepārtrauktas animācijas.
- Eksperimentēt ar interaktivitāti.
- Attīstīt radošumu un problēmu risināšanas prasmes, izmantojot vadītu projektu.

Materiāli:

Datori, kuros ir instalēta Processing un Python bibliotēka.

Projektors vai tāfele skolēnu demonstrācijām.

Darba lapa “Radošais programmēšanas uzdevums: lidojoša bumba”.

Nodarbības plāns:

1. Skolotājs atkārti, kas ir mainīgie un kā darbojas `setup()` un `draw()`. Iepazīstina ar animācijas jēdzienu, izmantojot kustīgu objektu, kas laika gaitā maina savu pozīciju. Parāda vienkāršu piemēru, kā bumba pārvietojas horizontāli, izmantojot mainīgo lielumu pozīcijai. [1][2][3][4] (10 minūtes)
2. Skolotājs parāda, kā panākt, lai bumba atlēktu no loga malām, mainot virzienu, kad tā sasniedz malu. Tiek parādīts koda piemērs un izskaidrots solis pa solim, uzsverot mainīgo lielumu nozīmi pozīcijai, ātrumam un virzienam. [5] (10 minūtes)
3. Skolēni atkārti piemēru savos datoros, eksperimentējot ar (Skolotājs apstaigā klasi un palīdz skolēniem, ja nepieciešams.): (20 minūtes)
 - Dažādiem loga izmēriem (`size(width, height)`)
 - Bumbas krāsas maiņu (`fill(R, G, B)`)
 - Ātruma vai virziena regulēšanu.
4. Skolēni aizpilda darba lapu „Radošais programmēšanas uzdevums: lidojoša bumba”, rakstot programmu, kas liek bumbiņai kustēties un lēkt loga iekšienē. (30 minūtes)
5. Izvēlēti skolēni prezentē savus projektus, izmantojot projektoru. Skolotājs izceļ radošumu, pareizu mainīgo izmantošanu un problēmu risināšanas stratēģijas. (10 minūtes)
6. Pārdomās (5 minūtes):

Pāros skolēni īsi apspriež, kas lika viņu bumbiņai kustēties un lēkt. Katrs skolēns uz savas darba lapas uzraksta vienu īsu teikumu, atbildot uz jautājumu: “Kā mainīgie palīdzēja kontrolēt tavu animāciju?”

Papildus idejas:

- Pievienojiet programmai interaktivitāti: lieciet bumbiņai mainīt virzienu, nospiežot taustiņu, vai krāsu, uzklikšķinot uz tās.
- Izveidojiet vairākas bumbiņas, kas pārvietojas ar atšķirīgu ātrumu, izmantojot mainīgo sarakstus.
- Mudiniet spējīgākos skolēnus pārvērst lēkājošo bumbiņu par īsu mini-spēli (piemēram, „noķer bumbiņu ar raketi”).

- Eksperimentējiet ar skaņas vai fona krāsas izmaiņām, lai padarītu nodarbību vēl saistošāku.

Avoti:

- [1] https://processing.org/reference/draw_.html
- [2] https://processing.org/reference/setup_.html
- [3] <https://processing.org/examples/setupdraw.html>
- [4] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/setup-and-draw>
- [5] <https://processing.org/examples/bounce.html>

Darba lapa „Radošais programmēšanas uzdevums: lidojoša bumba”

Izmanto savas zināšanas par mainīgajiem, `setup()` un `draw()`, lai izveidotu animētu bumbu, kas pārvietojas un lido pa ekrānu.

1. uzdevums. Izpēti piemēru

Uzmanīgi izpēti zemāk sniegto piemēru:

```
def setup():
    size(400, 400)
    global x, y, speed_x, speed_y
    x = 200
    y = 200
    speed_x = 3
    speed_y = 2

def draw():
    background(255)
    fill(255, 0, 150)
    ellipse(x, y, 40, 40)

    global x, y, speed_x, speed_y
    x = x + speed_x
    y = y + speed_y

    if x > width or x < 0:
        speed_x = -speed_x
    if y > height or y < 0:
        speed_y = -speed_y
```

Atbildi uz jautājumu:

Kas notiek, kad bumbiņa sasniedz loga malu?

2. uzdevums. Pielāgošana un eksperimentēšana

Maini vismaz vienu animācijas aspektu:

- Bumbiņas krāsa (`fill(R, G, B)`)
- Bumbiņas ātrums (`speed_x, speed_y`)
- Bumbas izmērs
- Loga izmērs

Apraksti, ko tu mainīji:

3. uzdevums. Izveido savu lidojošo bumbu

Tagad izveido savu lidojošās bumbas versiju!

Tavai programmai ir:

- Jāizmanto vismaz trīs mainīgie.
- Jāliek bumbai kustēties un lēkt loga iekšienē.
- Pēc izvēles iekļauj krāsas maiņu, interaktivitāti vai unikālu kustību.

Uzraksti savu ideju šeit:

4.5. modulis. Ievads Arduino komponentos

Sasniedzamais rezultāts:

Līdz nodarbības beigām studenti spēs:

- Atpazīt un aprakstīt Arduino 101 plati, Education Shield, eksperimentu plati, sensorus un aktuātorus.
- Izskaidrot katras galvenās sastāvdaļas lomu shēmu un robotu veidošanā.

Materiāli:

Arduino CTC 101 Program – FULL (AKX00002) rīku komplekts.

Darba lapa “Arduino CTC 101 komponentu identificēšana”.

Nodarbības plāns:

1. Ievads Arduino, galvenās sastāvdaļas. Izskaidrojums par to, kā Arduino atšķiras no LEGO robotikas un citām skolēniem pazīstamām robotu platformām. [1] (10 minūtes)
2. Galveno Arduino 101 sastāvdaļu un to funkciju izskaidrojums: shields, plātnes, rezistori, moduļi, skaļruņi, vadi utt. [1][2][3][4][5][6][7][8][9] (20 minūtes)
3. Izmantojot darba lapu „Arduino CTC 101 komponentu identificēšana”, skolēni izpako Arduino CTC 101 komplektu un identificē katru komponentu. Kā papildu uzdevumu var dot tikai komponentu nosaukumus, liekot skolēniem „Attēls” ailē uzzīmēt katra komponenta shēmu. Šī aktivitāte nostiprina komponentu atpazīšanu un to funkciju izpratni. (30 minūtes)
4. Organizējiet klases diskusiju, kurā skolēni apsver un apspriež: „Kura sastāvdaļa, jūsuprāt, būs visbūtiskākā robota izveidei, un kāpēc?” (10 minūtes)
5. Refleksija (10 minūtes):

Organizējiet klases diskusiju, kurā skolēni nosauc vienu sastāvdaļu un izskaidro tās funkciju. Skolēniem nav atļauts atkārtot jau minētās sastāvdaļas.

Papildus uzdevumi:

- Tā vietā, lai darba lapā iekļautu attēlus, sniedziet skolēniem tikai komponentu nosaukumus un funkciju aprakstus.
- Skolēniem ir jāuzglezno vienkārša shēma vai attēls katrai sastāvdaļai darba lapas „Arduino CTC 101 Component Identification” (Arduino CTC 101 sastāvdaļu identificēšana) „Attēls” (Attēls) ailē.
- Mudiniet skolēnus pievienot saviem zīmējumiem atbilstošas piezīmes, piemēram, par krāsu, formu vai unikālām iezīmēm.
- Pēc zīmējumu pabeigšanas skolēni var salīdzināt savus skices ar reālajām sastāvdaļām, lai apspriestu līdzības un atšķirības

Avoti:

[1] <https://docs.arduino.cc/hardware/>

[2] <https://ctc101.arduino.cc/ctc101/module/module-1/lesson/processing>

[3] <https://ctc101.arduino.cc/ctc101/module/module-2/lesson/what-is-the-ctc-board>

[4] <https://ctc101.arduino.cc/ctc101/module/module-3/lesson/reading-analog-signals>

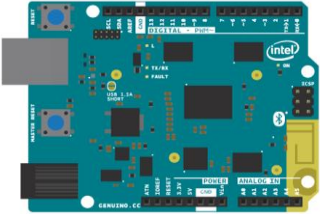
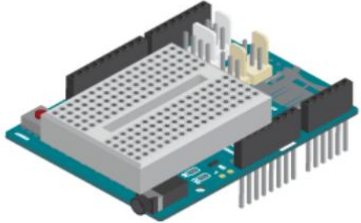
- [5] <https://ctc101.arduino.cc/ctc101/module/module-3/lesson/writing-analog-signals>
- [6] <https://ctc101.arduino.cc/ctc101/module/module-3/lesson/light-sensor>
- [7] <https://ctc101.arduino.cc/ctc101/module/module-3/lesson/serial-port>
- [8] <https://ctc101.arduino.cc/ctc101/module/module-4/lesson/types-of-motors>
- [9] <https://ctc101.arduino.cc/ctc101/module/module-5/lesson/what-is-an-imu>

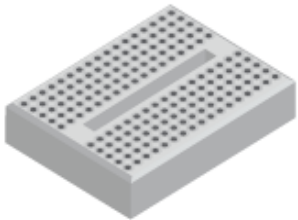
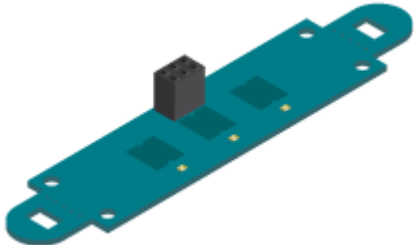
Darba lapa „Arduino CTC 101 komponentu identificēšana”

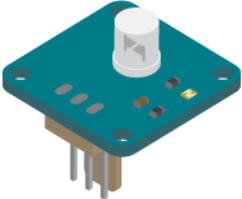
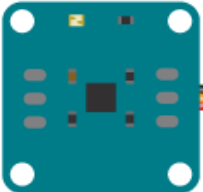
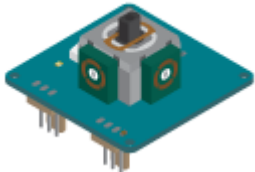
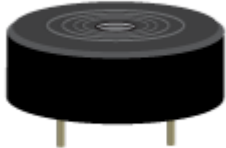
1. uzdevums

Izpakojiet savu Arduino CTC 101 komplektu un identificējiet katru komponentu, izmantojot zemāk esošo tabulu. Aizpildiet sleju „Piezīmes”, ja:

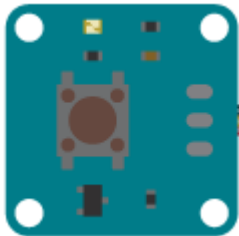
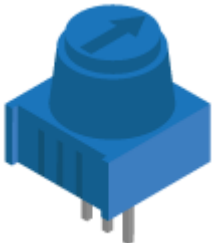
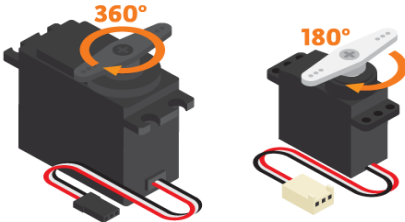
- komponenta krāsa atšķiras no attēlā redzamās,
- jums bija grūtības atrast komponentu,
- ir kādi citi komentāri vai novērojumi par komponentu.

