

Kraps, pall ja papagoi

Koostada rakendus (mäng) järgmise stsenaariumi alusel:

Kraps jalutab rahulikult palliga rannas edasi-tagasi. Järsku ilmub taevasse papagoi, kes lendab ringi ja narrib Krapsu. Kraps üritab mängija abiga tabada palliga papagoid: palli lennutamiseks peab kasutaja vajutama tühiku klahvi. Mängu eesmärgiks on tabada papagoid võimalikult rohkem kordi etteantud aja jooksul (max_aeg).

Rakendust arendada järk-järgult.

1. Krapsu liikumine.

Liikumise ja välimuseplokid; skriptide käivitamine.

Esiialgu võib Kraps liikuda ilma pallita.

Krapsul on kaks kostüümi. Liikumiseks koostada skript:

Määrata pööramise stiil 'ainult nägu vasakule/paremale'.

Kasutajal võiks olla võimalik juhtida Krapsu nooleklahvide abil.

Koostada skriptid, mis käivituvad vastavalt klahvidega 'vasak nool' ja 'parem nool'.



2. Papagoi tegevused

Spraitide ja kostüümide lisamine, skriptide kopeerimine ja redigeerimine, tingimused ja valikud, spraitide asukoha määramine, spraitide peitmine ja kuvamine, spraitide „ringliiklus“, juhuslike arvude kasutamine.

Lisada sprait **papagoi** ja selle teine kostüüm. Kopeerida **Krapsu** skript **papagoile** ja proovida.

Ringliikluse korraldamiseks muuta: kui papagoi jõuab lava parema servani, peita sprait (käsk **[peida]**) korraks (1-3 sek), viia lava vasakusse serva ja tuua nähtavale (**[näita]**).

Pausi tekitamiseks kasutada **juhuarve**.

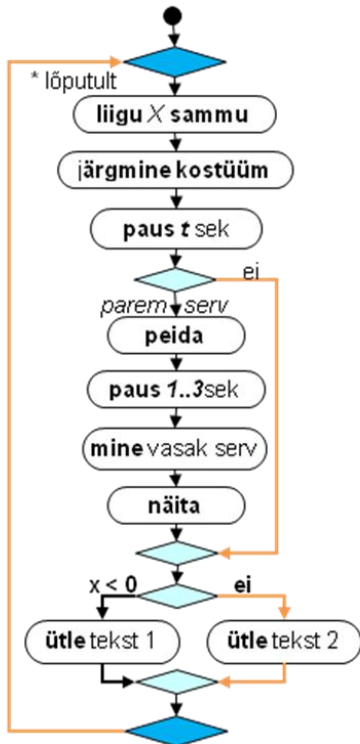
Et teha tabamine raskemas, võiks käsu **[pane x -240]** asemel võtta näiteks käsu **[pane x juhuarv -240 kuni 120]**. Papagoi ilmub nüüd nähtavale juhuslikus kohas antud vahemikus.

Lisada **papagoi** "jutt" (narrimine): **kui** asukoht < 0-st **jutt1** **muidu** **jutt2**



Tegevuste kirjeldamisel on juhendmaterjalil edaspidi kasutusel ka algoritmikeel ja UML tegevusdiagrammid. Võrrelda alltoodud algoritme papagoi skriptiga. Kas on sarnane?

Algoritm - UML' tegevusskeem



Algoritm- algoritmikeel

```

korda lõputult
liigu X sammu
järgmine kostüüm
paus t sek
kui parem serv siis
peida
paus 1..3 sek
mine vasak serv
näita
lõpp kui
kui x < 0 siis
  ütle tekst1
muidu
  ütle tekst2
lõpp kordus
    
```



3. Palli lisamine ja selle tegevused.

Joonistamisredaktor, sündmus 'puudutab', objekti viimine teise spraidi asukohta, spraidi kese, tingimuslik kordus, skripti peatamine, teadete saatmine ja vastuvõtmine.

Palli jaoks joonistada kujutis joonistamisredaktori abil: ring (shift-klahvi abil ellipsist) ja sellele mõned värvilised laigud. Spraidi (kostüümi) kese on tavaliselt keskel, võimalik määrata ka mujale.

Koostada tühikuga käivitav skript palli lendamiseks Krapsu juurest üles lava servani.

```

mine Kraps
näita
kordus kuni puudutab serv
  muudaY h võrra
peida
    
```



Nüüd tuleb kontrollida, kas pall tabab papagoi. Seda on võimalik teha nii palli kui papagoi skriptis. Otstarbekam lisada palli skripti, sest sel pole mõtet, kui pall ei lenda.

