

Scratch

Õppematerjalid

Kestvus – 8 x 1,5 tundi

Raskusaste



Näpunäited õpetjale

Materjal on koostatud selliselt, et esimesel lehel on ülesande kirjeldus ja näide oodatavast tulemusest. Näpunäidete lehel on toodud ülesande lahenduse kirjeldus ja programmi näidis. Üldjuhul sisaldab ülesanne mitut analoogset probleemi natuke erineva nurga alt. Programmi näide on ainult ühe ülesande kohta, ülejäänud analoogjate kohta antakse vihjeid.

Ülesanded 2-6 eeldavad eelnevate ülesannete lahenduste olemasolu. Juhul kui õpilane ei ole neid teinud, tuleks neile anda eelmise ülesande lahendusfail.

Ülesanded 7 ja 8 on oma olemuselt kõige keerulisemad ning eeldavad juba suuremaid teadmisi. Antud ülesandeid võib teha ka väiksemate eelteadmistega, aga siis tuleb arvestada, et ülesande lahendamiseks kulub tõenäoliselt rohkem aega.

Materjaliga on kaasas lahendusfailid:

1. Taustade joonistamine ja spraidi liigutamine ★
2. Teerajal püsimine ★★
3. Teadete saatmine ★★
4. Muutujate kasutamine ★★
5. Kiiruse muutmine, automaatne liikumine ★★★
6. Õhupallimäng ★★★
7. Looduses jalutamine ★★★★★
8. Takistuste vältimine ★★★★★

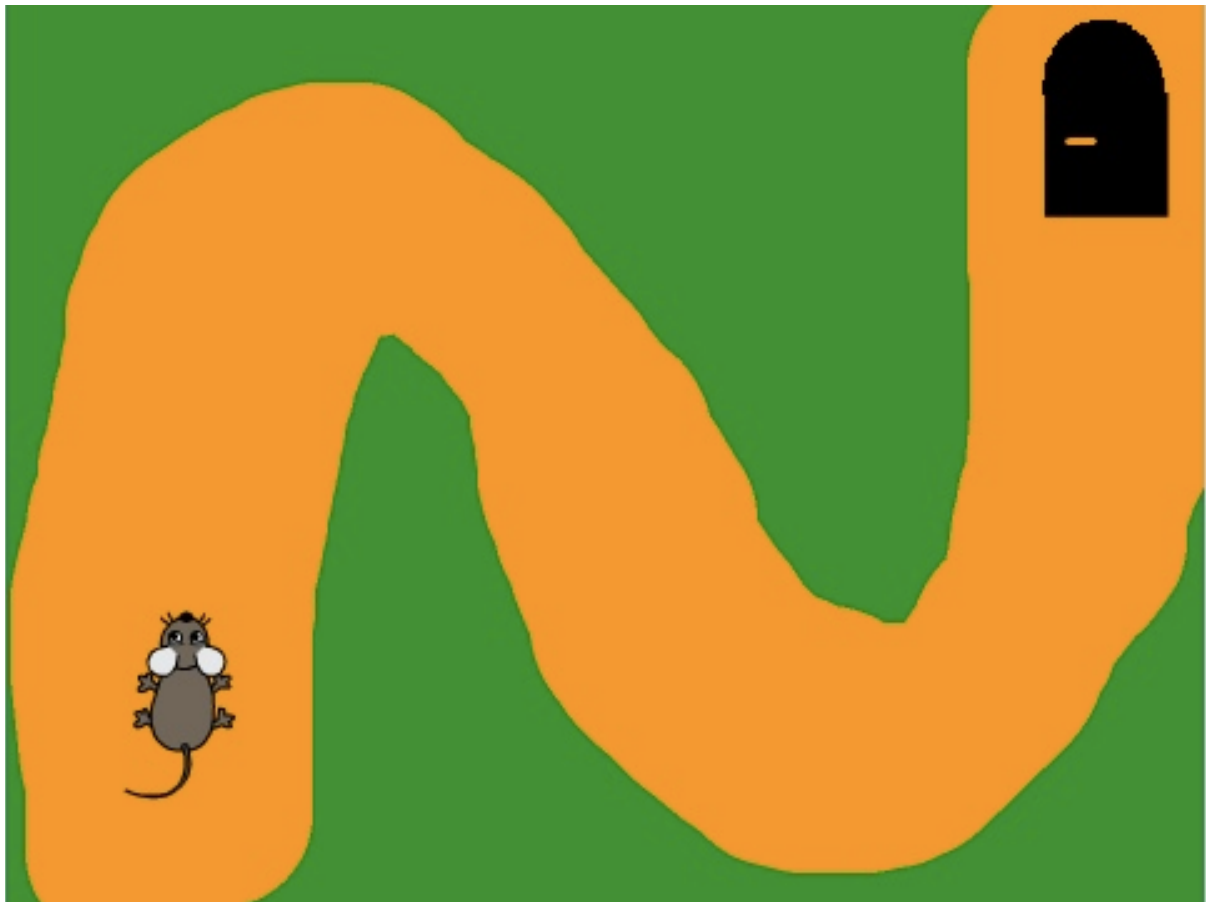
1. Taustade joonistamine ja spraidi liigutamine