Komponenta nosaukums	Attēls	Funkcija	Piezīmes	Atrasts ☒
Arduino 101 Board		Galvenais mikrokontrolers ar Bluetooth un IMU (inerciālo mērīšanas sistēmu)		☐
Education Shield		Paplašina plati, sakārto komponentes, SD/audio		☐

Komponenta nosaukums	Attēls	Funkcija	Piezīmes	Atrasts ☒
Breadboard		Veidojiet shēmas bez lodēšanas		☐
IR Array		Sastāv no 3 IR (infrasarkan) sensoriem		☐
LED		Gaismas indikators		☐
Resistor		Strāvas ierobežojums		☐
Push Button		Manuāla ieslēgšana/izslēgšana		☐

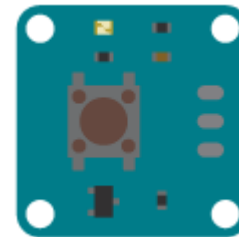
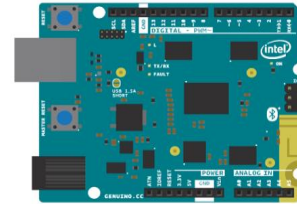
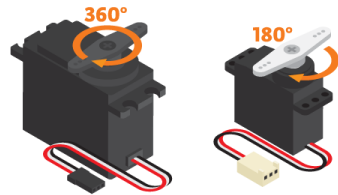
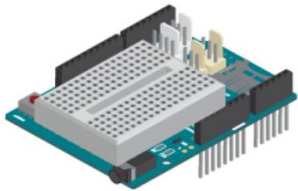
Komponenta nosaukums	Attēls	Funkcija	Piezīmes	Atrasts ☒
Light Sensor Module		Gaismas intensitātes noteikšana		☐
Tilt Sensor Module		Orientācijas noteikšana		☐
Joystick Modulino		Nodrošina 2-asu ieeju		☐
Piezo Speaker		Skaņas izvade caur audio savienotāju		☐

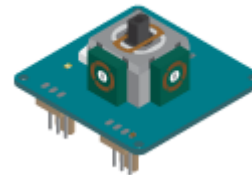
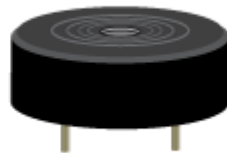
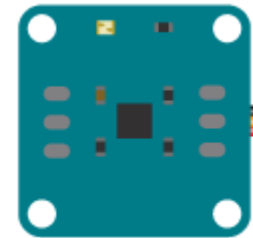
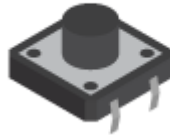
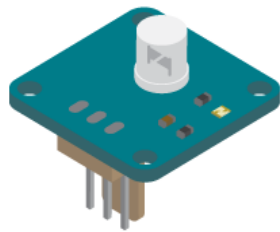
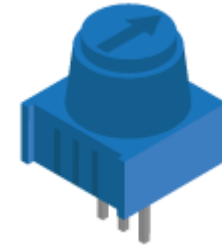
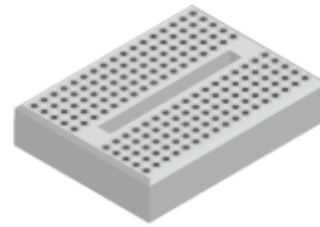
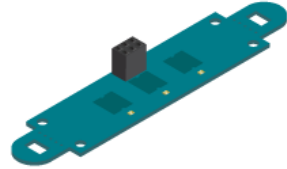
Komponenta nosaukums	Attēls	Funkcija	Piezīmes	Atrasts ☒
Speaker		Skaņas izvade		☐
USB Cable		Pieslēgt plati datoram		☐
Jumper Wires		Pieslēgt komponentus		☐
Module Connector Wire		Pieslēgt ārējo aparatūru		☐

Komponenta nosaukums	Attēls	Funkcija	Piezīmes	Atrasts ☒
Push Button TaModule		Manuāla ieslēgšana/izslēgšana		☐
Potentiometer		Vadīt projektu, nodrošinot mainīgu analoģu ieejas signālu		☐
Web Camera		Video uztveršana un pārraide		☐
Servo Motor		Precīza kustības kontrole		☐

2. uzdevums

Pie katra attēla ierakstiet atbilstošo komponenta nosaukumu no sniegtā saraksta: Arduino 101 Board, Education Shield, Breadboard, IR Array, LED, Resistor, Push Button, Light Sensor Module, LED, Tilt Sensor Module, Joystick Modulino, Piezo Speaker, Speaker, USB Cable, Jumper Wires, Module Connector Wire, Push Button Module, Potentiometer, Web Camera, Servo Motor.





5. modulis. Vadības plātnes programmēšanas pamati

Mērķis: Izprast vadības plātnes uzbūvi. Apgūt digitālo tehnoloģiju pamatus, lai vadītu digitālos aktuatorus un nolasītu digitālo sensoru rādījumus.

Pārskats:

- 2.1. modulis. Plātnes instalēšana un pieslēgšana IDE (integrētā attīstības vidē)
- 2.2. modulis. LED mirgošanas un skaņas radīšanas programmēšana, izmantojot pjezo skaļruni
- 2.3. modulis. Projekta izstrāde: nelielas elektroniskas spēles izveide un spēlēšana, kas simulē sporta spēles

Laiks/stundas: 6 vai 8 (3 vai 4 dubultstundas – 2x40 min)

5.1. modulis. Plates uzstādīšana un pieslēgšana IDE (integrētā attīstības vidē)

Mācību mērķis:

Līdz nodarbības beigām studenti spēs izprast, kas ir vadības platiņa, un aprakstīt tās uzbūvi, uzstādīt un pieslēgt vadības platiņu IDE, strādāt ar IDE, izmantojot rīkjoslū un ziņojumu logu, augšupielādēt un kompilēt programmas, kā arī izprast ciparu signālus.

Nepieciešamais aprīkojums:

- 101 vadības plate
- Interneta savienojums
- USB kabelis

Nodarbības plāns:

1. Izskaidrojiet studentiem vadības plātnes nozīmi un uzbūvi, izmantojot Arduino 101 plātni kā piemēru. Miniet citus piemērus [1, 4] (10 minūtes).
2. Iepazīstiniet studentus ar komponentiem, kas nepieciešami plātnes pieslēgšanai (2 minūtes).
3. Sniegt studentiem instrukcijas par to, kā instalēt IDE. Ļaut viņiem instalēt IDE [2 vai 1. uzdevums] (20 minūtes).
4. Iepazīstināt studentus ar IDE, t.i., parādīt rīkjoslū un ziņojumu zonas funkcijas [1]. Lūgt studentiem izpildīt 3. uzdevumu (10 minūtes).
5. Pieslēgt plati pie IDE, sekojot instrukcijām, kas sniegtas 2. uzdevumā vai [3] (20 minūtes).
6. Izskaidrojiet Education Shield iespējas [5] un digitālos signālus [6] (10 minūtes).
7. Pārdomās (8 minūtes)
 - Veiciet uzdevumus 1, 2, 3 un 4.
 - Studenti apkopojiet to, ko esat iemācījušies, un dalieties ar personīgajām atziņām vai vēlmēm par vadības plātēm, to veidiem un Education Shield.

Avoti:

- [1] <https://ctc101.arduino.cc/ctc101/module/module-2/lesson/what-is-the-ctc-board>
- [2] <https://ctc101.arduino.cc/resources?id=31820576>
- [3] <https://ctc101.arduino.cc/resources?id=31818016>
- [4] <https://ctc101.arduino.cc/resources?id=32381557>
- [5] <https://ctc101.arduino.cc/resources?id=31800414>
- [6] <https://ctc101-dev.arduino.cc/ctc101/module/module-2/lesson/digital-signals>

Darba lapa „IDE iepazīšana”

1. uzdevums. IDE instalēšana

Windows – ar administratora tiesībām

1. Apmeklējiet šo lapu, lai atrastu jaunāko IDE versiju:
<http://arduino.cc/en/Main/Software>
2. Pārvietojieties uz leju un noklikšķiniet uz „Windows Installer“, lai lejupielādētu jaunāko IDE versiju
3. Kad lejupielāde ir pabeigta, atveriet lejupielādēto failu.
4. Noklikšķiniet uz „I agree“, lai piekristu licences līgumam.
5. Izvēlieties visus instalējamos komponentus un noklikšķiniet uz „Next” (Tālāk).
6. Atcerieties galamērķa mapi un noklikšķiniet uz „Install” (Instalēt).
7. Kad instalēšana ir pabeigta, noklikšķiniet uz „Close” (Aizvērt).

Windows – bez administratora tiesībām

1. Apmeklējiet šo lapu, lai atrastu jaunāko IDE versiju:
<http://arduino.cc/en/Main/Software>
2. Pavelciet uz leju un noklikšķiniet uz „Windows ZIP file (for non-administrator install)”, lai lejupielādētu.
3. Kad lejupielāde ir pabeigta, izpakoiet failu uz datora cietā diska.

Drivers

1. Pievienojiet vadības plati datoram, izmantojot USB kabeli.
2. Pagaidiet, līdz Windows sāks draiveru instalēšanas procesu. Pēc brīža process neizdosies, neskatoties uz visiem centieniem.
3. Atveriet „Vadības paneli”.
4. Atveriet sadaļu „Sistēma un drošība”. Pēc tam noklikšķiniet uz „Sistēma”. Kad atvērsies „Sistēma” logs, atveriet „Ierīču pārvaldnieku”.
5. Meklējiet sadaļā „Ports (COM & LPT)”. Jums vajadzētu redzēt atvērtu portu ar jūsu vadības plātnes nosaukumu un portu, kas izskatās šādi: „(COMxx)”. Ja nav sadaļas „COM & LPT”, meklējiet sadaļā „Citas ierīces” pēc „Nezināma ierīce”.
6. Ar peles labo pogu noklikšķiniet uz ierīces ar jūsu vadības plātnes nosaukumu un izvēlieties opciju „Atjaunināt draiveri”.
7. Tālāk izvēlieties opciju „Meklēt draiveri datorā”.

8. Beidzot, atveriet un izvēlieties draiveru failu ar nosaukumu „arduino.inf”, kas atrodas Arduino programmatūras lejupielādes mapē „Drivers” (nevis apakšmapē „FTDI USB Drivers”). Ja izmantojat vecu IDE versiju (1.0.3 vai vecāku), izvēlieties Uno draivera failu ar nosaukumu „Arduino UNO.inf”.
9. Windows turpinās draivera instalēšanu no šī brīža.

Mac OS X

1. Apmeklējiet šo lapu, lai atrastu jaunāko Arduino IDE versiju:
<http://arduino.cc/en/Main/Software>
2. Pavelciet uz leju un noklikšķiniet uz „Mac OS X”, lai sāktu lejupielādi.
3. Kad lejupielāde ir pabeigta, divreiz noklikšķiniet uz .zip faila.
4. Kopējiet Arduino lietojumprogrammu mapē „Applications”.