Tabamusest tuleks kindlasti teavitada (teade ...), et ka teised spraidid (Kraps ja papagoi) saaks reageerida (ühe spraidi skript ei saa määrata tegevusi teiste spraitidega). Teavitama peaks ka mitte-tabamisest.

```

mine Kraps
näita
kordus kuni serv
  muudaY h võrra
  kui puudutab papagoi
    peida
    teavita Pihtas
  peata skript
lõpp kui
lõpp kordus
peida
teavita Mõõdas
    
```

Kassi ja papagoi reaktsioonid:

Pihtas: papagoi peidetakse korraks, Kraps ütleb midagi

Möödas: papagoi naerab, laskub hetkeks allapoole, Kraps - mingid graafikaefektid (keere jm).

```

Kraps
kui saabub teade Pihtas
  kustuta graafikaefektid
  ütle Said?! 1 sekundit

kui saabub teade Möödas
  muuda keere efekti 50 võrra
  ütle Kohe saad! 1 sekundit
    
```

```

Papagoi
kui saabub teade Pihtas
  peida
  oota juhuarv 1 kuni 3 sek
  pane värv efekt juhuarv 0 kuni 200 -le
  pane x juhuarv -240 kuni 120 -ks
  näita

kui saabub teade Möödas
  mängi heli naer
  muuda y -100 võrra
  oota 0.3 sek
  muuda y 100 võrra
    
```

4. Visete ja tabamuste loendamine

Muutuja, omistamine

Lua kaks globaalset (nende väärtusi saavad kasutada ja muuta kõigi spraitide skriptid) muutujat loendamiseks: **viskeid** ja **tabas**. Praegu on mõistlik teha nende muutujate väärtuste muutmiseks kas eraldi skriptid või lisada vastavad käsud palli skripti. Visete arv **viskeid** peaks muutuma palli lenduminekul ja tabamuste arv **tabas** pärast papagoi puudutamist.

NB! Muutujate algväärtustamine!

Korduval käivitamisel jätkatakse eelmisest väärtusest. Uue mängu algul tuleks loendurid nullida. Kui neid muutujaid palli skriptide juures juba kasutusel on, võiks ka algväärtustamised olla palli teises skriptis, mis käivitub mängu alustamisel (roheline lipukesega).

```

kui klõpsatakse
  võta viskeid = 0
  võta tabas = 0
    
```

```

kui vajutatakse klahvi tühik
  muuda viskeid 1 võrra
  mine Kraps
  näita

korda kuni puudutab serv ?
  muuda y 20 võrra
  kui puudutab papagoi ?
    muuda tabas 1 võrra
    peida
    teavita Pihtas
    peata skript
  peida
  teavita Möödas
    
```

5. Aja kuvamine ja mängu lõpetamine.

Taimer, Käsk [peata kõik], liugur (kerimisriba) muutujate väärtuste määramiseks, "teatespraidid"

Lua kaks muutujat: **aeg** (jooksev aeg) ja **max_aeg** (mängu aeg). max_aeg monitorile laval lisada liugur (kerimisriba), mille abil saab "käsitsi" muuta muutuja väärtust, ja määrata min ja max väärtused. Valida max_aeg väärtuseks esialgu 10 (sek).

Skript taimeri nullimiseks, aja näitamiseks ja mängu lõpetamiseks võib olla suvalise spraidi või ka lava oma.

Taimeri plokid on grupis **Andurid**.

Kui klõpsatakse rohelist lippu
 taimer algseisu
korda kuni aeg > max_aeg
 aeg = taimer
 peata kõik

```

kui klõpsatakse
  taimer algseisu
  korda kuni taimer > max_aeg
    võta aeg = taimer
  ütle Mäng on läbi! 2 sekundit
  teavita Tule ja oota
  peata kõik
    
```



Teadete, selgituste jmt kuvamiseks võib ploki **ütle** asemel kasutada nõ "tekstisprait". Spraidil võib olla mitu kostüümi: igaüks neist kujutab omaette teadet. Vastavalt olukorrale võetakse sobiv kostüüm ja tehakse sprait nähtavaks.

Siin on kasutusel kaks teadet:

Aeg on läbi! (kostüüm Aeg)

Visete arv on täis! (kostüüm Visked)

Sõltuvalt lõpetamise põhjusest valitakse ja kuvatakse vastav teade.

Mängu võib lõpetada ka lubatud visete arvu möödumisel. Loo muutuja **m_viskeid**, lisada liugur (ka min ja max), valida sobiv väärtus.

Käsk 'oota kuni ...' seab skripti ooteseisu (kaudselt kordus).

Skripti võib panna suvalise spraidi moodulisse: näiteks lava või Kraps



6. Palli liikumine koos Krapsuga. Olekumuutuja.

Pall peaks liikuma koos Krapsuga ja võiks ka pöörelda (NB! **Lisaülesanne** kiirematele). Sellise liikumiste sünkroniseerimise saab teha erinevalt. Näiteks võib Kraps iga kord pärast sammu astumist „kutsuda“ palli enda juurde.