Ülesanne:

- ✓ Loo lava, millel on üks sprait (näiteks *Hiir*) ja maastik koos teerajaga.
- ✓ Koosta programm, et *Hiir* saaks arvuti nooleklahvide abil ringi liikuda
- ✓ Joonista uus sprait (näiteks *Uks*)
- ✓ Joonista *Uksele* lisakostüüm (näiteks avatud uks)

Näide:



Näpunäited:

Hiire liigutamiseks tuleb programmeerimisalale lohistada järgmised



plokid.

Kuidas liigutada *Hiirt* tagurpidi?

Mis juhtub kui liikumise/pööramise kastis olevat arvu suurendada-vähendada?

2. Teerajal püsimine



Ülesanne:

- ✓ Koosta programm, kus juhul kui *Hiir* puudutab *Ust*, kuvatakse avatud uks
- ✓ Koosta programm, kus *Hiir*, juhul kui ta väljub teerajalt, liigub tagasi alguspunkti

Näide:



Näpunäited:

Ukse avamiseks tuleb *Ukse* spraidile koostada programm, mis käivitudes (rohelise lipu vajutamise) kuvab ukse (plokk näita) ning alustab lõputut tsüklit (plokk korda lõputult). Lõputu kordus on vajalik, sest tegevus peab toimuma alati kui puudutakse *Ust*. Ukse avamiseks kasutatakse tingimust (kui ... siis) - juhul kui sprait *Hiir* puudutab ust, näidatakse ukse kostüümi 2 (avatud uks), juhul kui ust ei puudutata kuvatakse kostüümi 1 (suletud uks).



Hiire teerajal püsimiseks tuleb *Hiire* spraidile kooostada programm, mis käivitudes kontrollib pidevalt, kas *Hiir* on väljunud teerajalt (puudutab mingit värvi, mida ei ole teerajal). `puudutab värvi ?`

Kui *Hiir* on rajalt väljunud saadetakse see kindlasse alguspunkti (nt x:-165 y:-148). `mine x: -165 y: -148`

NB! kui teerada on kirju tuleks teeraja servadesse teha ääred, kasutades värvi, mida teerajal ei ole.