Linux - with administrator rights

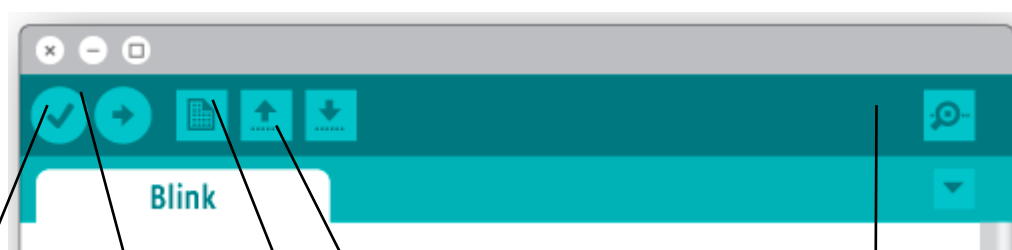
1. Apmeklējiet šo lapu, lai atrastu jaunāko Arduino IDE versiju:
<http://arduino.cc/en/Main/Software>
2. Pavelciet uz leju un noklikšķiniet uz „32 bit” vai „64 bit”, atkarībā no jūsu sistēmas tipa, lai lejupielādētu.
3. Apmeklējiet šo lapu un izpildiet jūsu distribūcijai paredzēto pamācību
<http://playground.arduino.cc/Learning/Linux>

2. uzdevums. Plates pieslēgšana IDE

1. Atveriet izvēlni „Tools → Port” un pārbaudiet, kādi porti ir pieejami.
2. Pieslēdziet vadības plati datoram, izmantojot USB kabeli.
3. Atveriet izvēlni „Tools → Port” vēlreiz – tur būtu jāparādās jaunam portam, kas ir jūsu plate.
4. Izvēlieties šo jauno portu.

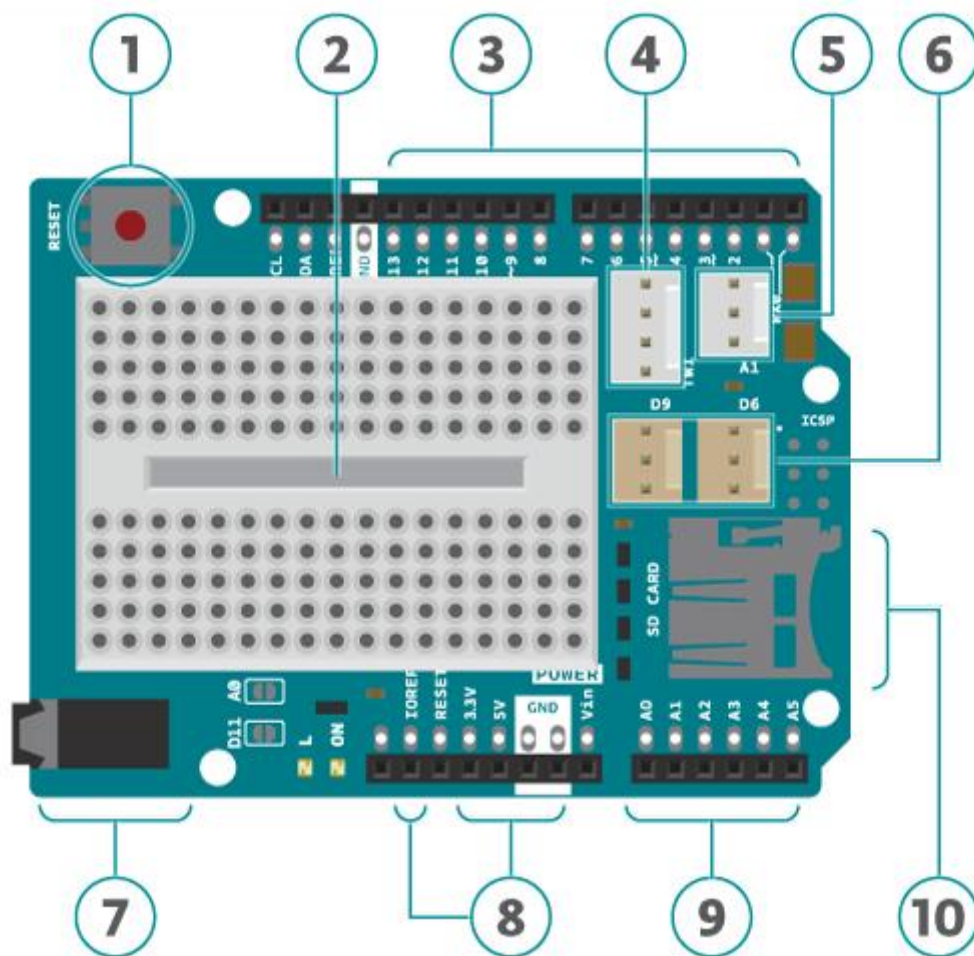
3. uzdevums. IDE rīkjoslas funkciju apraksts

Aprakstiet katras IDE rīkjoslā iekļautās funkcijas darbību. Uzrakstiet aprakstu lodziņos, kas saistīti ar šīm funkcijām.



4. uzdevums. Vadības plates elementu atpazīšana

Zem attēla esošajā sarakstā pie katra numura ierakstiet attiecīgā plātnes elementa pareizo nosaukumu. Izmantojiet elementu nosaukumu sarakstu: **reset button** (atiestatīšanas poga), **analog input pins** (analogās ieejas kontakti), **I2C connector** (I2C savienotājs), **digital 3-pin header ports** (digitālie 3-kontaktu galvenes porti), **speaker plug** (skaļruņa spraudnis), **digital input and output pins** (digitālās ieejas un izejas kontakti), **SD card port** (SD kartes ligzda), **breadboard** (maketplate), **ground and power pins** (zemējuma un barošanas kontakti), **A1 3-pin header port** (A1 3-kontaktu galvenes ports).



- | | | | |
|----|--------------|-----|-------|
| 1. | reset button | 6. | _____ |
| 2. | _____ | 7. | _____ |
| 3. | _____ | 8. | _____ |
| 4. | _____ | 9. | _____ |
| 5. | _____ | 10. | _____ |

5.2. modulis. LED mirgošanas un skaņas radīšanas programmēšana, izmantojot Piezo Speaker

Sasniedzamais rezultāts:

Līdz nodarbības beigām skolēni spēs izprast digitālos signālus, atšķirt digitālās ieejas un izejas, pieslēgt dažādus sensorus, izstrādāt vienkāršas programmas ārējo digitālo sensoru (piem., ārējo LED, skaļruni) nolasīšanai un digitālo aktuatoru vadībai, kā arī izprast un izmantot dažādas klases, funkcijas un komandas no bibliotēkas „Education Shield“.

Materiāli:

- 1 x vadības plate
- 1 x Arduino Education Shield
- 5 x LED
- 5 x 220 omu rezistors
- 1 x piezo skaļrunis
- 10 x savienotājsvadi

Nodarbības plāns:

1. Atkārtojiet digitālo signālu jēdzienu [1]. Parādiet vienkāršu programmu, kā panākt, lai uz platīnes esošais LED mirgotu, un izskaidrojiet visas koda rindas [2, 1. piemērs] (5 minūtes).
2. Izskaidrojiet LED polaritāti. Parādiet skolēniem, kā savienot savu LED ar vadības plati [2] un kā pasargāt to no pārkaršanas, izmantojot rezistorus [3] (5 minūtes).
3. Parādiet skolēniem paraugprogrammu, kā panākt, lai ārējais LED mirgotu [2, 2.2. piemērs]. Parādiet visus nepieciešamos materiālus (vadības plati, LED, rezistoru, Arduino Education Shield, eksperimentu plati, savienojuma vadu), izskaidrojiet to mērķi [4, 5]. Parādiet, kā tos pieslēgt vadības platei (5 minūtes).
4. Izskaidrojiet visas paraugprogrammas koda rindas [2, piemērs 2.2. vai 2.3. ar mainīgo], bibliotēkas nozīmi un tajā izmantotās funkcijas (5 minūtes).
- 4*. Papildus izskaidrojiet koda struktūru, ja skolēni nav apguvuši šī kursa 1. moduli.
 - Katra koda rinda beidzas ar semikolu ;.
 - Koda bloki ir ietverti izliektās iekavās { }.
 - Funkcijas sākas ar definīciju, piemēram, `void setup() { ... }.`
 - Funkcijām ir parametri, kas ietverti iekavās (), for example, `pinMode(inputPin, INPUT); digitalWrite(ledPin, HIGH); piezo.play(numberOfNotes, melody, noteDurations, 1);`. Parametru skaits ir atkarīgs no funkcijas.
5. Galvenā izpēte (20 minūtes)
 - 5.1. Ļaujiet skolēniem izpildīt uzdevumu, kas aprakstīts [2, 2.2. piemērs]. Lūdziet skolēnus pievienot savu LED un citus materiālus eksperimentu platē.

5.2. Nokopējiet kodu no [2, 2.2. piemērs]. Lūdziet skolēniem ar savienojuma vadu pievienot 220 omu rezistoru citam digitālajam kontaktam uz eksperimentu plāksnes (citam nekā norādīts 2.2. piemērā).

5.3. Lūdziet skolēniem modificēt kodu [2, 2.2. piemērs] un izmantot mainīgo, piemēram, int PinNumber, lai saglabātu digitālās pines numuru, pie kuras ir pieslēgts LED, nevis vienkārši ierakstīt numuru funkcijā pinMode(). Skatīt piemēru [2, 2.3. piemērs]. Kompilējiet un augšupielādējiet programmu.

5.4. Lūdziet studentiem mainīt laika vērtības delay funkcijā, izmantojot kodu [2, 2.2. vai 2.3. piemērs]. Ļaujiet viņiem izmēģināt dažādus veidus, kā padarīt gaismas mirgošanu ātrāku vai lēnāku.

5.5. Lūdziet studentiem novērot, kas notiek, ja delay vērtība kļūst ļoti maza, piemēram, delay(10), un izdarīt secinājumus. Lūdziet studentiem pabeigt 1. uzdevumu, t.i., ievadīt eksperimenta rezultātus tabulā.

6. Skaņas ģenerēšana, izmantojot pjezo skaļruni (3 minūtes). Izskaidrojiet, kā skaņa tiek radīta, ģenerējot vibrācijas ar pjezoelektrisko skaņas signālu (pjezo skaļruni), liekot tai skanēt. Izskaidrojiet pjezo skaļruņa ierīci un to, kā tā darbojas.

7. Pjezo skaļruņa pieslēgšana un programmēšana (5 minūtes). Parādiet skolēniem, kā pieslēgt pjezo skaļruni vadības platē [6], un izskaidrojiet visas paraugprogrammas [2, piemērs 2.4. vai 2.5.] koda rindas.

8. Toņos un frekvencēs (5 minūtes). Izskaidrojiet toņu nosaukumus un tiem atbilstošās frekvences, kā arī to, kā atskaņot dažādus toņus, izmantojot funkciju tone() [6]. Parādiet koda piemēru, kurā funkcija tone() tiek izmantota, lai ģenerētu A toņu ar frekvenci 440 hercu (5 minūtes).

9. Izpēte (7 minūtes).

– Lūdziet skolēniem kompilēt un augšupielādēt programmu [6, piemērs 2.4. vai 2.5.].

– Lūdziet skolēniem ģenerēt C, D, E, F, G vai B, mainot paraugprogrammas kodu [6, piemērs 2.6.]. Ļaujiet skolēniem mainīt skaņas ilgumu, pielāgojot funkcijas tone() trešo parametru.

