



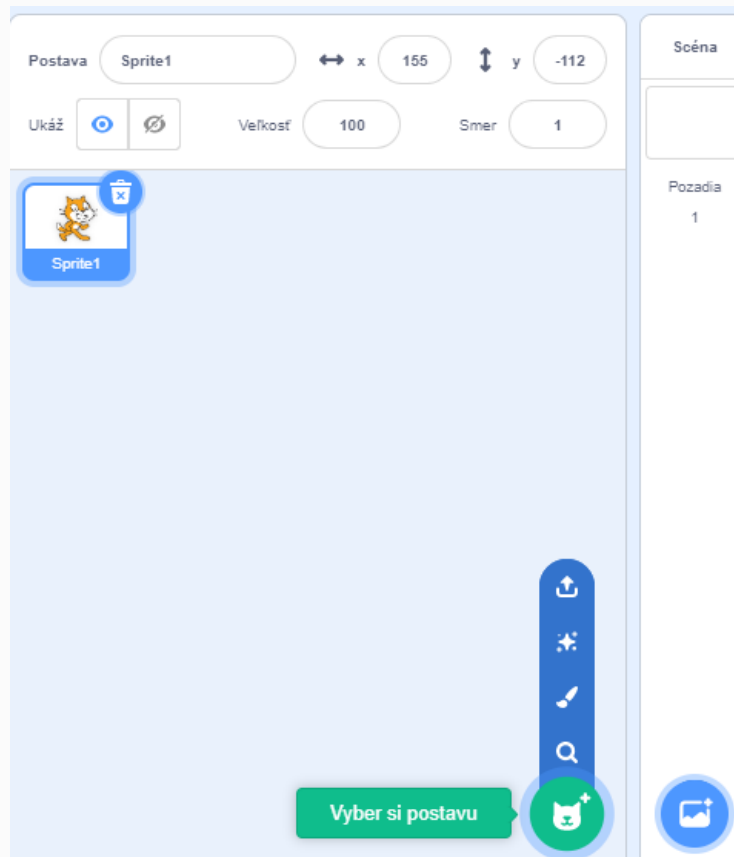
Programko.sk

Scratch

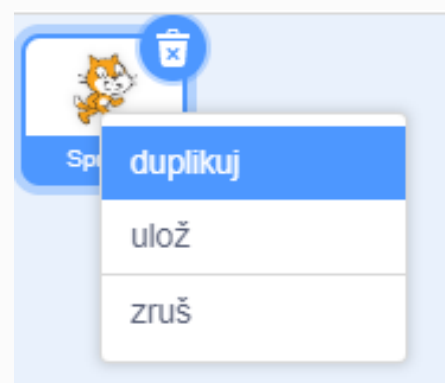
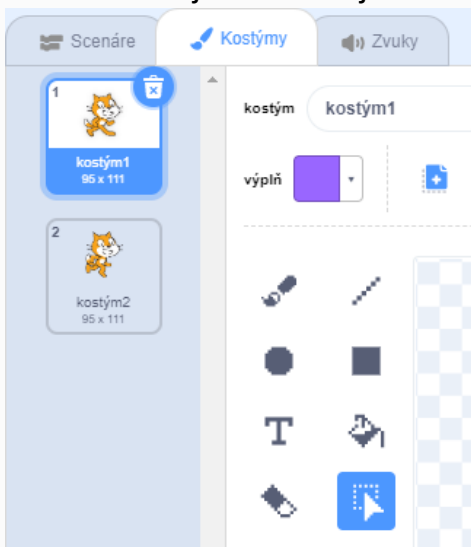
Spoznanie s prostredím

- vytváranie nových postáv (objekty)
- spustenie, zastavenie programu

Možnosť vytvárania vlastných kostýmov pre postavy (objekty), vkladanie vlastných obrázkov



Možnosť dať jednému objektu niekoľko kostýmov a duplikovanie objektov spolu s príkazmi



Možnosť nastaviť pozadie (scénu), scéna funguje rovnako ako postava (objekt), môže mať príkazy a rôzne kostýmy



Správy (komunikácie medzi objektami), podmienky

Objekty môžu medzi sebou komunikovať pomocou správ. Nájdeme ich v kategórii **Udalosti**.



Úloha – ovládanie mačky tlačidlami

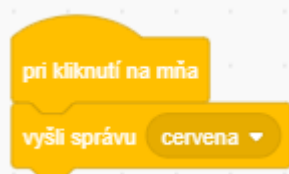
Spravíme jednoduchý editor na kreslenie. Na vrchu sú 4 farebné kruhy, 3 tlačidlá, 2 obrázky kostýmov. Kliknutím na kruhy sa mení farba pera. Tlačidlá nastavujú pero a mažú plochu. Obrázky psa a mačky menia kostým.



Chceme nechať žiakov nakresliť kostýmy pre zmeny farby (farebné kruhy), tlačidlá, ktoré zapínajú/vypínajú písanie pera a mažú doteraz nakreslené čiary, prípadne kludne aj tlačidlá meniace kostým. Pre hlavnú postavu môžeme nechať voľnosť výberu zo širokej ponuky kostýmov v Scratchi.

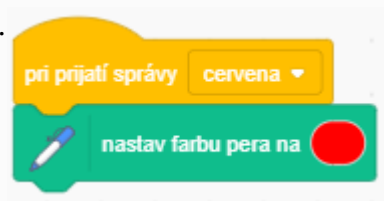
Napríklad keď klikneme na červený kruh, odošleme správu „cervena“, túto správu prijme naša mačka a zmeníme farbu pera na červenú. Tento princíp bude pri všetkých tlačidlách.

Červený kruh bude mať takéto príkazy



a hlavná postava bude mať príkaz,

aby mohla túto správu prijať takýto .



Pre každú správu musíme spraviť takýto samostatný príkaz.

Podmienky

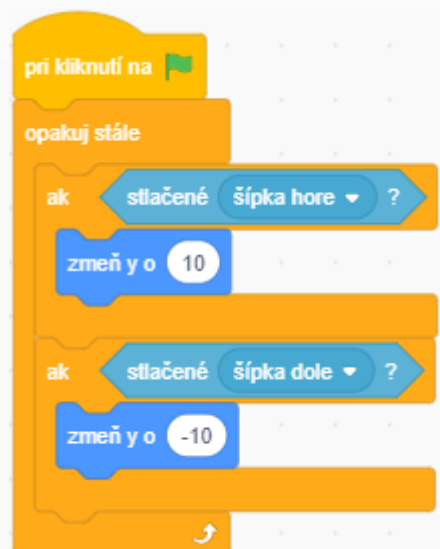
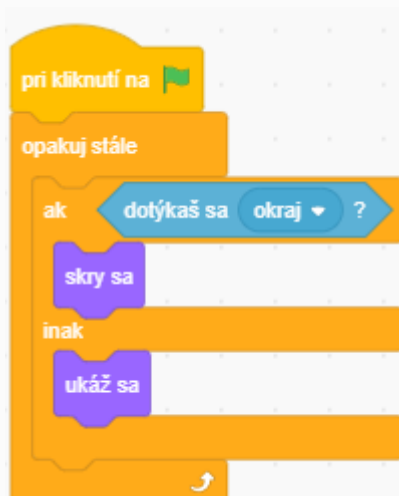
Ak sa niečo stane, potom urobím. Ak dostanem jednotku, pôjdeme na zmrzlinu. Ak budem dobrý, dostanem na Vianoce autíčko. Ak som chorý, nepôjdem do školy. Ak držím zelenú farbičku, kreslím na zeleno. Toto sú podmienky ako ich poznáme my v reči, ale v programovaní sú veľmi silným nástrojom na špecifické odchytenie konkrétnych prípadov.

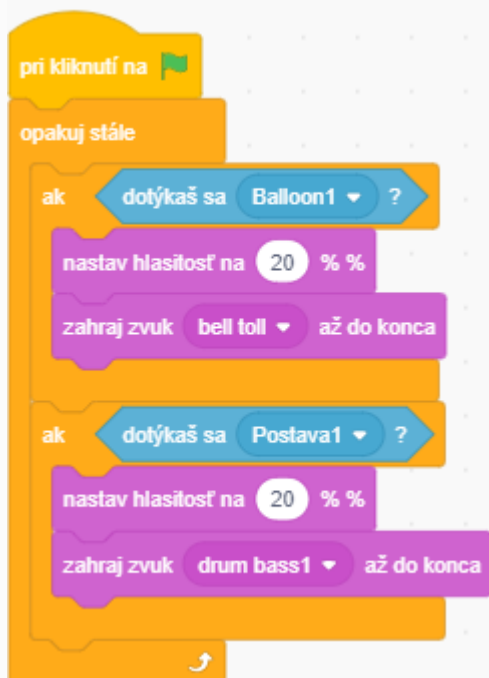
Úloha – interakcia medzi objektami

Máme objekty mačky, balónu a zvona. Podmienky budeme využívať v tejto úlohe nasledovne. Pohyb mačky doprava a doľava vytvoríme pomocou podmienok a to neustálou kontrolou v nekonečnom cykle. Pohyb balóna hore a dole vytvoríme rovnakým spôsobom a budeme kontrolovať či sa dotýka hrany, pokiaľ áno – balón sa skryje. Zvon sa nebude pohybovať ale budeme podmienkami kontrolovať či sa nedotýka jedného z týchto 2 objektov. Podľa toho, ktorého sa dotýka, taký zvuk zvon zahrá. (Podmienky, kto kedy robí zvuk alebo skrýva si môžete aj sami pozmeniť)



Mačke pri spustení programu nastavíme vždy začiatočnú polohu a orientáciu a vytvoríme jej ovládanie.