Lisada teavitamise käsk [teavita *tule* ja oota] Krapsu põhiskripti.

Koostada palli jaoks skript, mis käivitub teatele **Tule**: viia pall **Krapsu** juurde ja alla (valida sobiv nihe) ning pöörata palli, arvestades Krapsu liikumise suunda.

Mis juhtub, kui nüüd pall pannakse lendama? Tuleb tagasi!

Võtta kasutusele nõ olekumuutuja (tunnus) **lendab**:

- kui **pall** lendab, on muutuja väärtus **1** muidu **0** (kokkulepe)
- muutuja väärtust muudab **palli** põhiskript
- algväärtus 0 (koos teiste muutujate algväärtustamisega).
- lisada Krapsu skripti valikuplokk kui lendab=0 teatega 'tule'.



7. Rakenduse kirjeldus (dokumentatsioon)

Rakendusest arusaamiseks, vajadusel muutmiseks ja otstarbekaks kasutamiseks koostatakse alati ka projekti dokumentatsioon:

- kasutusjuhend – eesmärgid, tegevuste käivitamise võimalused
- spetsifikatsioon – selgitused spraitide ja nende skriptide (tegevuste) ning muutujate kohta.

Scratchis võib sellised kirjeldused kirjutada projekti märkustesse (vt menüüs Fail – projekti märkused).

Skriptide juures võivad olla kommentaarid - lühikesed selgitavad tekstid. (Loomiseks paremklõps skriptide alas, ilmuvast menüüst valida "**lisa kommentaar**"; saab muuta asukohta, siduda konkreetse käsuga jm.)

Kasutusjuhend (näiteks eraldi spraidina) võiks olla kättesaadav ka projekti kasutajaliideses.

Suuremate projektide korral võivad taolised märkused olla vormistatud eraldi dokumentidena.

Allpool on näitena toodud vaadeldava projekti (rakenduse) kirjeldus.

Kasutusjuhend

Kraps jalutab palliga edasi-tagasi. Papagoi lendab ringiratast ülalpool.

Eesmärk on tabada palliga papagoi võimalikult palju kordi valitud aja jooksul: `max_aeg` (saab muuta).

Programm näitab aega mängu algusest, tehtud visete ja tabamuste arvu. Visete arvu saab piirata muutuja `m_viskeid` väärtusega.

Uue mängu algus: **Roheline Lipp**. Pall läheb lendu Krapsu kohalt tühikuklahvile vajutamisel.

Selgitused

Põhispraidid: Kraps, papagoi ja pall. Lisaks sprait **Teated**.

Kraps (2 kostüümi, 5 skripti). Jalutab palliga edasi-tagasi.

Põhiskript (käivitub rohelise lipuga) juhib Krapsu (ja ka palli) liikumist edasi-tagasi. Kui pall ei lenda, tuuakse see igal sammul Krapsu juurde (palli vastava skripti reaktsioon teatele **Tule** Krapsu skriptis).

Nooleklahvide (parem ja vasak) skriptid võimaldavad Krapsu pöörata ja liigutada vastavas suunas.

Palli teadete Pihtas ja Möödas toimel käivitavad skriptid määravad reaktsioonid nendele sündmustele.

Papagoi (2 kostüümi, 3 skripti).

Põhiskript: juhib papagoi "ringliiklust" ja kuvab tekstid Krapsu narrimiseks.

Palli teadete Pihtas ja Möödas toimel käivitavad skriptid määravad papagoi reaktsioonid nendele sündmustele.

Pall (1 kostüüm, 3 skripti).

Põhiskript: juhib palli lendu, kontrollib puudet papagoi või servaga. Kummagi sündmuse korral lend katkestatakse. Skript muudab ka muutujate viskeid, tabas ja lendab väärtusi. Muutujate algväärtustamine toimub eraldi skriptiga (käivitub rohelise lipuga). Krapsu põhiskriptis oleva teate Tule toimel käivituv skript viib palli Krapsu juurde ja pöörab veidi Krapsu loomise suunas.

Lava (1 taust, 2 skripti). Taustaks ranna pilt või ka mingi muu.

Üks skript muudab ja kontrollib taimeri abil aega. Kui aeg ületab muutuja `max_aeg` väärtuse, initialsiseerib vastava teate kuvamise ja lõpetab kogu projekti töö. Teine skript kontrollib lõpetamist visete arvu järgi.

Muutujad:

viskeid – tehtud visete arv; muudab palli põhiskript

tabas – tabamuste arv; muudab palli põhiskript

aeg – aja jooksev väärtus; muudab lava skript

max_aeg – lubatud mängu aeg, kasutaja saab valida (muuta) liuguri abil

m_viskeid – visete maksimaalne arv, valib kasutaja

lendab – palli olekumuutuja: 1 - lendab, 0 - ei lenda; muudab palli skript