– Lūdziet studentiem izpildīt 2. uzdevumu.

10. Darbs ar vairākiem LED (8 minūtes). Izskaidrojiet, kā vairākus LED var savienot kopā vienā rindā, izmantojot VU-meter, un kā tos vadīt, izmantojot EducationShield bibliotēku. Parādiet piemēru ar 5 LED [7]. Izskaidrojiet parauga kodā izmantotos operatorus un funkcijas, piemēram `VuMeter me, me.config(), me.begin(), me.fill(), me.clear(), me.on(), me.off(), me.blink(), me.blinkAll(), me.scrollLeft(), me.scrollRight(), me.fillFrom()`.

11. Izpēte (12 minūtes).

Lūdziet skolēniem:

– izpildīt 3. uzdevumu;

– pieslēgt trīs gaismas diodes uz plates saskaņā ar norādījumiem, kas sniegti [7];

– nokopēt parauga kodu no [7 vai izmantojot IDE rīkjoslas izvēlni File>Example>EducationShield>Help>vuMeter], kompilēt un augšupielādēt programmu;

– veikt izmaiņas kodā, lai panāktu šādu rezultātu:

1. visas LED gaismas mirgo 5 reizes pa 1,5 sekundēm;

2. ieslēdz vienu LED gaismu pēc otras no kreisās puses uz labo, katra LED gaisma ir ieslēgta 2 sekundes;
3. ieslēdz vienu LED gaismu pēc otras no labās puses uz kreiso, katra LED gaisma ir ieslēgta 1 sekundi;
4. visas LED gaismas mirgo 5 reizes pa 1,5 sekundēm;
5. mirgo 2. un 3. LED gaisma ar 2 sekunžu pauzi;
6. mirgo tikai 3. LED ar 2 sekunžu pauzi;
7. ieslēdz visus LED uz 5 sekundēm;
8. izslēdz 1. LED uz 3 sekundēm;
9. ieslēdz 1. LED uz 2 sekundēm;
10. izslēdz visus LED.

12. Pārdomu brīdis (5 minūtes)

– Skolēni apkopos apgūto un dalīsies ar personīgajām atziņām vai viedokli par programmēšanas vadības plati, ārējo digitālo sensoru (ārējais LED, skaļrunis) nolasīšanu un digitālo aktuatoru vadību.

Papildu uzdevumi:

- Izmanto gaismu, lai imitētu sirdsdarbību — kaut ko līdzīgu „tik-tak... pauze... tik-tak... pauze...”.
- Uzraksti programmu, kas atskaņo skaņu, nospiežot pogu.
- Pievieno LED, kas mirgo, nospiežot pogu.
- Uzraksti melodiju, izmantojot funkciju `tone()` [8].
- Uzraksti melodiju, izmantojot funkciju `play()` no bibliotēkas „EducationShield”.

Avoti:

[1] <https://ctc101.arduino.cc/resources?id=32372703>

[2] <https://ctc101.arduino.cc/ctc101/module/module-2/lesson/blink>

[3] <https://ctc101.arduino.cc/resources?id=32372703>

[4] <https://ctc101.arduino.cc/resources?id=29415213>

[5] <https://ctc101.arduino.cc/resources?id=31800414>

[6] <https://ctc101.arduino.cc/ctc101/module/module-2/lesson/beep>

[7] <https://ctc101.arduino.cc/resources?id=32379590>

[8] <https://ctc101.arduino.cc/resources?id=32150446>

Darba lapa „LED izpēte”

1. uzdevums. Gaismas mirgošanas eksperimenta rezultātu apraksts

No.	Parametra vērtība pirmajā delay() funkcijā	Parametra vērtība otrajā delay() funkcijā	Novērotā rezultāta skaidrojums
piemērs	1000	1000	LED mirgo ar 2 sekunžu intervālu
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Darba lapa „Runātāja iepazīšana”

2. uzdevums. Īsas melodijas ģenerēšana

Ģenerējiet norādīto melodiju, izmantojot piecas notis, piemēram, Do-Mi-Sol-Mi-Do. Uzrakstiet kodu, izmantojot funkciju `tone()` un 2.1. attēlā sniegto skaidrojumu par toņu frekvencēm.



TONE		FREQUENCY (in hertz)		DELAY VALUE
Do (C)	→	261.63	→	1915 μs
Re (D)	→	293.66	→	1700 μs
Mi (E)	→	329.63	→	1519 μs
Fa (F)	→	349.23	→	1432 μs
Sol (G)	→	392.00	→	1275 μs
La (A)	→	440.00	→	1136 μs
Si (B)	→	493.88	→	1014 μs

2.1. attēls. Tona frekvences izskaidrojums

Piemēram,

Šajā piemērā mēs izmantojam funkciju `tone()`, lai radītu melodiju no trim notīm Do-Fa-La, katru noti atskaņojot 2 sekundes.

```
void setup()  
{  
    tone(8, 261.63, 2000);  
    tone(8, 349.23, 2000);  
    tone(8, 440, 2000);  
}  
void loop() { }
```

Darba lapa „EducationShield bibliotēkas izpēte”

3. uzdevums. EducationShield bibliotēkas funkciju atpazīšana
Saskaņojiet katru funkciju vai operatoru ar pareizo aprakstu.

<code>VUMeter</code>	turns LEDs on from the first to length (Ieslēdz LED diodes no pirmās līdz norādītajam skaitam)
<code>config(int length, int pins)</code>	turns all LEDs off (Izslēdz visas LED diodes)
<code>begin()</code>	turns on LEDs from startIndex to stopIndex (Ieslēdz LED diodes no startIndex līdz stopIndex)
<code>clear()</code>	creates a VUMeter object (Izveido VUMeter objektu)
<code>on(int index)</code>	stops the program for a certain amount of time (Aptur programmu uz noteiktu laiku)
<code>scrollRight(int speed,int startIndex)</code>	blinks all LEDs (Mīrgo visas LED diodes)
<code>blink(int index, int speed,int times)</code>	initializes the the VU-meter (Inicilizē VU-meter)
<code>off(int index)</code>	turns on one LED at a time from left to right(Ieslēdz pa vienai LED diodei no kreisās puses uz labo)

<code>fillFrom(int startIndex, int stopIndex)</code>	Allows to configure specific PIN as an output using function pinMode (Ļauj konfigurēt konkrētu PIN kā izeju, izmantojot funkciju pinMode()).
<code>fill(int length)</code>	configures the VUmeter (Konfigurē VUmeter)
<code>setup()</code>	repeats given block of code (Atkārtο norādīto koda bloku)
<code>digitalWrite(pinNumber , HIGH/LOW)</code>	turns one LED on (Ieslēdz vienu LED)
<code>blinkAll(int speed,int times)</code>	blinks one LED (Mirgo ar vienu LED)
<code>delay(int time)</code>	turns one LED off (Izslēdz vienu LED)
<code>loop()</code>	writes HIGH or LOW to the digital pin pinNumber (Raksta HIGH vai LOW uz digitālo pinu pinNumber)

5.3. modulis. Projekta izstrāde: nelielas elektroniskas spēles, kas simulē sporta spēles, izveide un spēlēšana

Sasniedzamais rezultāts:

Līdz nodarbības beigām skolēni spēs saprast, kā izstrādāt savu projektu, kurā tiek reģistrēti rezultāti, izmantojot gaismas sensoru, fototranzistoru, gaismas diodes, pjezo skaļruni vai citu sensoru; apkopot un prezentēt projekta rezultātus, kā arī strādāt gan individuāli, gan grupā.

Materiāli:

1. Projekts „Basketbols”

- 1 x vadības plate
- 1 x Arduino Education Shield
- 1 x gaismas sensora modulis
- 1 x pjezo skaļrunis
- 5 x LED
- 5 x 220 omu rezistors
- 1 x moduļa kabelis
- 12 x savienojuma vads

Citi materiāli:

- 1 x galda tenisa bumbiņa
- 1 x līmlente
- 1 x šķēres
- 1 x plastmasas glāze (var būt nepieciešamas vairākas, ja kāda saplīst)

2. Project “React”

- 1 x vadības plate Arduino 101
- 1 x Arduino Education Shield
- 1 x eksperimentu plate
- 1 x pjezo skaļrunis
- 3 x LED
- 3 x 220 omu rezistors
- 3 x 1 Mohm rezistors
- 18 x savienojuma vads (3 gari)

Citi materiāli:

- 3 x alumīnija folijas gabals
- 1 x līmlentes rullis

Nodarbības plāns:

1. Atkārtojiet vienu no projektiem, izmantojot norādījumus, kas sniegti [1 vai 2], strādājot individuāli vai pāros (15 minūtes).
2. Skolēni veic izmaiņas paraugprojektā, t. i., izveido izvēlētās spēles citu variāciju (25 minūtes). Zemāk ir sniegtas dažas idejas izmaiņām.

Projekts “Basketbols”

- pievieno vairāk gaismas diožu un palielini maksimālo punktu skaitu
- Izveido savu melodiju

Projekts “React”

- maini reakcijas laiku un gaidīšanas laiku
- Izveido savu melodiju

- maini gaismas diožu mirgošanas veidu, kad spēle ir beigusies
 - maini gaismas diožu mirgošanas veidu, kad spēle ir beigusies
 - izmēģini pievienot vairāk gaismas diožu un sensoru
3. Skolēni demonstrē savus projektus un izskaidro veiktās izmaiņas (40 minūtes).
 4. Skolēni izstrādā savu projektu, apraksta projektu un tā rezultātu, izmantojot 1. veidni „Projekta apraksts“, strādājot individuāli vai pāros. Ļaujiet skolēniem izvēlēties jebkuras detaļas (sensorus utt.) no Arduino CTC 101 komplekta, izdomāt ideju savam projektam un to īstenot (50 minūtes).
 5. Skolēni prezentē savus projektus (30 minūtes).

Darba lapa „Projekta veidne”

Projekta nosaukums	Projekta nosaukums
Projekta autors	Studenta vārds un uzvārds
Izveides datums	DD.MM.GGGG
Pārskatīšanas datums	DD.MM.GGGG
Vērtējums	Punkti vai %



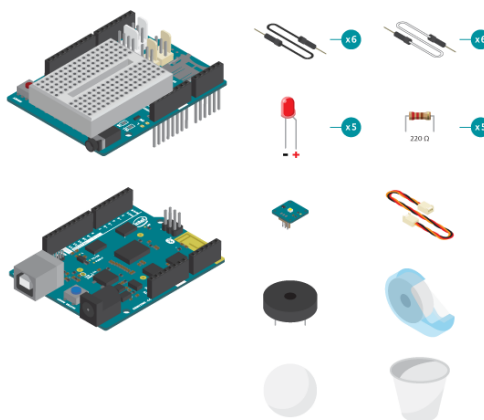
PROJEKTA IDEJAS APRAKSTS

Piemēram,

Šajā spēlē spēlētāji mēģinās iemest pingponga bumbiņu kausā. Lai uzvarētu, jāiegūst pieci punkti. Rezultāts tiek uzskaitīts, izmantojot gaismas sensoru, konkrētāk – fototranzistoru.

MATERIĀLI

Piezīme: papildus students var pievienot komponentu attēlus.