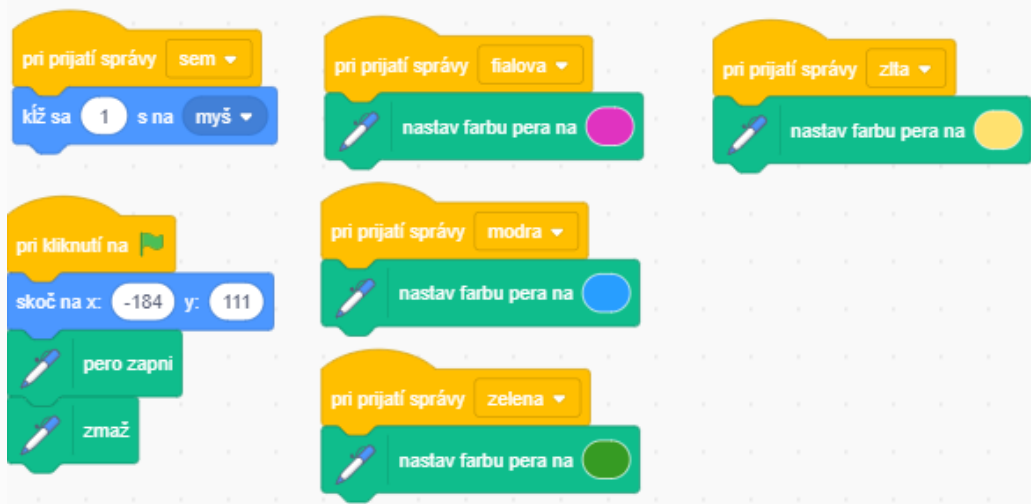
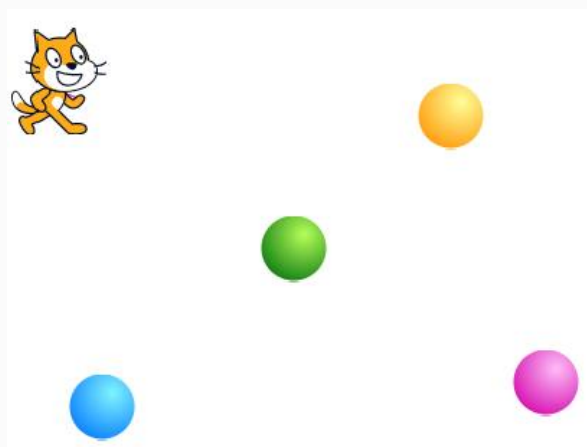


Pre balón takto vytvoríme detekciu či sa dotýka hrany a jeho ovládanie. Tu máme pre tieto 2 veci separátne cykly.

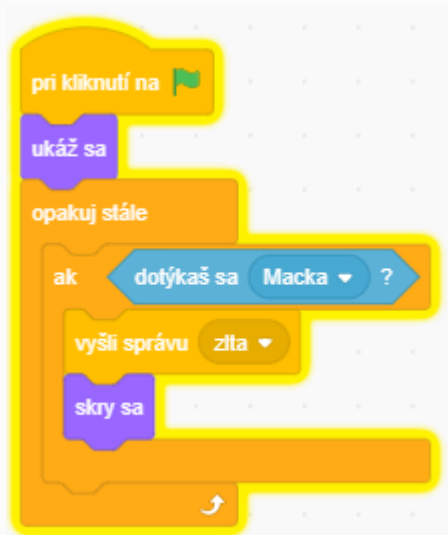
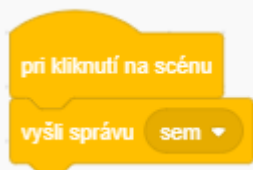
Toto sú podmienky, ktoré kontrolujeme v zvone.

Úloha – zbieranie farieb (BONUS)

Kliknutím na scénu sa nám mačka na dané miesto bude kízať a píše perom. Pri dotyku s niektorou z farieb, sa daná farba skryje a zmení farbu pera.



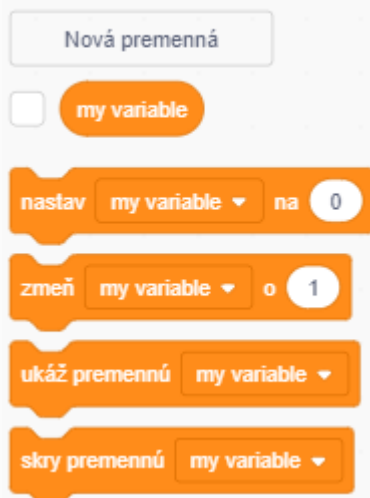
Mačka prijíma všetky správy a podľa nich dávame príkazy. Správu „sem“ vysiela scéna a jednotlivé správy o farbe vysielaajú farebné guľičky.



Detekciu v guľkách farieb vykonávame rovnakým spôsobom ako v minulej úlohe. V nekonečnom cykle.

Premenné

Premenné



Premenná je nejaká hodnota (v Scratchi sú to čísla). Tie môžeme počas behu programu meniť a riadiť nimi program.

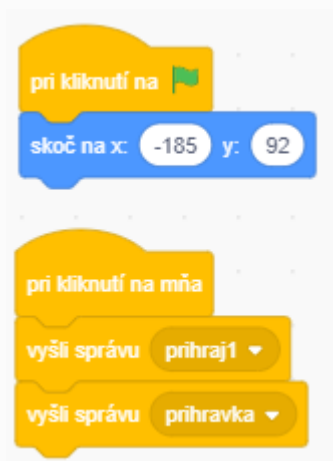
V kategórii Dáta si vieme vytvoriť novú premennú a dať jej ľubovoľný názov, naša sa volá my variable. Vidíme funkcie, ktorými vieme manipulovať s danou premennou. Aktuálny stav premennej si vieme zobraziť aj rovno na obrazovke pomocou **ukáž premennú**. Vieme ju aj vkladať do funkcií namiesto čísel.

Úloha – Baseball

V tejto úlohe si budú mačky medzi sebou prihrávať baseballovú loptičku. Počet prihrávok budeme počítat pomocou premennej Prihravky. Každá mačka musí vysielat špecifickú správu, aby loptička vedela, ku ktorej mačke sa nahráva. Pri každej nahrávke máme ešte správu, ktorá nám zabezpečí, aby sme premennú Prihravky zväčšovali.



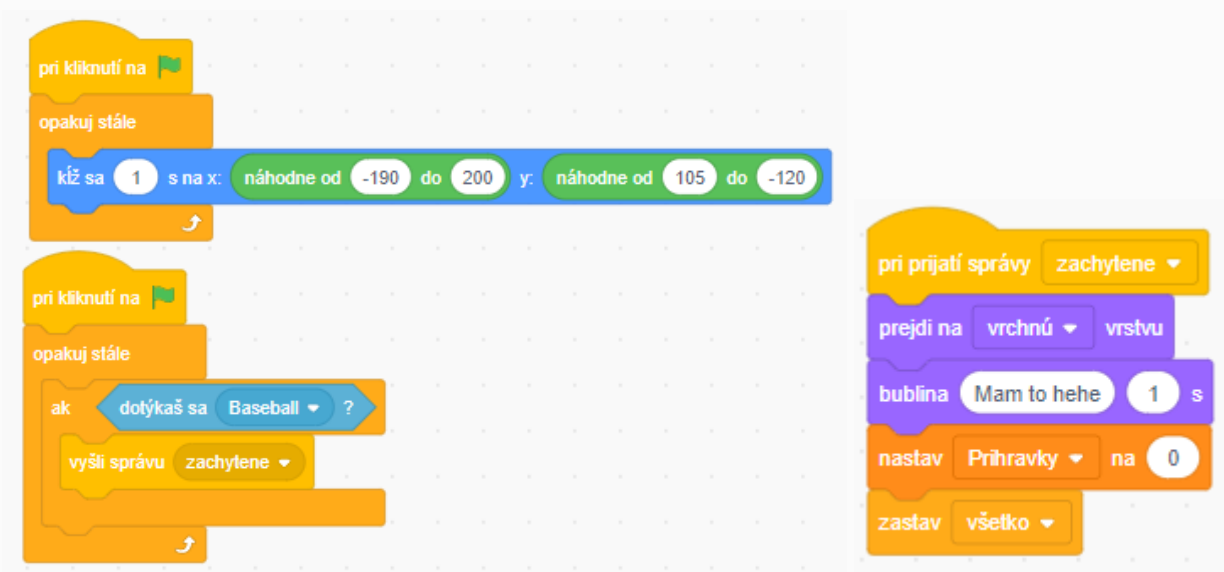
Objekt loptičky prijíma správy od jednotlivých mačiek a zvyšuje premennú Prihravky.



Objekty mačiek vysielajú správy. Taktiež majú nastavenú svoju počiatočnú polohu.