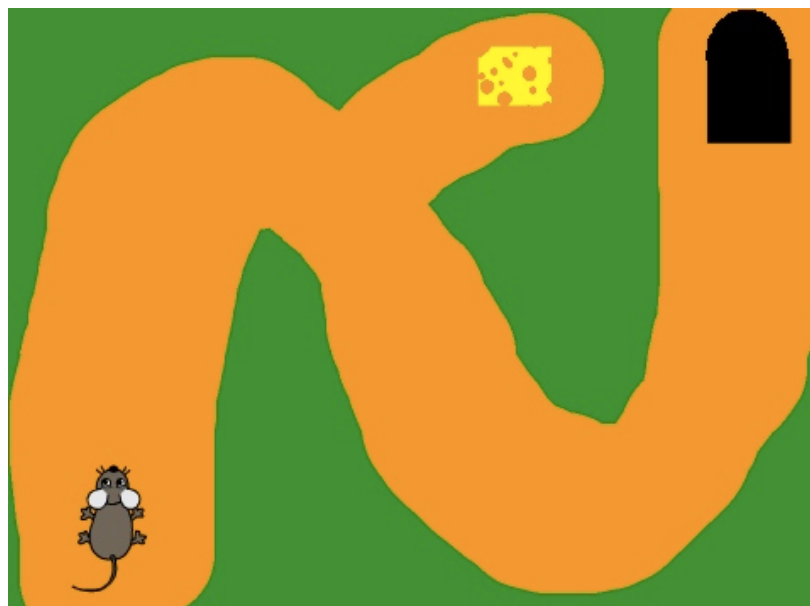
3. Teadete saatmine



Ülesanne:

- ✓ Tee uuest lavast koopia ja lisa sellele üks harutee
- ✓ Joonista uus sprait *Uks2*
- ✓ Joonista uus sprait *Juust*
- ✓ Programmi käivitamisel peidetakse *Juust*
- ✓ Koosta programm, kus *Ukse* puudutamisel kuvatakse ühe sekundi jooksul avatud ust ja väljastatakse teade "tase 2" ning peidetakse *Uks*
- ✓ Juhul kui *Hiir* saab teate "tase 2" liigub ta tagasi algpositsioonile
- ✓ Juhul kui *Lava* saab teate "tase 2" võetakse taustaks teine taust (taust, millel on harutee)
- ✓ Juhul kui *Juust* saab teate "tase 2" muudetakse *Juust* nähtavaks
- ✓ Juhul kui *Uks2* saab teate "tase 2" muudetakse *Uks2* nähtavaks

Näide:



Näpunäited:

Teate saatmiseks tuleb esmalt teha uus teade



Uue teate tegemiseks vajutage teate ploki väiksel mustal kolmnurgal ja avanenud rippmenüüs valige “uus teade ...”

Sisestage avanenud aknasse teate nimi (näiteks “tase 2”)

Kui soovite teadet kasutada valige teamise plokk ja sobiv teade. Teate vastuvõtja juures tuleb kasutada plokki “kui teade ...”

Näiteks *Hiire* ja *Ukse* puudutusel peab kasutama järgmisi plokkide:

Uksel



Hiirel



Erinevate spraitide puhul tuleb tähelepanu pöörata sellele, millal nad peaksid olema nähtavad ja millal mitte. Näiteks *Juust* ei tohiks programmi käivitamisel nähtav olla, seega rohelisest lipukest vajutades tuleb sprait *Juust* peita.