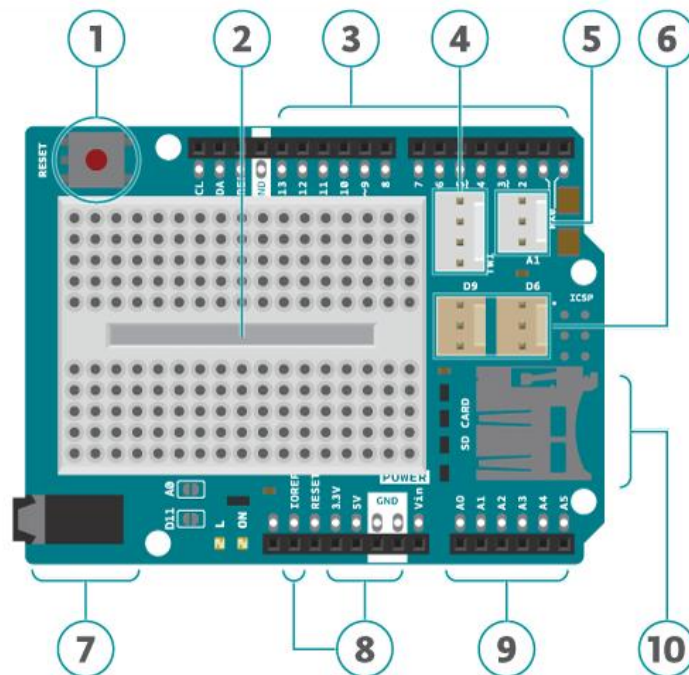
- 1 x vadības plate
- 1 x Arduino Education Shield
- 1 x gaismas sensora modulis
- 1 x pjezo skaļrunis
- 5 x LED
- 5 x 220 omu rezistors
- 1 x moduļa kabelis
- 12 x savienojuma vads

Citi materiāli:

- 1 x galda tenisa bumbiņa
- 1 x līmlente
- 1 x šķēres
- 1 x plastmasas glāze (var būt nepieciešamas vairākas, ja kāda saplīst)

INSTRUKCIJAS

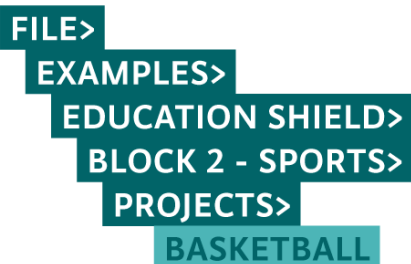
1. Pievienojiet paplašinājuma plati plātnes augšpusē.
2. Pievienojiet piecus gaismas diodus (LED) uz eksperimentālās plātnes atstarpes.
3. Pievienojiet 220 omu rezistoru digitālajam kontaktam Nr. 2. Pēc tam pievienojiet rezistoru pirmā gaismas dioda garajai kājiņai.
4. Pēc tās pašas metodes pievienojiet katru no digitālajiem kontaktiem Nr. 3 līdz Nr. 6 atbilstošajam gaismas diodam.
5. Pievienojiet gaismas diodu īsās kājiņas GND.
6. Pieslēdziet pjezo skaļruni pāri eksperimentu plāksnes spraugai un vienu kājiņu pieslēdziet digitālajam kontaktam 8 (nav svarīgi, kuram), tad otru – GND.
7. Pieslēdziet gaismas sensora moduli analogā moduļa savienotājam A1 uz aizsargplāksnes.
8. Izmantojiet moduļa kabeli, lai pieslēgtu moduli aizsargplāksnei.
9. Paņemiet līmlenti, šķēres un plastmasas glāzi.
10. Izgrieziet caurumu plastmasas glāzes apakšā, lai tajā ietilptu gaismas sensors.
11. Ievietojiet gaismas sensoru caurumā.
12. Pielīmējiet sensoru pie kausa.
13. Pārlicinieties, ka sensors ir redzams caur kausa caurumu (skatīt shēmu).



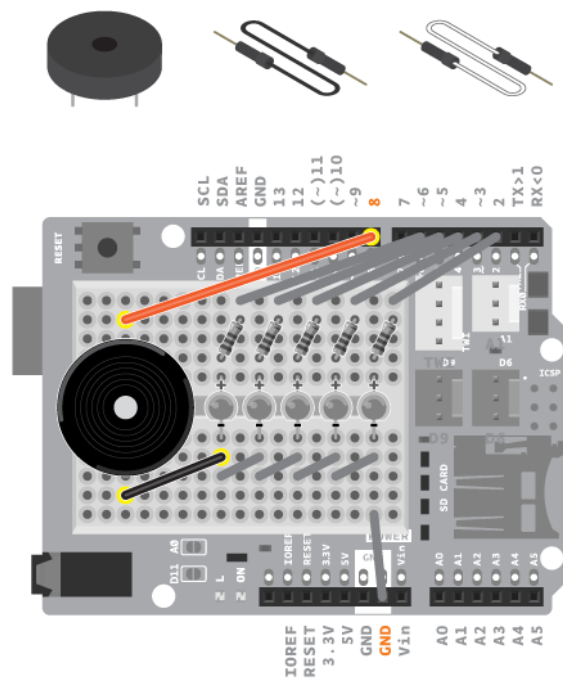
Shēmas paraugs

Piezīme: Skolēniem ar krāsainiem marķieriem vai zīmuļiem uz tāfeles jāuzzīmē komponenti, norādot, kur un kā tie jāsavieno.

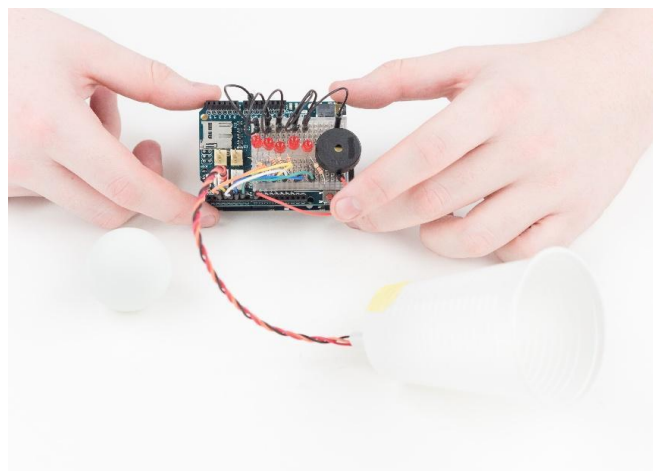
14. Pārbaudiet, vai vadu savienojumi ir gatavi, un pieslēdziet plati datoram.
15. Atrodiet programmas „LightSensorTest“ paraugu un atveriet to. Šo programmu varat izmantot, lai kalibrētu gaismas sensoru un iestatītu sliekšņa vērtību.
16. Augšupielādējiet programmu uz plati.
17. Atveriet sadaļu „Sērijas komunikācijas rokasgrāmata“ un iepazīstieties ar tās darbības principiem. Pēc tam atveriet sērijas monitoru.
18. Pārbaudiet sensora vērtību, kad bumbiņa nav kausā, un pierakstiet to.
19. Ielieciet pingponga bumbiņu kausā virs gaismas sensora. Pierakstiet vērtību, kad sensora ir aizklāts. Aprēķiniet abu skaitļu (kad sensora ir atklāts un kad aizklāts) vidējo vērtību kā sliekšņa vērtību.
20. Tagad, kad zināt vērtību, kad sensora ir atklāts, un vidējo vērtību, pārbaudiet, vai vadu savienojumi ir gatavi un pieslēgti datoram.
21. Atrodiet programmu Basketball un atveriet to.



22. Atrodiet sintakses rindu `sensor.config( uncovered , threshold )` un mainiet parametru vērtības, attiecīgi aizstājot tās ar `uncovered` un `threshold` vērtībām. Pēc tam augšupielādējiet programmu uz plati.
23. Sāciet spēlēt basketbolu.



Scheme: Scheme piemērs
vai



Attēls: Plates attēls

KODS

Ievadi programmas kodu

6. modulis. Analogo signālu un seriālās komunikācijas pamati

Sasniedzamais rezultāts: Izprast analogu signālu nolasīšanu un rakstīšanu. Apgūt sērijas komunikācijas pamatus, lai savienotu vadības plati ar datoru un apmainītos ar datiem.

Pārskats:

- 3.1. modulis. Analogu signālu nolasīšana un rakstīšana, kā arī analogu sensoru izmantošana robotu programmēšanā.
- 3.2. modulis. Sērijveida komunikācija starp vadības plati un datoru datu sūtīšanai un saņemšanai
- 3.3. modulis. Projekta izstrāde: elementa izveide, kas simulē kādu procesu (mūzikas ģenerēšana, ziņojuma parādīšana, ritmu atskaņošana)

Laiks/stundas: 6 (3 divkārtas nodarbības – 2x40 min)

6.1. modulis. Analogo signālu nolasīšana un rakstīšana, kā arī analogu sensoru izmantošana robotu programmēšanā

Sasniedzamais rezultāts: Līdz nodarbības beigām skolēni spēs izprast analogos signālus un veidot vienkāršus projektus, nolasīt un reģistrēt analogos signālus, kā arī izmērīt dažus parametrus, izmantojot dažādus sensorus.

Materiāli:

- 101 vadības plate
- 1 x Education Shield
- 1 x potenciometrs
- 3 x savienojuma vadi
- 1 x LED
- 1 x 220 omu rezistors
- Projektors vai tāfele skolotāja demonstrācijai

Nodarbības plāns:

1. Izskaidrojiet analogā signāla jēdzienu, miniet analogu sensoru piemērus (piemēram, skaņas sensori, gaismas sensori, temperatūras sensori, mitruma sensori, spiediena sensori, tuvuma, krāsu, gāzes, starojuma sensori utt.) un reālus piemērus par analogu sensoru izmantošanu mērījumiem [1, 5] (5 minūtes).
2. Izskaidrojiet studentiem potenciometru [2] un tā pielietojumu, piemēram, skaļuma regulētāju, gaismas dimmeru slēdžus, televizoru spilgtuma regulētājus un audio iekārtu faderus (5 minūtes).
3. Izskaidrojiet studentiem, kādas sastāvdaļas ir nepieciešamas praktiskajai darbībai [1, 3.1. piemērs] (2 minūtes).

4. Sniegt studentiem norādījumus, kā potenciometru pieslēgt vadības platē un kādas komandas izmantot signāla nolasīšanai. Izskaidrojiet katru koda rindu no [1, 3.1. piemērs] (3 minūtes).

5. Lūdziet studentiem izpildīt 3.1. piemēru, ievērojot norādījumus (15 minūtes).

6. Izskaidrojiet studentiem, kā rakstīt analogos signālus. Izskaidrojiet, kādas detaļas ir nepieciešamas praktiskajam uzdevumam [3, 3.2. piemērs] un piemēra kodam (5 minūtes).

7. Lūdziet studentiem izpildīt 3.2. piemēru, ievērojot sniegtos norādījumus (15 minūtes).

8. Lūdziet studentiem modificēt kodu, lai (15 minūtes):

- mainītu “elpošanas” ātrumu tā, lai tas izdziestu ātrāk/lēnāk;

- pievienotu potenciometru un izmantotu to LED vadībai;

- izpildītu 3.3. un 3.4. piemēru [4].