Chytač

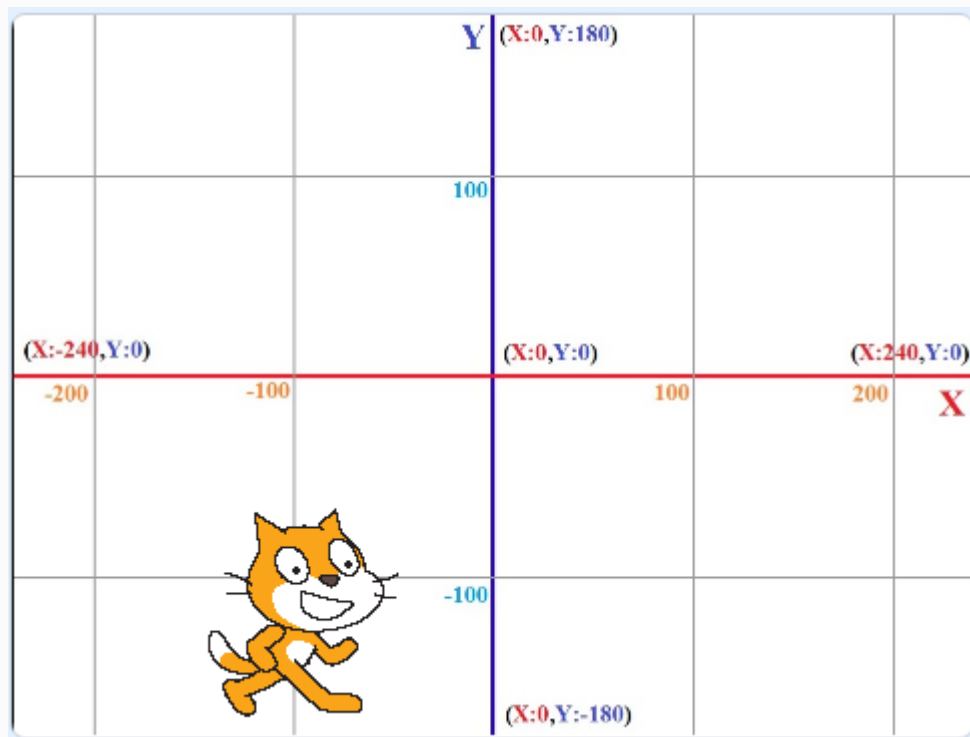
Pridáme postavu chytača, 5. mačky, ktorá sa bude pohybovať v priestore medzi ostatnými postavami. Na tejto postave môžeme vysvetliť súradnicovú sústavu, pretože vďaka nej presne vymedzujeme priestor, kde sa môže chytač hýbať. Konkrétne na to používame príkazy náhodne od – do z kategórie Operácie a dosadzujeme ich na miesta súradníc x, y. Ďalej kontrolujeme či sa chytač nedotkol loptičky, ak áno vyšleme správu zachytene, ktorú prijímame v postave lopty. Nastavujeme



Súradnicová sústava

Každá pozícia na obrazovke v Scratchi má svoju súradnicu, číselne vyjadrenie jej polohy. Súradnica má x-ovú zložku a y-ovú zložku.

Stred obrazovky má súradnicu x:0 y:0. X-ová súradnica je vodorovná a y-ová zvislá. Tieto súradnice vidíme aj v pravom dolnom rohu obrazovky v Scratchi.



Grafické efekty

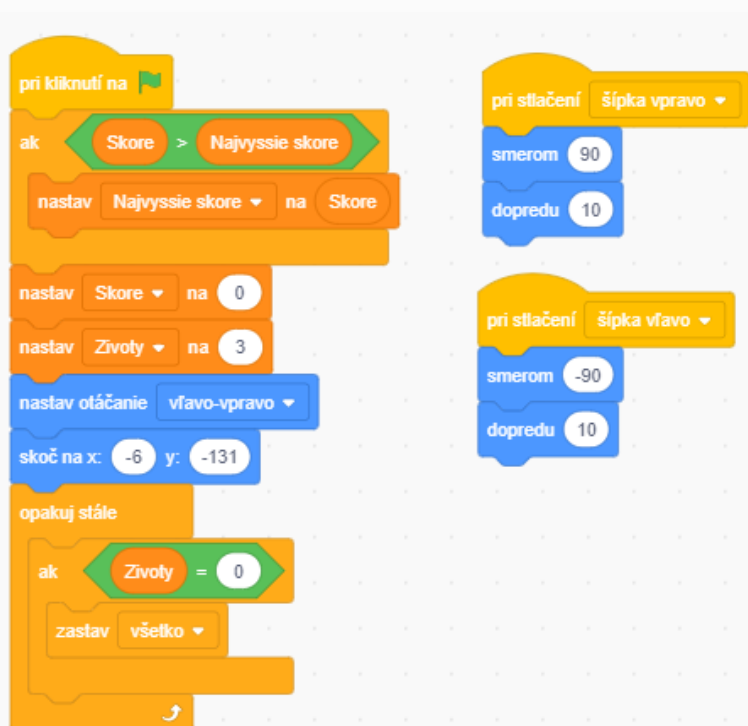
Názorne si ukážeme grafické efekty na kostýmoch na nasledujúcej úlohe.

Úloha – Čarodejnica

Úlohou je nazbierať čo najviac tekvíc, pokiaľ nám nedôjdu životy. Budeme si rátať skóre, uchovávať si naše najvyššie skóre a životy. Začnime prípravou čarodejnice.

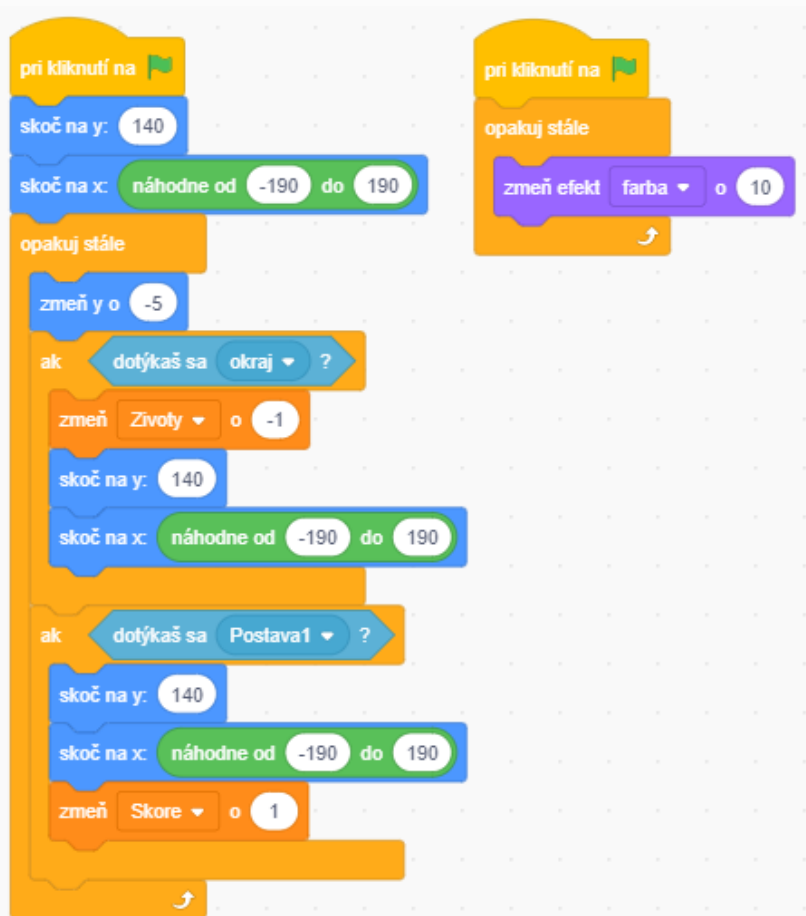


Čarodejnica sa musí vedieť hýbať zo strany na stranu. Taktiež jej dáme na starosť základné nastavenie našej hry. To znamená zapamätanie si najvyššieho skóre, počiatočné nastavenie skóre, životov a polohy. Tiež si kontroluje kedy už nemáme životy a treba hru ukončiť.



Následujú tekvice. Tekvice budú mať svoj grafický efekt – menenie farby. Chceme, aby tento efekt prebiehal stále, preto si ho vložíme do nekonečného cyklu. Takisto ako čarodejnica, aj tekvice musia mať nejakú počiatočnú polohu, ale nie presne rovnakú. Budú rovnako vysoko, ale pre rôzne umiestnenie budeme využívať náhodný výber od -190 do 190 pre x-ovú súradnicu.

Keď sa objavia hore, musia nejak klesať dole. Dojem približovania sa čarodejnice k tekviciam vytvoríme znižovaním y-ovej súradnice o 5 v nekonečnom cykle. Keď sa tekvica dotkne okraja pod čarodejnicou, strácame život a tekvica sa vracia naspäť hore. Keď sa dotkne čarodejnice, zvyšujeme skóre o 1 a vracia sa tiež naspäť hore.