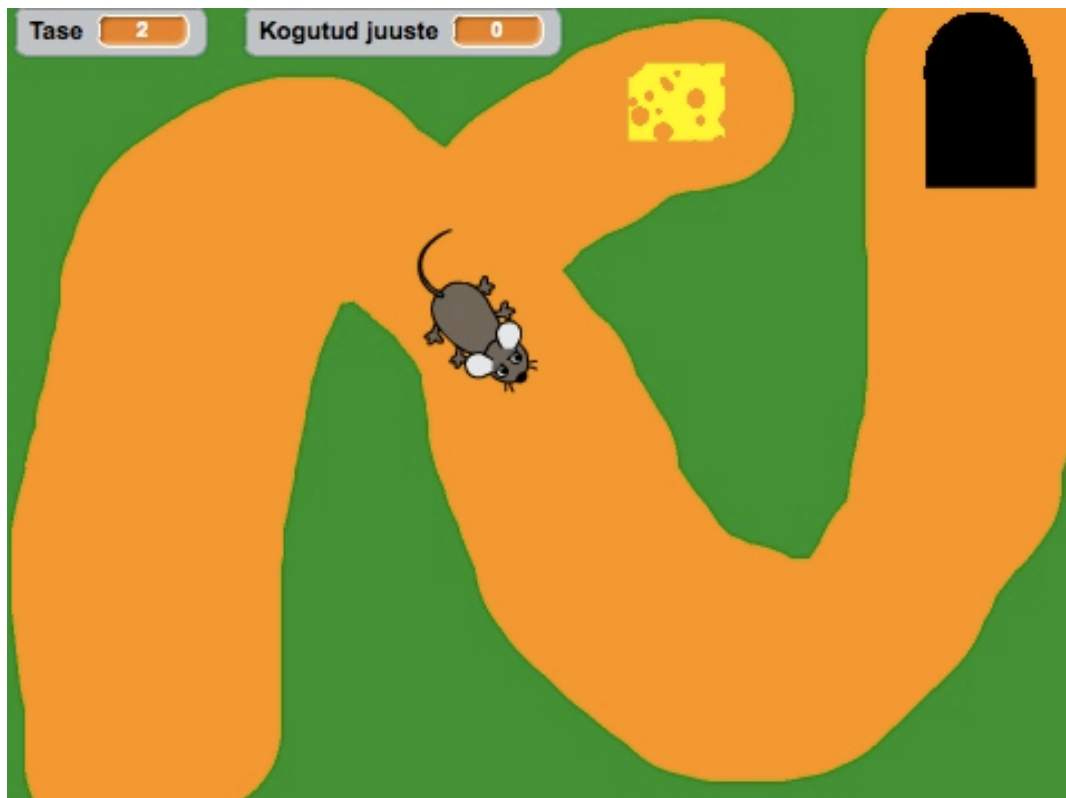
4. Muutujate kasutamine



Ülesanne:

- ✓ Loo uus muutuja "Kogutud juuste"
- ✓ Kui *Hiir* puudutab *Juustu* suurendage muutujat ja peitke *Juust*
- ✓ *Uks2* avaneb ainult siis kui *Hiir* on *Juustu* ära söönud
- ✓ Loo muutuja, mis kuvab mängu algusest, mitmendal tasemel parajasti ollakse
- ✓ Tee olemasolevast lavast koopia ja joonista sinna üle tee jõgi
- ✓ Kui *Juust* on söödud pääseb uksest uude tasemesse (jõega tase)
- ✓ Loo uus sprait *Uks3*, mis kuvatakse kolmandas tasemes

Näide:



Näpunäited:

Muutuja kasutamiseks tuleb esmalt teha uus muutuja (Andmete plokid → Loo muutuja). Loodud muutuja korral tekivad järgmised plokid:

☒ Kogutud juuste

Kui linnuke on muutuja ees, kuvatakse põhiekraanil muutujat. Plokk annab muutuja arv väärtuse.

võta Kogutud juuste = 0

Muutujale mingi kindla väärtuse omistamine

muuda Kogutud juuste 1 võrra

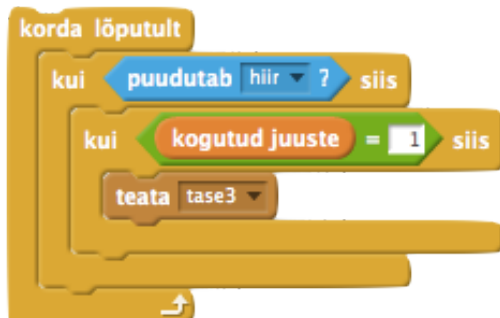
Muutuja väärtuse suurendamine teatud arvu võrra (vähendamiseks kasutada arvu ees - märki)

näita muutujat Kogutud juuste

Muutuja näitamine/peitmine (vajalik näiteks mängu alustamisel, kui alguses ei ole muutuja kuvamine oluline)

peida muutuja Kogutud juuste

Juhul kui soovitakse muutujat kasutada ukse avamise tingimusena, tuleb enne ukse avamist kontrollida, kas kogutud *Juustude* arv on õige:



5. Kiiruse muutmine, automaatne liikumine



Ülesanne:

- ✓ Täienda *Hiire* programmi selliselt, et juhul kui *Hiir* ületab jõge (puudutab sinist värvi), siis *Hiire* liikumiskiirus väheneb
- ✓ Joonista uus sprait *Kala*
- ✓ Pane *Kala* mööda jõge edasi-tagasi liikuma
- ✓ Juhul kui *Hiir* puudutab *Kala* saada *Hiir* tagasi algusesse

Näide:

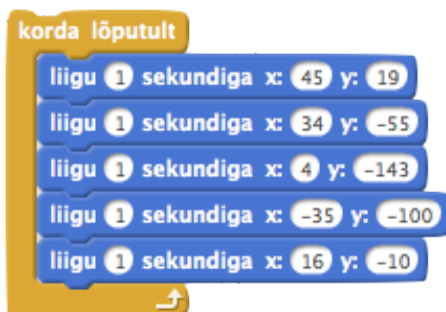


Näpunäited:

Hiire liikumiskiiruse vähendamiseks tuleb *Hiire* liikumise plokkidele lisada tingimus vee puudutamise kohta:



Kala liigutamiseks tuleb määrata punktid, mida *Kala* läbima peab ning seejärel teda nende punktide vahel liigutama:



Hiire ja *Kala* puudutust on võimalik lahendada kahte moodi:

- * esimene võimalus on lisada *Kala* puudutuse kontroll *Hiire* spraidi alla (kontrollitakse, kas puudutab *Kala*, analoogia teerajal püsimisega).
- * teine võimalus on lisada *Kala* spraidile kontroll, kas puudutatakse *Hiirt* ja puudutuse korral saata *Hiire* spraidile teade (näiteks mine algusesse)

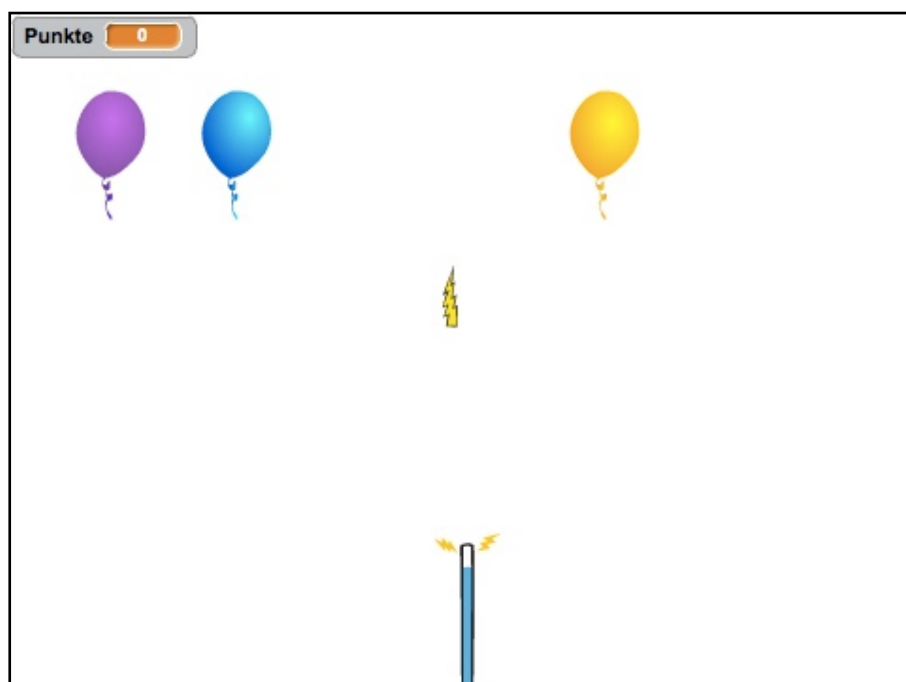
6. Õhupallimäng



Ülesanne:

- ✓ Tee kolm uut erinevat värvi õhupalli spraiti, sprait *Võlukepike* ja *Välgunool*
- ✓ Pane õhupallid ekraani ülaservas erineva kiirusega liikuma
- ✓ Koosta programm, et vasakule-paremale noole abil *Võlukepike* pööraks
- ✓ Tühiku klahvile vajutades saadab *Võlukepp* välja *Välgunool*. Kui *Välgunool* puudutab serva siis see kaob. Kui *Välgunool* tabab mõnda õhupalli, siis kaovad mõlemad, õhupall ilmub uuesti nähtavale 3 sekundi pärast
- ✓ Lisa mängule punktide arvestus

Näide:



Näpunäited:

Selleks, et õhupallid erineva kiirusega liikuma panna on kaks varianti:

* kasutades plokki **liigu 1 sekundiga x: -79 y: 87**, määrake kindla aja asemel juhuslikku aega (plokk **juhuarv 1 kuni 3**)



* kasutades samasid plokkide, mis eelmisel juhul, jättes aja muutmata ja kasutades juhuarvu hoopis x-koordinaadil (horisontaalne liikumine)



Enne *Võlukepikese* liikuma panekut tuleb kostüümi kese määrata *Võlukepi* alumisse serva.