8. Pārdomās (10 minūtes)

– Lūdziet studentiem atbildēt uz šādiem jautājumiem:

- Miniet reālus analogu signālu piemērus.
- Miniet analogu sensoru piemērus.
- Kādi kontakti izmanto funkcijas `analogread()` un `analogwrite()`?
- Kādas vērtības (min, max) nolasa funkcija `analogread()`?
- Kādas vērtības (min, max) nolasa funkcija `analogwrite()`?

– Studenti apkopojiet to, ko esat iemācījušies, un dalieties ar personīgajām atziņām vai priekšrocībām attiecībā uz analogajiem sensoriem un potenciometru lietojumiem.

Avoti:

[1] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/lesson/reading-analog-signals>

[2] <https://ctc101-dev.arduino.cc/resources?id=32151800>

[3] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/lesson/writing-analog-signals>

[4] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/lesson/light-sensor>

[5] <https://components101.com/article/different-types-of-sensors-and-sensing-technologies>

6.2. modulis. Sērijveida datu pārraide starp vadības plati un datoru datu nosūtīšanai un saņemšanai

Sasniedzamais rezultāts: Līdz nodarbības beigām skolēni spēs izprast sērijas komunikāciju, kā arī to, kā savienot vadības plati ar datoru, nosūtīt datus no vadības plates uz datoru un saņemt datus no datora uz vadības plati.

Materiāli:

- 1 x Education Shield
- 1 x USB kabelis
- 1 x gaismas sensora modulis
- 1 x moduļa kabelis
- Projektors vai tāfele skolotāja demonstrācijai

Nodarbības plāns:

1. Izskaidrojiet skolēniem, kā vadības plate sazinās ar datoru, un parādiet, kā to pieslēgt datoram [1]. Izskaidrojiet un parādiet vadības plates kontaktus, kurus izmanto datu nosūtīšanai uz vadības plati un saņemšanai no tās [2, 3.5. piemērs] (5 minūtes).
2. Parādiet studentiem piemēru, kā nosūtīt teksta ziņojumu uz seriālo portu [2, piemērs 3.5.], un izskaidrojiet kodu. Mainiet teksta ziņojumu. (7 minūtes)
3. Pieslēdziet gaismas sensora moduli. Parādiet paraugprogrammu, kā nolasīt gaismas sensora vērtības [2, piemērs 3.6.]. Izskaidrojiet katru koda rindu (8 minūtes).
4. Lūdziet skolēniem izpildīt 1. uzdevumu, sekojot norādījumiem [2, 3.6. piemērs]. Rezultātus ievadiet tabulā. (20 minūtes)
5. Izskaidrojiet komandas, kas tiek izmantotas datu saņemšanai caur seriālo portu. Parādiet piemēru [3, 3.7. piemērs], izskaidrojot visas koda rindas (10 minūtes).
6. Lūdziet studentiem pārbaudīt jebkuru no piemēriem, kas atrodas File -> Examples -> EducationShield-> Help, kur tiek izmantots seriālais ports. (35 minūtes)
9. Refleksija (5 minūtes)
 - Lūdziet studentiem atbildēt uz šādiem jautājumiem:
 - Kādus datus var nosūtīt vai saņemt caur seriālo portu?
 - Cik daudz un kādus kontaktus izmanto seriālais ports?
 - Kādas komandas tiek izmantotas, lai izdrukātu tekstu vai vērtības?
 - Studenti apkopojiet to, ko esat iemācījušies, un dalieties ar personīgajām atziņām vai priekšrocībām attiecībā uz vadības paneļa programmēšanu, ārējo digitālo sensoru (ārējais LED, skaļrunis) nolasīšanu un digitālo aktuatoru vadību.

Avoti:

- [1] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/lesson/serial-port>
- [2] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/lesson/sending-to-computer>
- [3] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/lesson/receiving-from-computer>

Darba lapa „Sērijas komunikācijas izpēte”

1. uzdevums. Gaismas intensitātes mērīšana

1. Izveidojiet projektu, ievērojot norādījumus [1, 3.6. piemērs].
2. Deklarējiet mainīgo, kas glabā sensora rādījuma kartēto vērtību (izmantojiet funkciju map()).
3. Izvadiet abas vērtības uz seriālo monitoru un novērojiet, kā tās mainās, kad jūs mijiedarbojaties ar gaismas sensoru.
4. Veiciet eksperimentus, mērot gaismas intensitāti dažādās vietās/telpās/telpas pusēs.
5. Rezultātus atspoguļojiet sniegtajā tabulā.

Datums, laiks	Vieta	Gaismas sensora rādījums

6.3. modulis. Projekta izstrāde: elementa izveide, kas simulē kādu procesu (mūzikas radišana, ziņojuma parādīšana, ritma atskaņošana)

Sasniedzamais rezultāts: Līdz nodarbības beigām skolēni spēs saprast, kā izstrādāt projektu, ievērojot dotās instrukcijas un izmantojot dažādus sensorus (piemēram, gaismas sensoru, fototranzistoru, gaismas diodes, pjezo skaļruni) un citas detaļas (piemēram, pogu, tīmekļa kameru), strādāt gan individuāli, gan grupās, kā arī prezentēt savu projekta rezultātus.

Materiāli:

1. Projekts „Binary LP”

- 1 x vadības plate
Arduino 101
- 1 x Arduino
Education Shield
- 1 x infrasarkanais
sensoru bloks
- 1 x pjezo skaļrunis
- 7 x savienojuma vadi
(5 garie)
- 1 x „Binary LP”
komplekts

Citi materiāli:

- 1 x „Binary LP”
papīra disks
- 1 x līmlente
- 1 x šķēres
- 1 x printeris

2. Project “Boombox”

- 1 x vadības plate
- 1 x Arduino
Education Shield
- 1 x micro SD karte
- 3 x pogas
- 1 x skaļrunis
- 3 x 10 kOhm rezistori
- 9 x savienojuma vadi

3. Projekts „Cookie monster”

- 1 x vadības plate
Arduino 101
- 1 x Arduino
Education Shield
- 1 x eksperimentu
plate
- 1 x tīmekļa kamera
- 1 x strāvas LED
modulis

4. Projekts “Drawdio”

- 1 x vadības plate
Arduino 101
- 1 x Arduino
Education Shield
- 1 x maizes plate
- 1 x pjezo skaļrunis
- 1 x 1 Moma rezistors
- 5 x savienotājevadi

- 1 x 1 Mohm rezistors
 - 1 x moduļa kabelis
 - 3 x savienojuma vads (1 garš)
 - 1 x Sīkdatņu monstros komplekts
- Citi materiāli:
- 1 x grafīta zīmulis
 - 1 x līmlente

Citi materiāli:

- 1 x alumīnija folijas gabals
- 1 x vadītspējīga sīkdatņu trauks
- 3–4 x gumijas lenta
- 1 x līmlente

5. Projekts “Cnock cnock box”

- 1 x vadības plate Arduino 101
- 1 x Arduino Education Shield
- 1 x eksperimentu plate
- 2 x pjezo skaļrunis
- 1 x 1 MΩ rezistors
- 1 x 9 V baterija
- 1 x barošanas spraudnis
- 1 x 3,3 V Zenera diode
- 6 x savienojuma vadi
- 1 x „Knock Knock” kastes komplekts

Citi materiāli:

- 1 x lenta

7. Project “Sequencer”

- 1 x vadības plate
- 1 x Arduino Education Shield
- 1 x micro SD karte
- 1 x 8 omu skaļrunis
- 5 x 680 omu rezistors

6. Projekts “P.O.V”

- 1 x vadības plate
- 1 x Arduino Education Shield
- 5 x LED
- 5 x 220 omu rezistors
- 10 x savienojuma vads
- 1 x 9 V baterija
- 1 x barošanas spraudnis

- 1 x 220 omu rezistors
- 1 x 470 omu rezistors
- 1 x 1,2 kOhm rezistors
- 9 x savienojuma vads
- 9 x atsevišķs vads
- 1 x sekvenceru komplekts

Nodarbības plāns:

1. Atkārtojiet vienu no projektiem, izmantojot piegādātos komplektus un norādījumus, kas sniegti [1–7], strādājot individuāli, pāros vai lielākās grupās (30 minūtes).
2. Skolēni pielāgo paraugprojekta kodu (25 minūtes).
3. Skolēni demonstrē savus projektus un izskaidro veiktās izmaiņas (25 minūtes).

Avoti:

- [1] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/binary-lp>
- [2] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/boombox>
- [3] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/cookie-monster>
- [4] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/drawdio>
- [5] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/knock-knock-box>
- [6] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/pov>
- [7] <https://ctc101-dev.arduino.cc/ctc101/module/module-3/project/sequencer>

7. modulis. Darbs ar motoriem

Sasniedzamais rezultāts: Izprast un pielietot motoru izmantošanas principus ar Arduino CTC 101, tostarp standarta un nepārtrauktas servo vadības principus.

Programmas saturs:

Iepazīties ar dažādiem motoru veidiem un to funkcijām mehānikas un elektronikas sistēmās. Iemācīties, kā pieslēgt, vadīt un programmēt standarta un nepārtrauktas darbības servodzinējus. Izveidot ieejas vadītu kustību, izmantojot sensorus un mainīgos lielumus. Projektēt un izgatavot nelielu robotu vai kustības balstītu projektu. Piemērot radošumu un komandas darbu noslēguma uzdevumā vai konkursā, kurā iesaistīti vairāki servodzinēji. Laiks/stundas: 6 vai 8 (3 vai 4 dubultstundas – 2x40 min).

7.1. modulis. Dzinēju veidi un standarta servodzinēji

Mērķis: Iepazīties ar galvenajiem motoru veidiem un apgūt standarta servomotora pieslēgšanu un vadību, izmantojot Arduino.

Sasniedzamais rezultāts:

- Nodarbības beigās studenti spēs:
- Atpazīt un aprakstīt galvenos motoru veidus (līdzstrāvas, servo, soļotājus).
- Izskaidrot, kā darbojas standarta servodzinējs.
- Pieslēgt un vadīt standarta servodzinēju, izmantojot Arduino kodu.
- Novērot, kā servodzinēja pozīcija atbilst koda vērtībām.

Nepieciešamie materiāli:

- Arduino platine un USB kabelis
- Servodzinējs (standarta)
- Savienojuma vadi
- Mēģinājumu plate
- Portatīvais dators ar instalētu Arduino IDE
- Darba lapa „Servodzinēju izpēte”
- Projektors vai tāfele skolotāja demonstrācijai

Nodarbības plāns:

1. Skolotājs izskaidro dažādu veidu motorus (līdzstrāvas, servo, soļotājus), izmantojot attēlus vai īsus video. Diskusija: Kur mēs ikdienā sastopamies ar motoriem? [1] (15 minūtes)
2. Skolotājs izskaidro, ar ko servo motors atšķiras: to kontrolē pēc pozīcijas, nevis ātruma. Iepazīstina ar galvenajiem terminiem: impulsa platuma modulācija (PWM), leņķa diapazons (0–180°). [2] (10 minūtes)
3. Skolotājs parāda, kā pieslēgt servodzinēju pie Arduino (Signāls–Strāva–Zeme) [2]. Skolēni pāros atkārtu uzstādīšanu. (10 minūtes)
4. Skolotājs parāda koda piemēru, izmantojot Servo bibliotēku [2][3]: (15 minūtes)

```
#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9);
}

void loop() {
  myServo.write(0);
  delay(1000);
  myServo.write(90);
  delay(1000);
  myServo.write(180);
  delay(1000);
}
```

Apspriediet, kas notiek un kāpēc servodzinējs apstājas.