Hra – Kľúč od brány

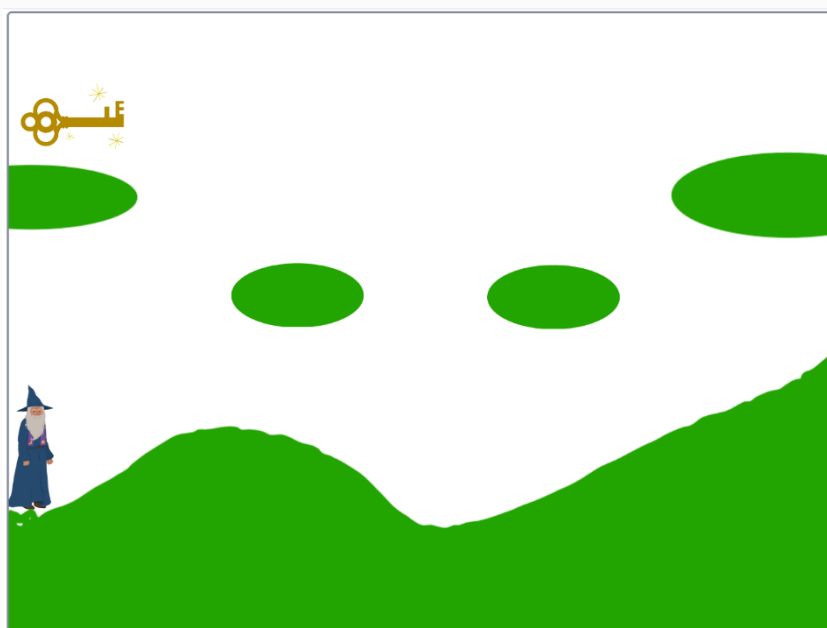
Čarodej si hľadá kľúč od záhadnej brány, ktorá sa zjaví iba tomu kto kľúč získa (ak vôbec existuje :o).

Čarodej sa pohybuje pomocou šípok vľavo, vpravo a pomocou šípky hore skáče

Postava chodí po zelenej. Vie vyskočiť a ostať stáť na lietajúcich plošinách, inak padá k zemi (gravitácia).

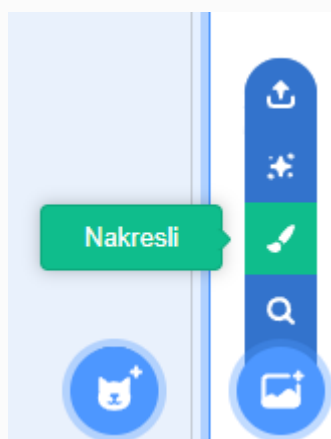
Po získaní kľúča sa nám ukáže brána na opačnej platforme.

Vstúpením do brány hra končí.
BONUS: Možnosť vytvoriť ďalšie pozadie ako nasledujúcu úroveň.



Kreslenie mapy

Zamýšľame sa nad tým o čo v tejto hre ide a ako to môžeme navrhnuť, aby to bola jednoduchá platformová skákačka.



Chceme nakresliť naše vlastné pozadie a použijeme na to nástroj štetec



, nástroj výplň



a nástroj elipsa



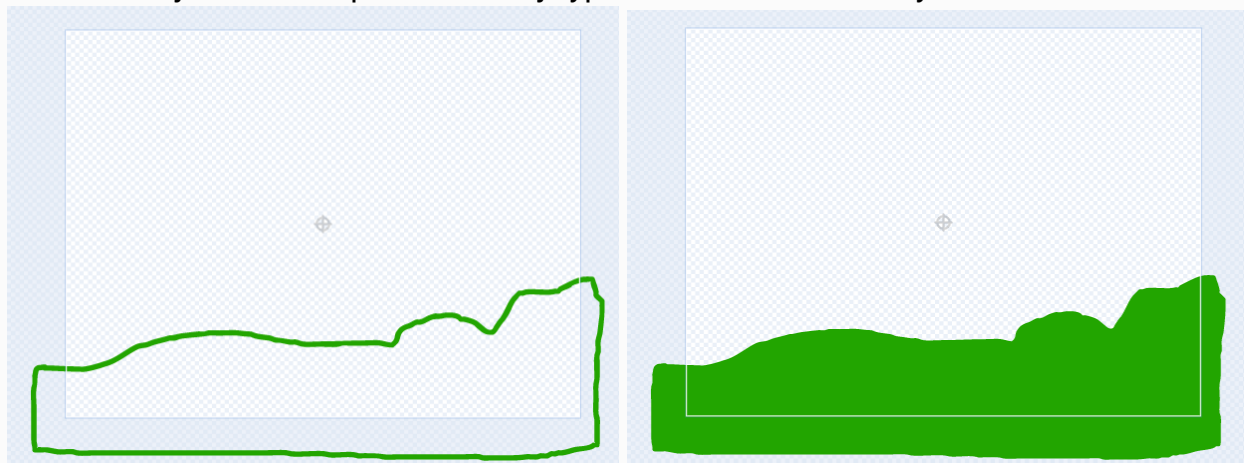
Veľmi dôležité je, že chceme všade použiť rovnakú zelenú farbu, aby sme mohli jednoducho vytvoriť gravitáciu (vysvetlené v postave Čarodej).

Vo výbere výplne som si namiešal tmavú zelenú farbu, každý si môžeme urobiť hocijakú zelenú, iba pripomenúť fakt, že ju chceme všade rovnakú. Farbu si vieme namiešať


kombináciou farby, jej sýtosti a jas.

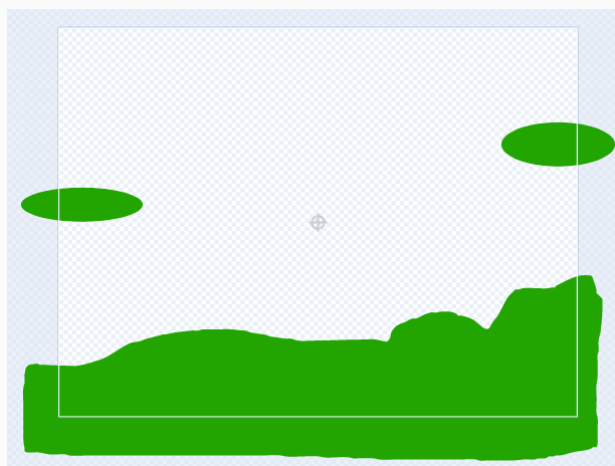



Začíname nákrešom obrysu spodnej časti mapy, ktorý uzatváram poza hranice nášho pozadia, aby som mohol jednoducho použiť nástroj výplň namiesto ručného vyfarbovania.



Pre vytvorenie nepohyblivých plôšín použijeme nástroj elipsa. Keď ho zaklikneme, vypneme obrys

označeným tlačidlom .



Ak sa stane, že nepoužívame všade rovnakú zelenú farbu použijeme  vo výbere výplne a následne na našom pozadí klikneme na farbu, ktorú chceme nastaviť.



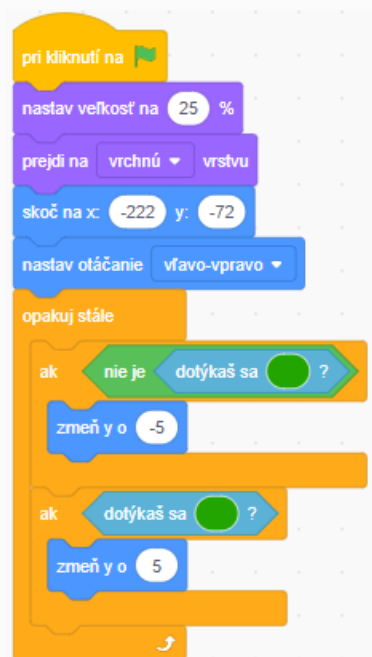
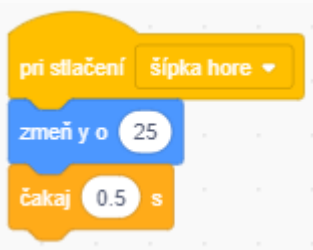
Pohyblivé plošiny



Plošiny po spustení hry (po kliknutí na vlajku) skočia na svoju začiatočnú pozíciu, odkiaľ sa budú neustále pohybovať. Postava – plosinay sa pohybuje vertikálne (zdola nahor), preto sa mení jeho Y. Postava – plosinax sa pohybuje horizontálne (zľava doprava), preto sa mení jeho X.

Čarodej

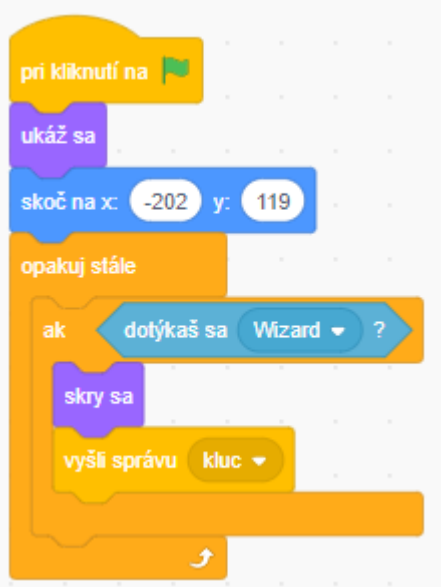
Nastavíme počiatočnú pozíciu, veľkosť, prechod na vrchnú vrstvu a otáčanie. Chceme aby sa čarodej pohyboval tesne nad zelenou (farba krajiny aj plošiny), preto ho posúvame trochu nahor (zmeň Y o 5). Inak ak sa čarodej nedotýka zelene padá k zemi (zmeň Y o -5), simulujeme tak gravitáciu.