Nool tuleks välja saata mingit kindlat klahvi (näiteks tühik) vajutades. Selleks, et *Välgunool* liiguks samas suunas, kuhu on osutatud *Võlukepike* tuleb esmalt luua muutuja noolesuund ning sellele anda *Võlukepi* spraidi suund kasutades Liikumise plokki suund **suund**. Peale muutujale suuna omistamist tuleks *Välgunoolele* saata teade (nt saadanool).

Mängu käivitamisel tuleb *Välgunoole* spraidil käivitada programm, mis kontrolliks pidevalt, kas nool puudutab mõnda õhupalli. Juhul kui on toimund õhupalli puudutus saadetakse vastavale õhupallile teade (nt sinine), ning peidetakse *Välgunool*.



Välgunoole spraidile tuleb lisada ka tegevus noole väljasaatmiseks. Enne noole väljastamist tuleb *Vägunool* saata *Võlukepikese* juurde ning määrata liikumissuund **osuta suunas** **noolesuund**. Seejärel peab *Vägunool* liikuma kuni servani ja end peitma.

Iga õhupalli juurde tuleb lisada ka programm, mis peidab vastava teate (nt sinine) saamisel õhupalli kolmeks sekundiks. Sama programmi alla tuleb lisada ka punktide lisamine.

7. Looduses jalutamine



Ülesanne:

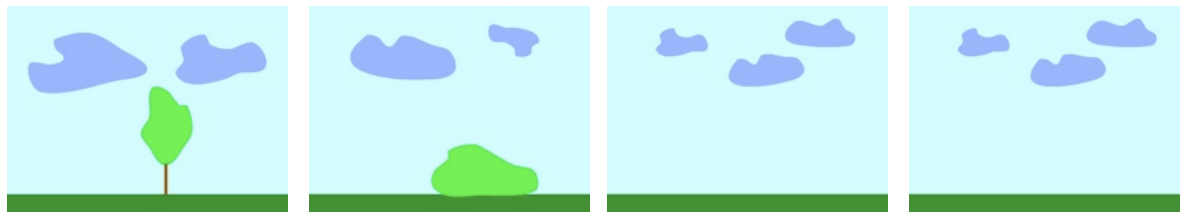
- ✓ Joonista neli ekraani suurust spraiti, kus on teerada, puud-põõsad ja pilved (midagi sellist, mis aitaks näidata liikumist). Alternatiivne variant on joonistada üks taust ja neli spraiti pilvedega. Võite joonistada ka pealtvaates maantee, mille ääres on põõsad või mille keskel on telgjoon.
- ✓ Paneme tausta liikuma. St tausta spraite liigutatakse ühest servast teise. Juhul kui esimese spraidi serv on jõudnud äärele, hakkame liigutama teist, teise spraidi serva äärele jõudmisel kolmandat jne.
- ✓ Muuda tausta liikumine aja jooksul kiiremaks

Näide:



Näpunäited:

Taustaspraitide joonistamisel tuleb pöörata tähelepanu sellele, et omavahel kokku puutuvate spraitide äärmised servad ühtiksid. Kõige lihtsam on servadesse mitte keerulisi objekte jätta.



Enne taustaspraitide liigutamist tuleb teha ka neli muutujat, näitamaks, milline sprait parajasti nähtav on (muutuju väärtused vastavalt 0 ja 1).

Mägu käivitamisel tuleb näidata esimest taustaspraiti (*Taust1*) ja ülejäänud peita. Lisaks tuleb sprait asetada ekraani keskele (juhul kui on täisekraan sprait, kui on nt ainult pilved vms, siis vastavale asukohale) ning määrata muutujas, vastava tausta nähtavus õiges (väärtus 1).