1. Skolēni pielāgo kodu, lai izveidotu savu kustības modeli. Skolotājs veicina radošumu un pārbauda savienojumus. (20 minūtes)
2. Skolēni atbild uz pārdomu jautājumiem darba lapā „Servodzinēju izpēte”. (10 minūtes)
3. Pārdomas (5 minūtes):
4. Pirms nodarbības beigām katrs skolēns atbild uz jautājumu: Kādi ir divi servodzinēju pielietojumi reālajā dzīvē?

Papildus idejas:

- Izveidojiet servo, kas vienmērīgi pārvietojas no viena leņķa uz otru.
- Pievienojiet LED indikatorus, kas parāda servo pašreizējo virzienu.
- Izpētiet, kur tiek izmantoti soļdzinēji un kā tie atšķiras no servodzinējiem.

Avoti:

- [1] <https://ctc101.arduino.cc/ctc101/module/module-4/lesson/types-of-motors>
- [2] <https://ctc101.arduino.cc/ctc101/module/module-4/lesson/standard-servo>
- [3] <https://ctc101.arduino.cc/ctc101/module/module-4/lesson/continuous-rotation-servo>

Darba lapa “Servo motoru izpēte”

Uzziniet, kā ar Arduino palīdzību pieslēgt un vadīt standarta servodzinēju.

1. uzdevums. Dzinēju tipu identificēšana

Nosauciet trīs izplatītākos dzinēju tipus un to galveno pielietojumu:

Motora tips	Motora tips
1	
2	
3	

2. uzdevums. Servodzinēja pieslēgšana

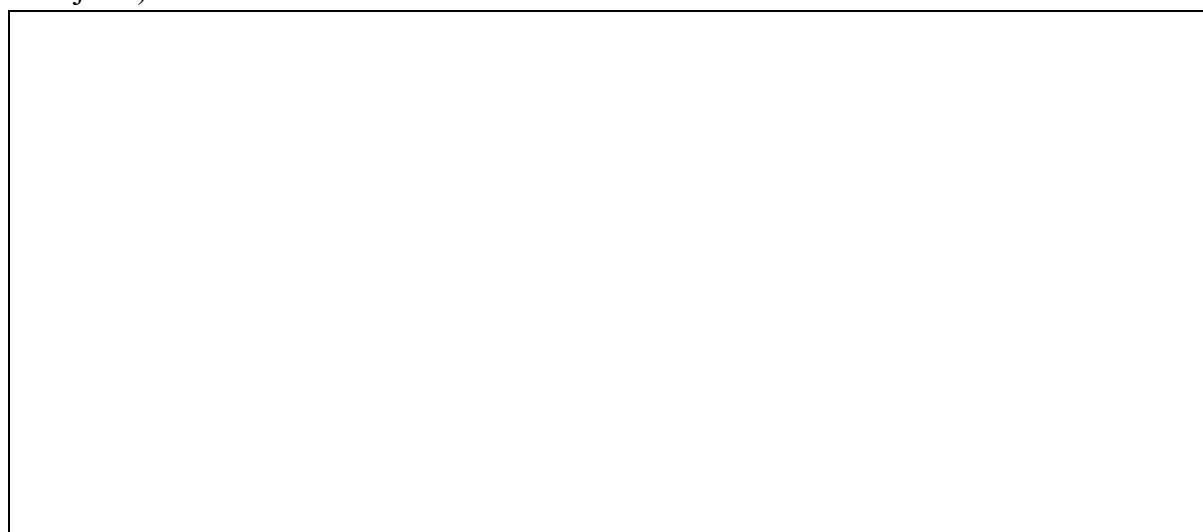
Uzziniet, kā pareizi pieslēgt standarta servodzinēju pie Arduino platītes.

Norādījumi:

1. Izņemiet vienu standarta servodzinēju no savas komplektācijas.
2. Noteikt trīs vadus:
3. Oranžs/dzeltens -> Signāls
4. Sarkans -> Barošana (+5V)
5. Brūns/melns -> Zeme (GND)
6. Izmantojiet savienojuma vadus, lai pieslēgtu servodzinēju pie Arduino platītes:

Servo vads	Pieslēdzas pie Arduino	Apraksts
Signāls (oranžs/dzeltens)	Pin 9	Nosūta vadības signālu
Barošana (sarkans)	5V	Nodrošina barošanu
Zeme (brūns/melns)	GND	Noslēdz ķēdi

Zemāk uzzīmējiet savu shēmu un skaidri apzīmējiet visus savienojumus (9. kontakts, 5 V, zemējums):



Ja tev ir krāsainie zīmuli, izmanto vadus atbilstošās krāsās.

3. uzdevums. Servo vadība

Izpēti parauga kodu un zemāk uzraksti savu versiju.

```
#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9);
}

void loop() {
  myServo.write(0);
  delay(1000);
  myServo.write(90);
  delay(1000);
  myServo.write(180);
  delay(1000);
}
```

Pielāgojiet kodu, mainot elementu izvietojumu vai sinhronizāciju.

4. uzdevums. Novērošana

Izpētiet parauga kodu un zemāk uzrakstiet savu versiju.

Palaidiet savu kodu.

Aprakstiet, kas notiek:

7.2. modulis. Nepārtraukta rotācija un ieejas vadība

Mērķis: Izmantot sensorus un servodzinējus, lai izveidotu interaktīvu mini-projektu – robotizētu roku, kas pamāj vai reaģē uz lietotāja komandām.

Sasniedzamais rezultāts:

- Nodarbības beigās skolēni spēs:
- Pieslēgt un vadīt servodzinēju, izmantojot ieejas signālus no potenciometra vai gaismas sensora.
- Saprast, kā analogās ieejas vērtības ietekmē motora darbību.
- Apvienot programmēšanu, sensorus un aktuatorus nelielā, funkcionējošā projektā.
- Praktizēt strukturētu kodēšanu un kļūdu novēršanu.

Materiāli:

- Arduino CTC 101 komplekts
- Savienojuma vadi un eksperimentu plāksnes
- Datori ar instalētu Processing IDE un Python vidi
- Darba lapa „Ievades vadības servo uzdevums”
- Projektors vai tāfele skolotāja demonstrācijai

Nodarbības plāns:

1. Skolotājs iepazīstina ar interaktīvo sistēmu jēdzienu – robotiem, kas reaģē uz apkārtējo vidi. Piemēri: automātiskās durvis, robotizētie mājdzīvnieki, viedās rotaļlietas.[1][2][3][4][5] (5 minūtes)
2. Skolotājs parāda iepriekš sagatavotu piemēru – „viļājošu roku”, kas izgatavota no servodzinēja, piestiprināta pie kartona izgriezuma un kustas atbilstoši potenciometra ievades signālam. Izskaidro, ka servodzinēja pozīcija (0° – 180°) mainās atkarībā no sensora ievades vērtības (0–1023). [6][7] (10 minūtes)
3. Skolēni pieslēdz potenciometru pie A0 un servo pie 9. pines saskaņā ar skolotāja piemēru. Viņi pārbauda vadu savienojumus, izmantojot darba lapu „Ievades vadīta servo uzdevums”. (15 minūtes).
4. Skolotājs palīdz skolēniem uzrakstīt kodu, lai nolasītu potenciometra rādījumus un vadītu servo. Skolēni ievada un pārbauda šo piemēru:

```
from Arduino import Arduino
import time

board = Arduino("9600")

pot_pin = 0    # Analog input A0
servo_pin = 9  # Digital pin 9

board.pinMode(servo_pin, "SERVO")

while True:
```

```

val = board.analogRead(pot_pin)
angle = int((val / 1023) * 180)
board.analogWrite(servo_pin, angle)
time.sleep(0.05)

```

5. Skolotājs paskaidro, kā analogā ieeja tiek piesaistīta servo lenķim (0–180°). (20 minūtes)
6. Skolēni izrotā savu servo ar vienkāršām kartona rokām vai figūrām, lai tas varētu pamāt, pamāt ar galvu vai pagriezties. Rosiniet radošumu (piemēram, pamājošs robots, kaķa aste, kustīgs karogs). (15 minūtes)
7. Brīvprātīgie demonstrē savus interaktīvos robotus. Skolotājs vada diskusiju: Kādas izmaiņas varētu padarīt kustības vienmērīgākas vai reaģētspējīgākas? (10 minūtes)
8. Pārdomu brīdis (5 minūtes):
Skolēni uzraksta vienu īsu teikumu par to, kas viņiem šķita visgrūtākais: vadu savienošana, programmēšana vai vērtību kartēšana, un paskaidro, kāpēc.

Papildus idejas:

- Aizstājiert potenciometru ar gaismas sensoru, lai izveidotu „gaismas reaģējošu viļājošu robotu”.
- Pievienojiet divus servomehānismus divām rokām, kas viļājas pārmaiņus.
- Vadiet servomehānismu ar Processing interfeisa slideriem uz ekrāna (pieredzējušiem skolēniem).

Avoti:

- [1] <https://ctc101.arduino.cc/ctc101/module/module-4/project/camera-robot>
- [2] <https://ctc101.arduino.cc/ctc101/module/module-4/project/light-chaser>
- [3] <https://ctc101.arduino.cc/ctc101/module/module-4/project/line-follower>
- [4] <https://ctc101.arduino.cc/ctc101/module/module-4/project/magic-box>
- [5] <https://ctc101.arduino.cc/ctc101/module/module-4/project/tickle-robot>
- [6] <https://ctc101.arduino.cc/ctc101/module/module-4/lesson/input-controlled-servo>
- [7] <https://www.instructables.com/Waving-Arm/>

Darba lapa “Ievades vadības servodzinēju izaicinājums”

Izveidojiet vienkāršu interaktīvu servo projektu, kas reaģē uz lietotāja ievadi (piemēram, potenciometru), lai robotizētā roka pamātu vai kustētos.

1. uzdevums. Izveidojiet shēmu

Pareizi pieslēdziet servodzinēju un potenciometru pie jūsu Arduino.

Norādījumi:

1. Izmantojiet vienu standarta servo un vienu potenciometru no jūsu Arduino CTC 101 komplekta.
2. Veiciet šādus savienojumus:

Komponents	Pieslēgums uz Arduino	Apraksts
Servo signāls (oranža/dzeltena)	Digital 9	Vadības signāls
Servo barošana (sarkana)	5 V	Barošanas avots
Servo masa (brūna/melnā)	GND	Kopējais zemējums
Potentiometra vidējais kontakts	Analog A0	Analogās vērtības nolasīšana
Potentiometra sānu kontakts 1	5 V	Sprieguma ieeja
Potentiometra sānu kontakts 2	GROUND	Zemējums

Zemāk uzzīmējiet savu shēmu un skaidri apzīmējiet visus vadus:

Pārbaudi:

- ☐ Visi servo un potenciometra vadi ir pareizi pieslēgti.
- ☐ Pieslēgums strāvai un zemējumam ir pareizs.
- ☐ Servo ir pieslēgts PWM kontaktam (9 vai 10).