Aby nám škriatok lepšie skákal aj v smere šípok ľavá a pravá, vytvoríme mu takýto pohyb (nie smerom + dopredu, ale smerom + zmeň x).

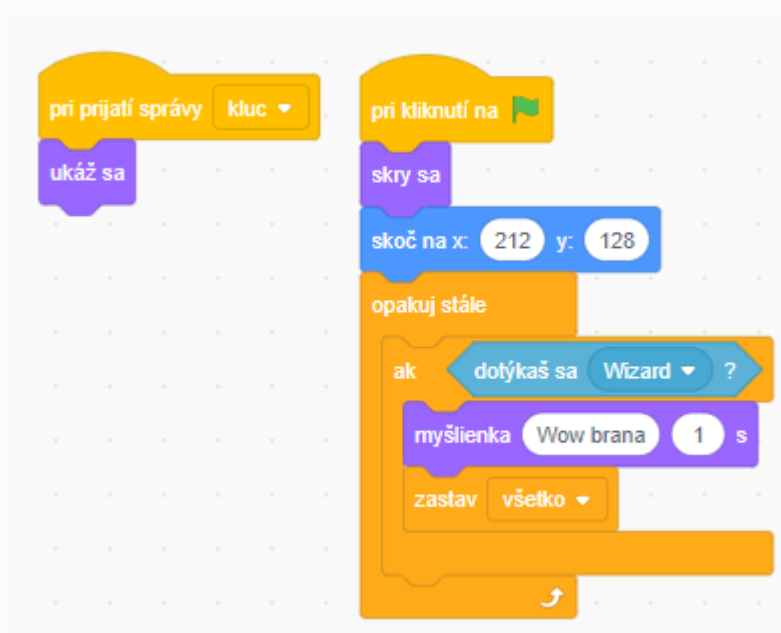
Kľúč od brány

Pri spustení hry chceme kľúč ukázať, nastaviť pozíciu a následne kontrolovať kedy sa ho dotkne postava čarodeja. Keď čarodej kľúč získa, skryje sa (čarodej ho zobral) a vyše správ.



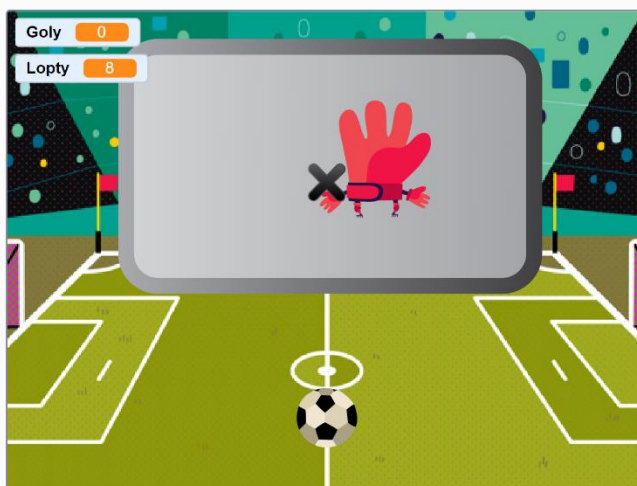
Brána

Pri prijatí správy kluc ju ukážeme, povery boli pravdivé a čarodej bránu vidí. Pri spustení hry postavu skrývame, nastavíme pozíciu a kontrolujeme, kedy sa jej dotkne čarodej a hru ukončíme. (Malá chyba, ktorú tu nechávam je, že bránu nemusíme vidieť, aby sa jej čarodej mohol dotknúť, takže v princípe stačí ísť rovno do brány. Ak si to niekto všimne tak ho určite pochváľte :D)



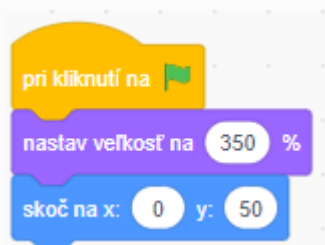
Penalty

Pomocou šípok posúvame po obrazovke terč. Stlačením medzerníku kopneme do lopty, ktorá poletí smerom na terč. Cieľom je trafiť sa do brány, v čom sa nám pokúša zabrániť brankár. Máme na to 8 pokusov a treba získať aspoň 5 bodov.



Brána

Iba nastavíme veľkosť na 350% a pozíciu.

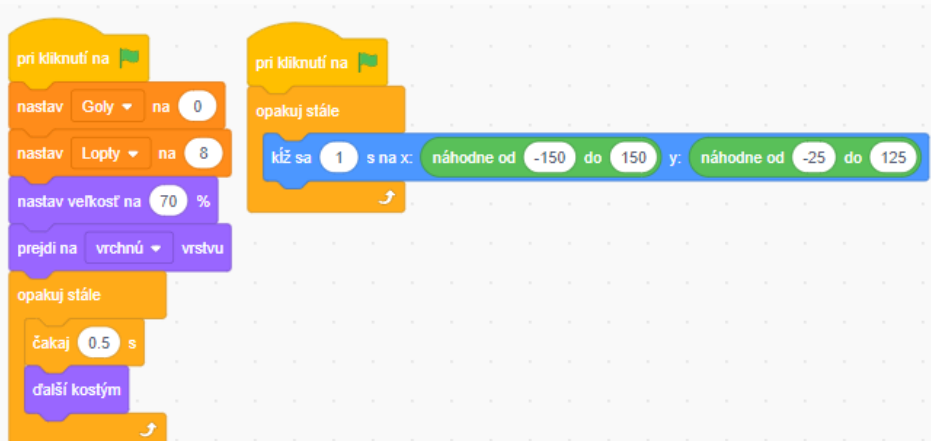


Brankár

Pohyb brankára

Postava brankára nastaví premenné Goly a Lopity na začiatkové hodnoty, svoju veľkosť na 70% a prejde na vrchnú vrstvu (hlavne pred bránu). Brankár má 2 kostýmy, ktoré budeme každých 0.5 sekúnd vymieňať.

Brankár sa bude pohybovať v oblasti bránky. Do buniek príkazu vyber náhodné číslo od ... do ... vložíme čísla polohy bránky. X – ľavý a pravý okraj brány, Y – dolný a horný okraj brány. Bránu si umiestňujem na nejaké rovné číslo, aby som dokázal aj ľahšie odhadnúť tie súradnice.

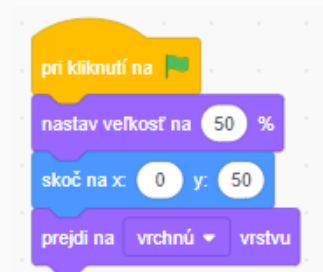


Ako ukázať podľa čoho ich vyberáme? Ja posúvam inú postavu ako bránu a ukazujem jej aktuálne súradnice pod obrazovkou hry pri oboch stranách brány.

K tejto postave sa ešte vrátíme keď budeme riešiť prijatie správ gol a vedľa.

Mieritko

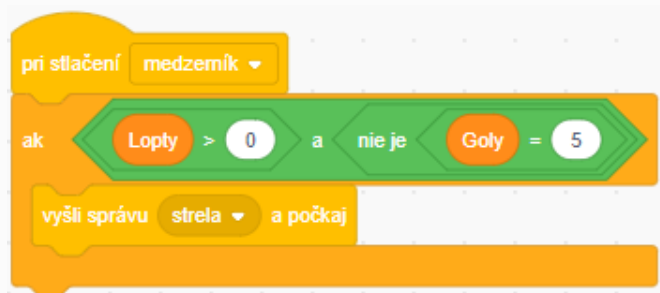
Nastavíme veľkosť postavy na 50%, pozíciu a prejdeme na vrchnú vrstvu.



Strel'ba do brány

Na strel'bu lopty do brány použijeme stlačenie klávesy medzerník. Ak je **Lopta > 0**, máme ešte voľne lopty a teda môžeme strieľať na bránu.

Pokiaľ nemáme 5 bodov a 0 lôpt chceme mať možnosť kopať. Preto použijeme zloženú podmienku **Lopty > 0 a Goly nie je = 5**. Ak to platí pošleme správu **strela** a počkáme.



Pre peknú animáciu letu lopty použijeme príkaz „kľž sa na postavu mieritko“. To budeme riešiť už v postave lopty.

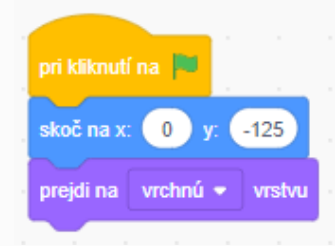
Pohyb mieritka

Na pohyb mieritka využijeme bežný spôsob pomocou šípok, aký sme už veľakrát používali.



Lopta

Nastavíme pozíciu postavy lopty a prejdeme na vrchnú vrstvu (nad bránu).