Ülejäänud taustaspraidid tuleb mängu alguses peita ning nende asukohaks määrata parem serv. Samuti tuleb muutuja määrata mittenähtavaks (väärtus 0).

Edasises programmis tuleb liigutada neid sprait, mille muutuja väärtus on 1.



Juhul kui taustaspraidi keskkoh on ekraani keskkohast vasakul ja järgmine taust ei ole nähtav (muutuja väärtus 0), tuleb järgmine taust

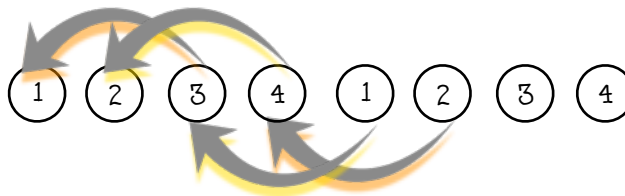
teha nähtavaks ning väljastada järmise spraidile teade, et ta liiguks alguspositsiooni ja ilmuks nähtavale.



NB! Järneva spraidi alguspositsiooni puhul arvestage, kuskohas paikneb spraidi keskkoh (meie näites ekraani keskel, seega tuleb algusele liita veel pool ekraani asukohta 240px).



Tee vastavad programmid kõigi taustaspraitide jaoks. Ära unusta, et viimane sprait peaks saatma teate esimesele. Lisaks tuleb iga tausta juures teate saatmisel muuta üleeelmise (eelmine võib veel nähtav olla) tausta nähtavus vääraks (muutuja 0). Arvesta, et esimese tausta jaoks on üleeelmiseks eelviimane ja teise tausta jaoks viimane taust.



Selleks, et saaksime tausta liikumise kiirust muuta tuleb luua uus muutuja (nt kiirus) ja kõik senised kindlaksmääratud kiirused sellega asendada. Arvestada tuleb ka sellega, et juhul kui liikumine toimub paremalt vasakule peab kiirus olema negatiivne.



PS! Ära unusta, kiirusele mängu alguses väärtust andmast.

Liikumiskiiruse sujuvaks muutmiseks tuleb seda iga teatud aja tagant suurendada. Aja määramiseks tuleb teha muutuja aeg ja seda iga sekundi tagant suurendada.



Kiiruse suurendamiseks tuleb senise lõputu korduse sisse lisada uus kordus, mis kehtaks kuni etteantud ajani (aeg, mille pärast kiirust suurendatakse). Kui etteantud aeg on täitunud, tuleb aeg ära nullida ning kiirust suurendada.