2. uzdevums. Uzraksti kodu

Panāc, lai servo kustētos atbilstoši potenciometra stāvoklim.

Koda piemērs:

```
| from Arduino import Arduino
```

```

import time

board = Arduino("9600")

pot_pin = 0      # Analog A0
servo_pin = 9    # Digital 9

board.pinMode(servo_pin, "SERVO")

while True:
    val = board.analogRead(pot_pin)      # Read potentiometer (0-1023)
    angle = int((val / 1023) * 180)      # Convert to 0-180 degrees
    board.analogWrite(servo_pin, angle) # Move servo
    time.sleep(0.05)

```

Ierakstiet savu rediģēto vai pielāgoto koda versiju.

3. uzdevums. Pielāgojiet savu robota roku

Piešķiriet savam servomotoram nedaudz individualitātes. Idejas:

- Pievienojiet kartona roku vai karodziņu pie servomotora kronšteina.
- Izrotājiet to kā robotu, kaķa asti vai vicinātu roku.
- Izmantojiet marķierus, papīru vai līmlenti. Esiet radoši!

Zemāk uzzīmējiet vai aprakstiet savu darbu:

4. uzdevums. Novērošana un pārdomas

Pārbaudi savu uzstādījumu:

1. Lēnām pagriez potenciometru.
2. Novēro servo kustību.

Uzraksti atbildi. Kas notiek, kad potenciometrs atrodas minimālajā/maksimālajā stāvoklī?

7.3. modulis. Rāpojošo radību konkurss

Mērķis: Izstrādāt, izgatavot un programmēt rāpojošu robota radību, izmantojot servodzinējus. Strādāt komandās, lai testētu, uzlabotu un sacenstos, kuras radība rāpo vistālāk vai pārvietojas visrādošāk.

Sasniedzamais rezultāts:

- Līdz nodarbības beigām skolēni spēs:
- Piemērot savas zināšanas par servodzinējiem, lai radītu kustību robotā.
- Eksperimentēt ar mehānisko konstrukciju un materiāliem, lai radītu rāpojošu kustību.
- Rakstīt un testēt Arduino kodu, lai kontrolētu kustības laiku un modeļus.
- Efektīvi sadarboties mazās komandās, lai plānotu, izgatavotu un prezentētu darbojošos prototipu.
- Novērtēt un pilnveidot savus projektus, izmantojot testēšanu un sacensības.

Materiāli:

Arduino CTC 101 komplekts

Savienojuma vadi un eksperimentu plāksnes

Datori ar instalētu Processing IDE un Python režīmu

Kartons, salmiņi, papīra skavas, gumijas siksnas, līmlente utt. (robota ķermenim un ekstremitātēm)

Darba lapa “Rāpojošo radību izaicinājums”

Mērlente vai lineāls (sacensībām)

Taimeris vai hronometrs (var būt tālruna taimeris vai fizisks hronometrs)

Nodarbības plāns:

1. Skolotājs parāda īsus klipus vai attēlus ar vienkāršiem rāpojošiem robotiem un bioloģiskajiem iedvesmas avotiem (piem., kukaiņi, tārpi, krabji). [1][2][3] Skolotājs izskaidro uzdevumu: Katra komanda izstrādās un izgatavos radījumu, kas var pārvietoties uz priekšu, izmantojot vismaz vienu servodzinēju. Nodarbības beigās visi radījumi sacentīsies par balvām attāluma un radošuma kategorijās. (10 minūtes)
2. Skolēni veido komandas (2–3 cilvēki) un apspriež:
 - Kā izskatīsies jūsu radījums?
 - Kā tas pārvietosies? (slīdot, šūpojoties, soļojot, lēkājot)
 - Kādus materiālus jūs izmantosiet?
 - Kur jūs piestiprināsiet servo(s)?
 - Komandas uzskicē savu dizainu un uzskaita materiālus darba lapā “Rāpojošo radību izaicinājums” (1. uzdevums). (10 minūtes)
3. Komandas saliek savu robotu un raksta kodu:
 - Saliekiat savu radību no pieejamajiem materiāliem.
 - Pievienojiet servo(s) pie Arduino.
 - Augšupielādējiet un pārbaudiet kustības kodu.

Piemēram kods:

```
#include <Servo.h>
Servo servo1;
Servo servo2; // optional second servo
```

```

void setup() {
  servo1.attach(9);
  servo2.attach(10); // optional
}

void loop() {
  servo1.write(60);
  delay(500);
  servo1.write(120);
  delay(500);

  servo2.write(120);
  delay(500);
  servo2.write(60);
  delay(500);
}

```

Komandas veic 2. un 3. uzdevumu darba lapā „Rāpojošo radību izaicinājums“. (40 minūtes)

4. Komandas pārbauda savus robotus uz līdzenas virsmas. Skolēni reģistrē rezultātus, nosaka, kas darbojas, un veic uzlabojumus kustībā, līdzsvarā vai konstrukcijā; veic 4. uzdevumu darba lapā „Rāpojošo radību izaicinājums“. (10 minūtes)

5. Katra komanda prezentē savu robotu fināla izaicinājumam. (15 minūtes)

Konkursa kategorijas:

- Attāluma čempions: 10 sekundēs nobrauc garāko attālumu uz priekšu.
 - Vienmērīgākā kustība: visreālākā vai stabilākā kustība.
 - Labākais dizains: visradošākā radības koncepcija un noformējums.
 - Skolotājs vai žūrija mēra attālumu un reģistrē rezultātus. Komandas aizpilda 5. uzdevumu darba lapā „Rāpojošo radību izaicinājums“.
6. Pārdomas (5 minūtes):
- Skolēni raksta par vienu izaicinājumu, ar ko saskārās, veidojot robotu, un to, kā to atrisināja.

Papildinājumu idejas:

- Pievienojiet sensoru (gaismas vai skaņas), lai radījums reaģētu uz apkārtējo vidi.
- Izmantojiet divus servodzinējus, lai nodrošinātu koordinētu ekstremitāšu kustību.
- Izveidojiet radījumu zooloģisko dārzu izstādei vai sacensībām.
- Organizējiet sacensības, kurās tiek mērīts laiks: nosakiet, cik ilgs laiks katram robotam nepieciešams, lai pārvietotos 50 cm taisnā līnijā. Robots, kurš visātrāk sasniedz finiša līniju, iegūst ātruma čempiona titulu. Mudiniet komandas uzlabot kustību efektivitāti, stabilitāti un virziena kontroli.

Avoti:

[1] <https://ctc101.arduino.cc/ctc101/module/module-4/project/crawling-robot>

[2] <https://www.youtube.com/watch?v=uOh2ZY1VHGE>

[3] <https://www.youtube.com/watch?v=S4eQXXxUnNE>

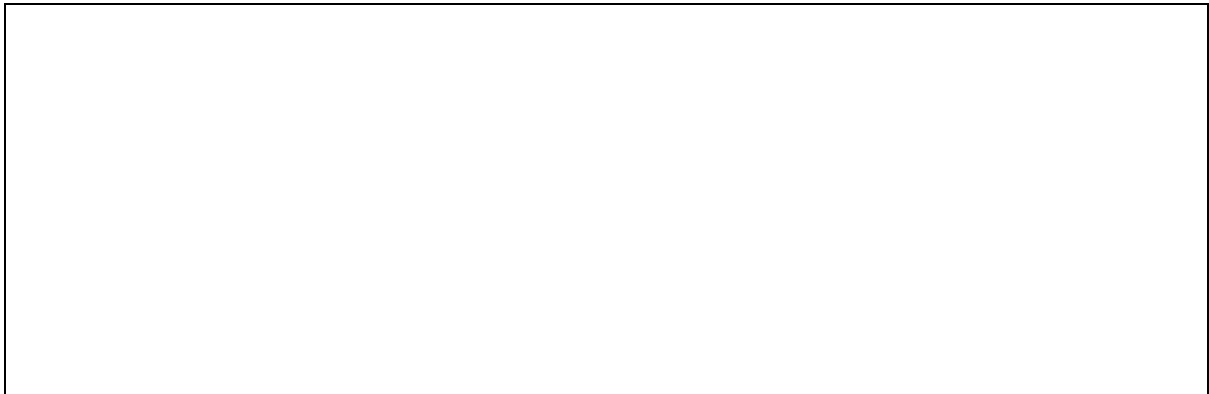
Darba lapa „Rāpojošo radību izaicinājums“

1. uzdevums. Izplānojiet savu radību

Veiciet ideju apmaiņu un izplānojiet savu projektu. Uzzīmējiet savu ideju un atbildiet uz šādiem jautājumiem:

- Kādu radību jūs veidojat?
- Kā tā pārvietosies?
- Kuras detaļas pārvietosies, izmantojot servodzinējus?

Zemāk uzzīmējiet un apzīmējiet savas detaļas:

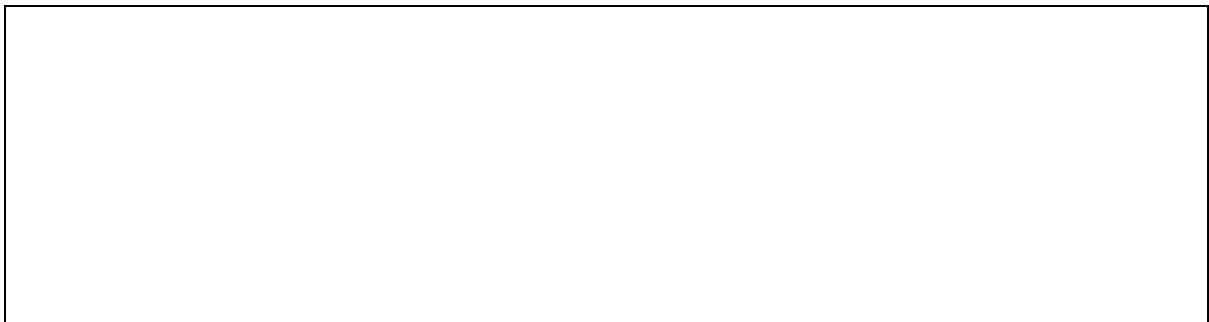


2. uzdevums. Sastāvdaļu saraksts

Komponents	Daudzums	Funkcija
Servo		
Vadi		
Mācību plate		

3. uzdevums. Montāža un programmēšana

Izmantojiet iepriekšējās nodarbībās apgūtās zināšanas, lai pieslēgtu servodzinējus un izstrādātu kustības kodu. Uzrakstiet kustības kodu:



4. uzdevums. Testēšana un uzlabošana

Pēc testēšanas:

- Kas izdevās labi? _____
- Ko tu mainīji, lai uzlabotu savu robotu? _____

5. uzdevums. Sacensību rezultāti

Izmēģinājuma brauciens: Attālums 10 sekundēs = _____ cm

☐ Virzījās uz priekšu;

☐ Griezās apļos;

☐ Apgāzās;

☐ Cits _____

Atbildi uz jautājumu: Kas nākamreiz palīdzētu tavam radījumam kļūt ātrākam vai stabilākam?