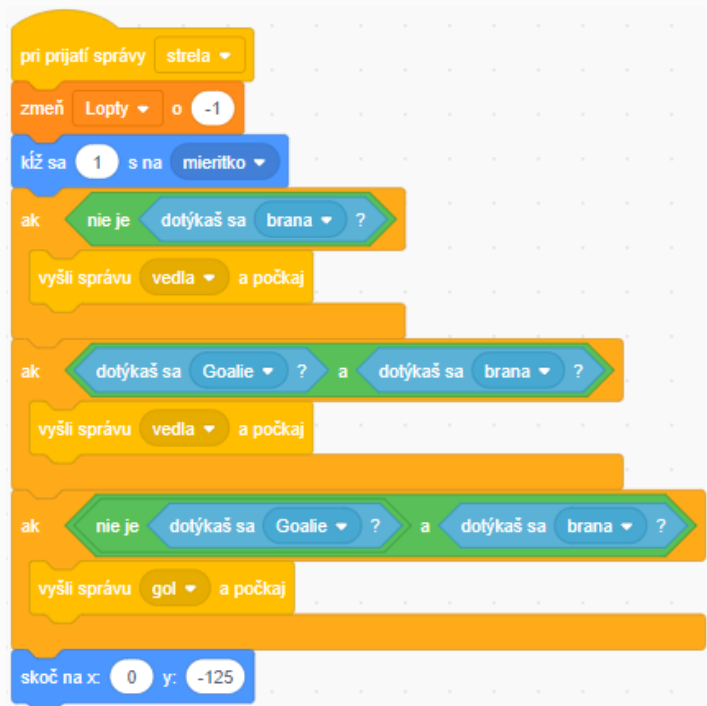
Strelba do brány

Máme voľné lopty a ešte sme nenahrali 5 bodov. Lopta poslala správu strieľam na bránu – strela.

- Loptu sme kopli do brány, preto nám jedna lopta ubudla => zmeň Lopty o -1
- Prejde na pozíciu mieritka

Môže sa stať:

- Lopta poletí mimo brány = nedotýka sa siete = dotýka sa siete? nie! Pošlem správu „vedľa“
- Lopta sa dotkne brány a nedotkne sa brankára. Dal som gól. Pošlem správu „gool“
- Lopta sa dotkla brány ale brankár ju chytil (dotkla sa brankára). Pošlem správu „vedľa“



Následne loptu vrátime na pôvodny bod.

Brankár

Sme späť v postave brankára, kde doplníme tieto prijatia správ, pretože dáva zmysel aby to či bol gól, vedľa alebo koniec hry oznámil brankár.

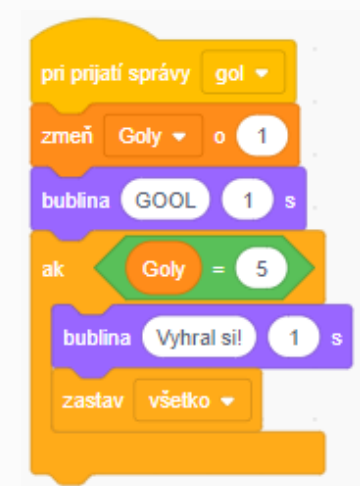
Strelba na bránu netrafil som – správa „vedla“

- Netrafil som sa, preto sa počet gólov nijako nezmení.
- Ak som sa netrafil do brány (správa „vedla“) a už nemám žiadne lopty (Lopty = 0), prehral som, oznámim to v bubline, presuniem brankára do stredu brány a zastavím všetko



Strelba na bránu trafil som – správa „gool“

- Úspešne sme dali gól, preto sa nám zvýši počet gólov
- Na 1 sekundu poviem: „GOOL“
- Dal som gól. Ak mám už 5 gólov (Goly = 5), tak som vyhral, oznámim to v bubline a zastavím všetko



Na dnešnej hodine si ukážeme **klonovanie** a **bloky**

Hra – Holuby

Našou úlohou bude odohnať holuby od dinosaura (alebo akejkoľvek inej postavy) predtým ako stihnú zašpniť jeho a celý chodník naokolo.



Príprava kostýmov a pozadia

Po výbere našej postavy potrebujeme, aby mala 5 ďalších kostýmov postupného zašpinenia od holubov. Jednoducho budeme vytvárať duplikáty predošlého kostýmu a pridať biele fláky. Rovnaký postup zopakujeme na pozadí.

Holub

V postave holuba (vybrali sme postavu holubice „Dove“ z knižnice postáv v Scratchi) použijeme **klonovanie** a **bloky**. Potrebujeme vytvoriť premenné „Odohnaných holubov“ a „Neodohnaných holubov“, ktoré budú pre všetky postavy. Ďalej vytvoríme premennú „Životy“, ktorá bude len pre postavu holuba (vďaka tomuto bude mať každý klon svoje vlastné životy). Vytvoríme bloky „Pociatocne nastavenia“, „Nový holub“, „Odohnať holuba“ a „Neodohnaný holub“.

Nová premenná

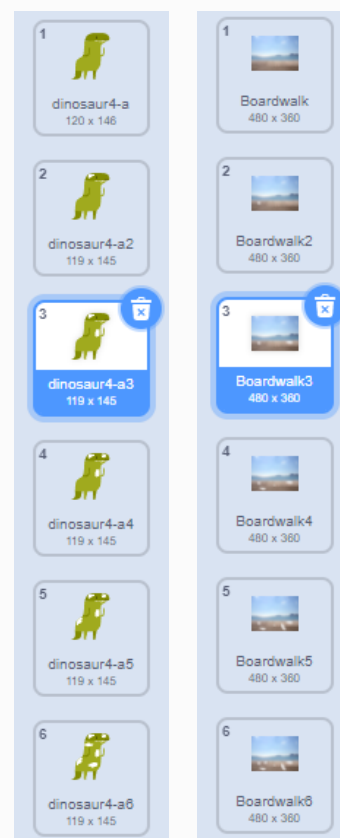
Meno premennej:

☐ Pre všetky postavy
 ☒ Len pre túto postavu

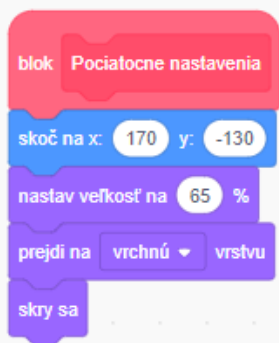
Odmietni

OK

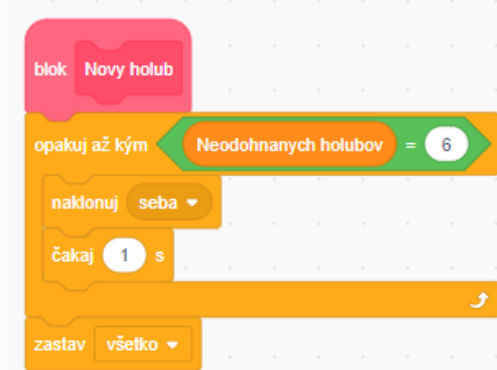
- ☒ Neodohnaných holubov
- ☒ Odohnaných holubov
- ☐ Životy



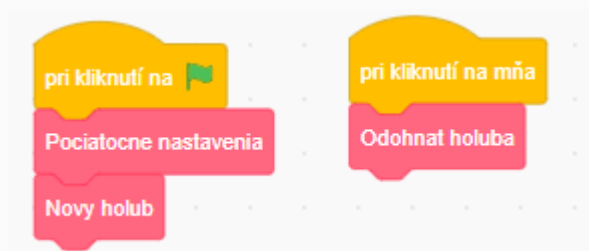
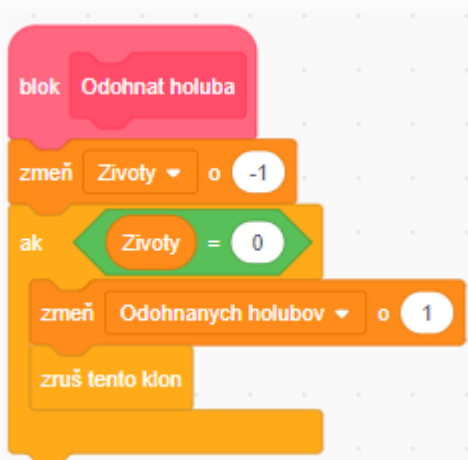
Blok „Pociatocne nastavenia“ bude obsahovať nasledovné príkazy.



Blok „Nový holub“ bude vykonávať **klonovanie** (z kategórie Riadenie) postavy v cykle typu **opakuj až kým** nedosiahne premenná „Neodohnaných holubov“ hodnotu 6, vtedy chceme hru zastaviť.

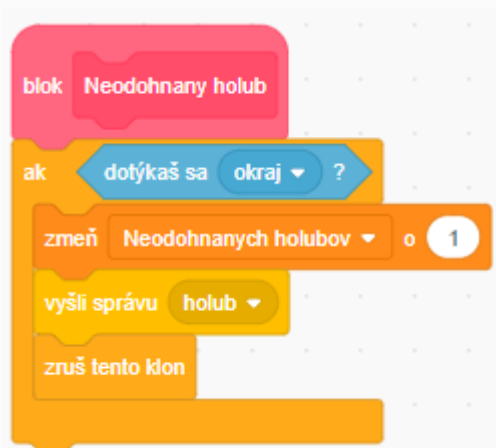


Blok „Odohnať holuba“ bude uberať životy klonov, mazať ich a upravovať premennú „Odohnaných holubov“.