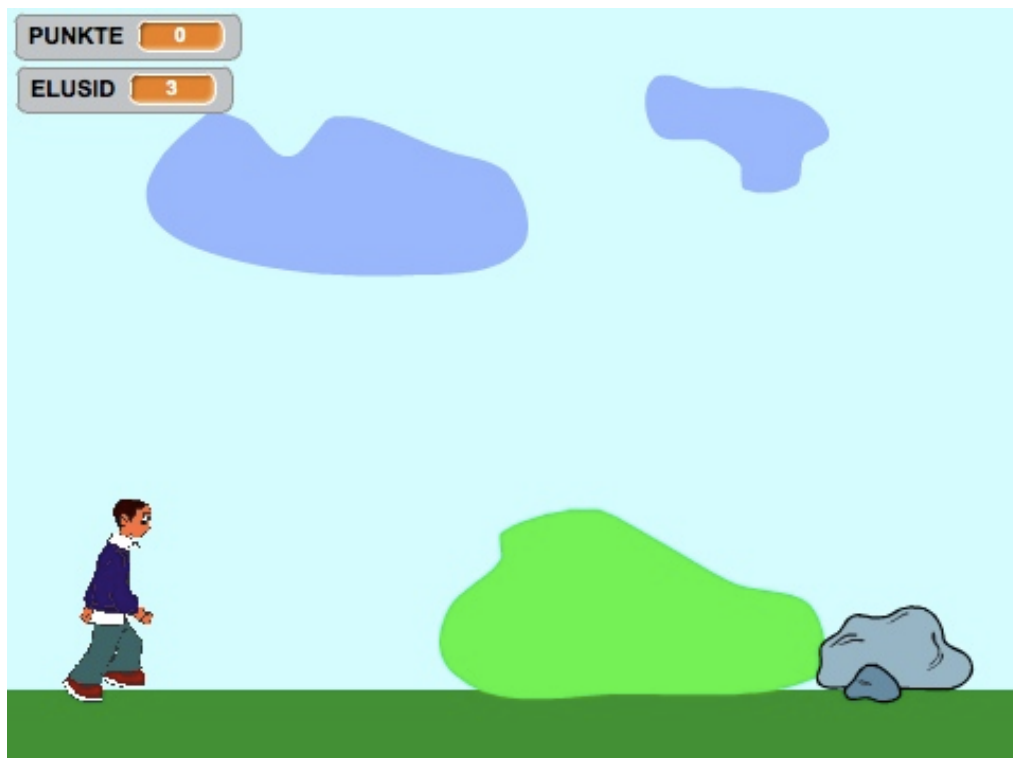
8. Takistuste vältimine



Ülesanne:

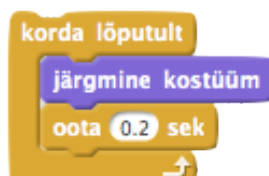
- ✓ Loo uus sprait *Poiss* ja *Kivi*
- ✓ Paneme poiss liikuma (kõndimise animatsioon)
- ✓ Paneme poiss klahvi ülesnool vajutusel hüppama
- ✓ Paneme *Kivid* liikuma ja väldime nende puudutust. Iga kivi otsa komistamine võtab elusid vähemaks, iga üle hüpatud kivi annab punkte juurde.

Näide:



Näpunäited:

Kõndimise animatsiooni lisamiseks tuleb mängu alustades iga teatud aja tagant hakata vahetama *Poisi* kostüüme.



Hüppamise tegemisel tuleb esmalt *Poiss* liigutada maapinnast ülespoole. Sujuvama liikumise saamiseks kasutame plokki **liigu 1 sekundiga x: -140 y: -100**, kus ajaks on mõistlik määrata u 0,5sekundit ning y-koordinaadiks umbes 130-140px praegusest asukohast.

Selleks, et *Poiss* maa peale tuleks, tuleb teda seni allapoole liigutada, kuni poiss maad (kindel värv) puudutab. Maa poole liikumise kiirus võiks arvestada ka tausta liikumise kiirust.



Kivi liikuma panemiseks tuleb kivi paremast servast vasakusse liigutada sama kiirusega, millega liigub taust. Tähele tuleb panna, et kivi peab liikuma seni, kuni puudutab paremat serva (st x-koordinaat on <0).



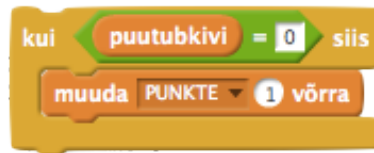
Kivi otsa komistamisel elude vähemaksvõtmisel tuleb esmalt luua muutuja elud ning seejärel iga kord, kui *Poiss* puudutab *Kivi* vähendada muutuja väärtus. Kui muutuja väärtuseks saab null, tuleks mäng peatada ja kuvada näiteks tekst "Mäng läbi".

Selleks, et *Kivi* puutumisel ei võetaks punkte seni maha, kuni kokkupuude kestab või punktid on nullis, tuleb luua uus muutuja (näiteks

puutubkivi) ning kiviga kokkupuutel lisaks elude vähendamisele muuta ka muutuja väärtust.



Punktide arvestus käib samuti läbi loodud muutuja - juhul kui kivi serva jõudes on muutuja väärtus 0 saame punkti lisada, vastasel juhul mitte.



PS! Ära unusta, muutujat puutubkivi iga uue kivi tulekuga uuesti ära nullida.

Selle materjali valmimist on toetanud:



HASARTMÄNGUMAKSU
NÕUKOGU

Swedbank



Materjali arendas MTÜ Robootika.



See teos on litsentseeritud Creative Commonsi Autorile viitamine + Mitteäriline eesmärk + Jagamine samadel tingimustel 4.0 Rahvusvaheline litsentsiga.