Tieto tri bloky využijeme nasledovne.

Ostáva nám posledný blok „Neodohnaný holub“, ktorý slúži ako detekcia pre holubov, ktorý nám uletia na horný okraj obrazovky. Pri každom uletení vyšleme správu „holub“, ktorú budeme využívať v postave Dinosaura a pozadí.



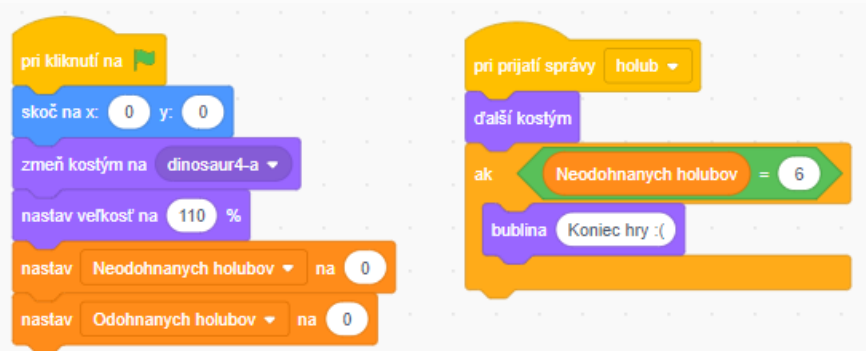


Pri klonovaní budeme využívať príkaz **Pri štarte klonu**, v ktorom vieme pre každý klon individuálne ovplyvniť jeho vlastnosti. Každý klon zvolí náhodnú pozíciu pri štarte a náhodný počet životov od 1 do 3. To v princípe znamená, že rôzne holuby budú potrebovať náhodný počet kliknutí, aby sme ich odohnali. Využívame tu náš blok “Neodohnany holub” za účelom, ktorý sme popísali pri jeho vytvorení.

Dinosaur

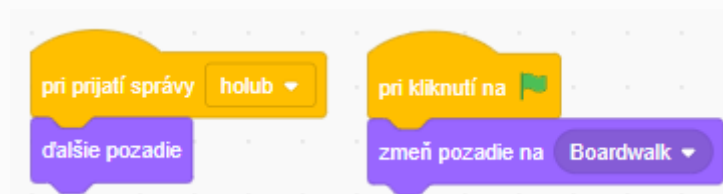
Táto postava obsahuje príkazy počiatočných nastavení vzhľadu a premenných, ktoré sme vytvorili. Ďalej tu využívame vyššie spomenutú správu “holub”. Vždy keď ju prijímame, vieme, že holub uletel a teda stihol

zašpiniť nášho dinosaura. Preto nastavujeme ďalší kostým a kontrolujeme stav premennej “Neodohnaných holubov”, aby sme na konci hry mohli vypísať bublinou s informáciou o tom.



Pozadie

Príkazy v pozadí sú veľmi priamočiare. Pri štarte hry nastaviť základne pozadie a pri prijatí správy “holub” nastaviť ďalšie pozadie.



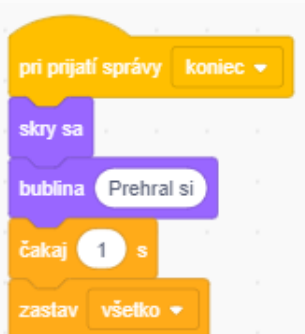
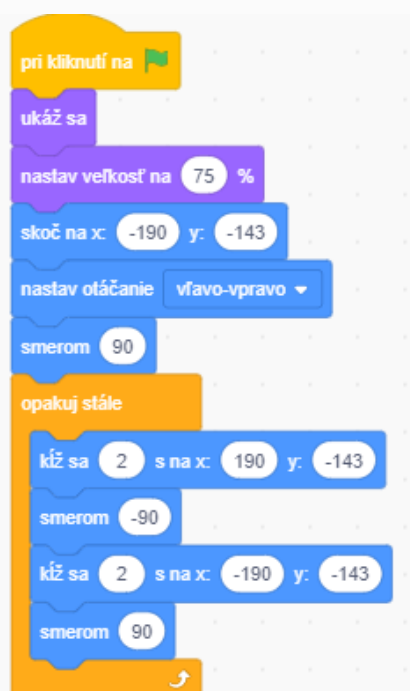
Hra – Ježko

Na ježka pod školou budeme zhadzovať jablká a zvony. Jablká chceme zhodiť tak aby ich chytil a zvony naopak tak aby ich nechytí. Za chytenie jablka pripočítame jedno chytenie, ak nechytíme jedno chytenie odpočítame. Ak zvon trafi ježka dostaneme -5 chytení jablák. V jednom kole zhadzujeme 5 predmetov, za každé splnené kolo dostaneme 1 bod.



Ježko

Najskôr si pripravíme postavu Ježka. Potrebujeme si klasicky pripraviť začiatkové nastavenia. S nimi spojíme rovno aj pohyb ježka zo strany na stranu.

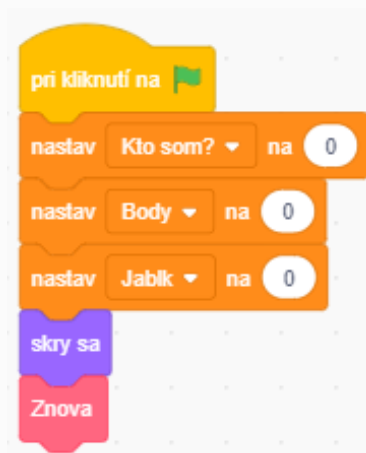


Keď hra skončí, chceme aby ježko ukázal bublinu s informáciou, že sme prehrali a skryl sa.

Jablko

Jablká, ktoré budeme zhadzovať budeme vytvárať práve klonovaním. Vďaka klonovaniu nepotrebujeme mať pre každé jablko zvlášť postavu ako doteraz, ale vytvoríme klon jednej postavy viac krát.

Pre začiatkové nastavenie použijeme tieto príkazy. Vidíme, že tu využívame nejaký príkaz „Znova“, ale nič také v Scratchi neexistuje. Príkaz „Znova“ je nami vytvorený blok príkazov, ktoré môžeme jednoducho použiť takýmto spôsobom.

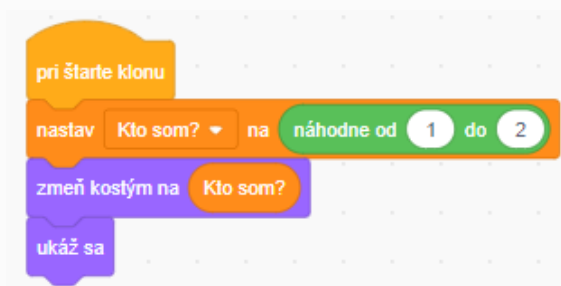


Do bloku vieme dať rôzne príkazy a inde ich všetky zavolať znova práve cez nami vytvorený blok. Zíjde sa nám to buď keď rovnaké príkazy používame viac krát na iných miestach alebo na sprehľadnenie celej štruktúry.

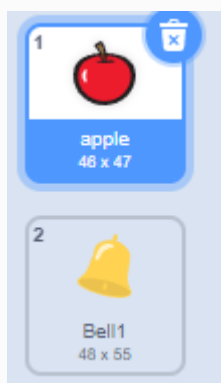


Blok „Znova“ voláme vždy keď chceme vytvoriť nových 5 klonov, po tom keď sme predošlých 5 už vyhodili. Preto nastavujeme premennú Odhodene na 0, ideme na počiatočnú pozíciu a klonujeme.

Klony vieme ovládať príkazom „pri štarte klonu“.



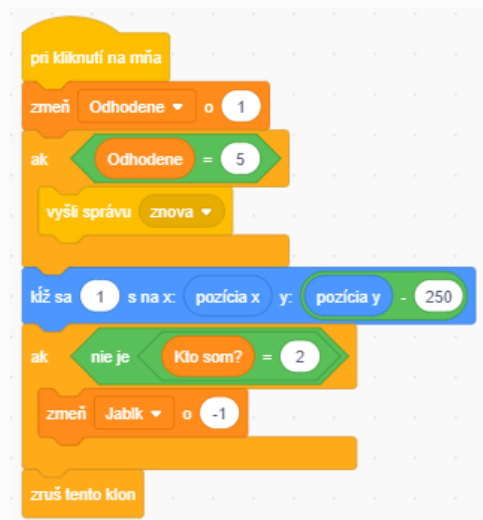
Pri každom klone vykonávame následovné príkazy. Premenná Kto som? je len pre túto postavu, aby každý klon mal svoju vlastnú premennú. Premennú „Kto som?“ využívame ako náhodnú voľbu či sa nám objaví jablko alebo zvonec. To čo je 1 a čo je 2 rozlišujeme podľa čísiel kostýmov v postave (poradie môže byť opačné, namiesto 1 = jablko by to mohol byť zvonec, pozor na to)



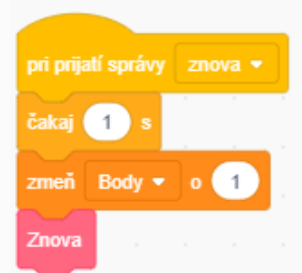
Následne už klony iba ukážeme a príkazom klzania vytvoríme animáciu „položenia“ predmetu na strechu.

Zhadzovanie predmetov

Naklonované jablká/zvony musíme teraz zhadzovať na ježka.



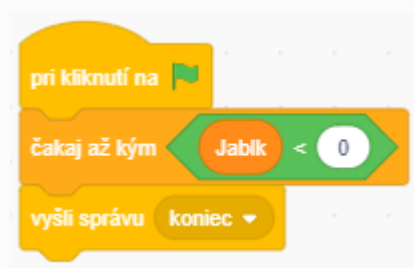
Keď klikneme na jeden z klonov, zvyšujeme si premennú odhodene, aby sme vedeli kedy potrebujeme vytvoriť nové klony. Klon klesá dole a následne podľa premennej Kto som? zisťujem či to bol zvon alebo jablko. Ak to bolo jablko odpočítavame jedno chytené jablko. Následne klon mažeme.



Správa znova nám zvýši premennú Body, keďže za každé kolo máme získať jeden bod a zavolá blok Znova.

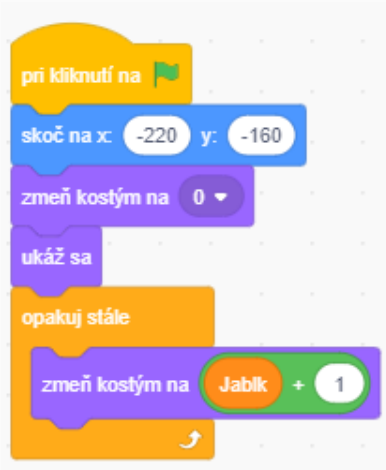


Pri kliknutí na klon musíme aj kontrolovať či sa dotkol postavy Jezko a podľa toho či je klon jablko alebo zvon upravujeme premennú Jablk. Potom tento klon mažeme.



Hru ukončíme keď budeme mať menej jablák ako 0 cez správu koniec.

BONUS: Jednoduché grafické počítadlo



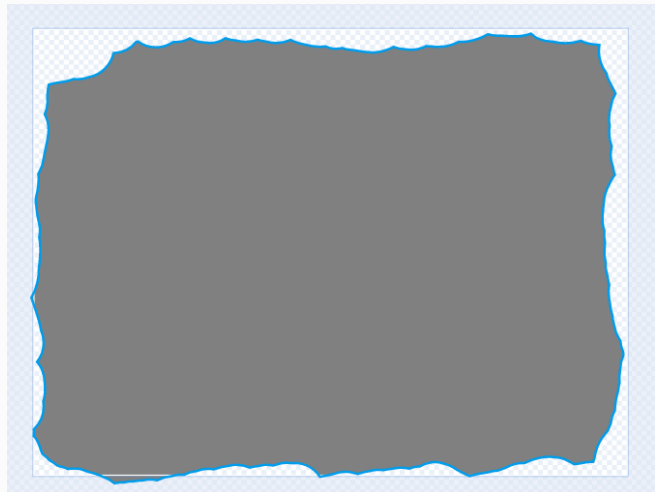
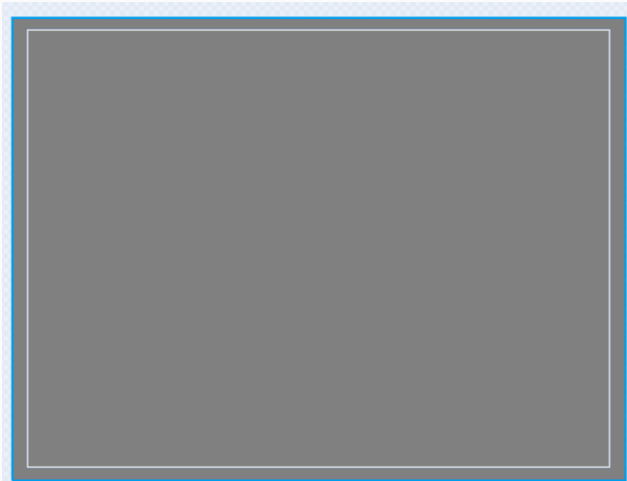
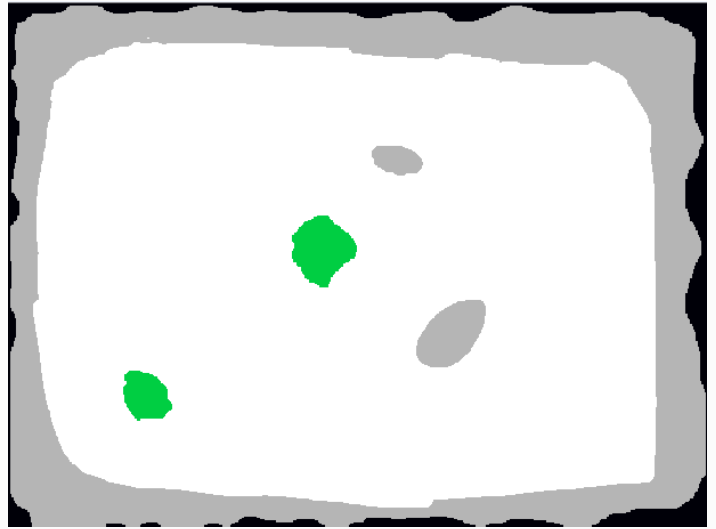
Postava Pocitadlo jablk nám v ľavom dolnom rohu počíta vždy po päť jablk. Kostým meníme pomocou premennej Jablk, ktorá nastaví číslo kostýmu.

Hra – Auto dráha

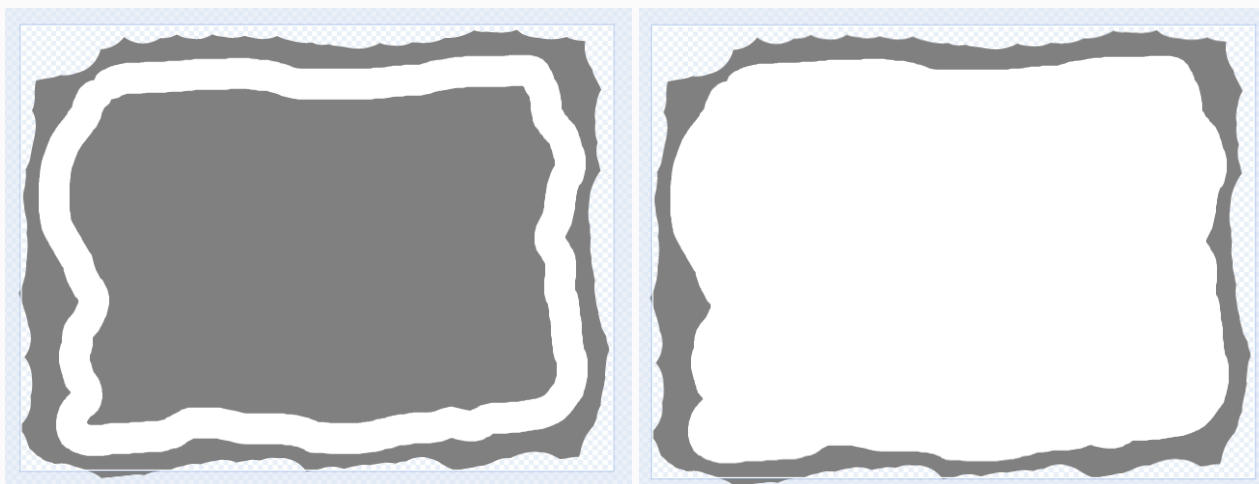
Nakreslenie obrazovky

Dráha, ktorá je naše pozadie bude vytvorená ako postava a nami nakreslený kostým v nej. Samotné pozadie (scénu) celú zakryjeme čisto čiernym (farba 0, sýtosť 0, jas 0) obdĺžnikom, aby pod našou dráhou bola čierna farba.

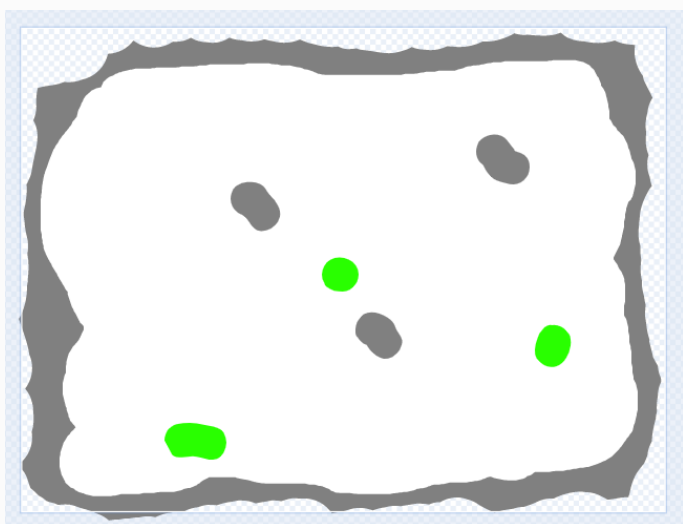
Pri vytváraní dráhy v postave začneme šedým obdĺžnikom (farba 0, sýtosť 0, jas 50), ktorého okraje „opálime“ nástrojom guma a nastavenou veľkosťou 100.



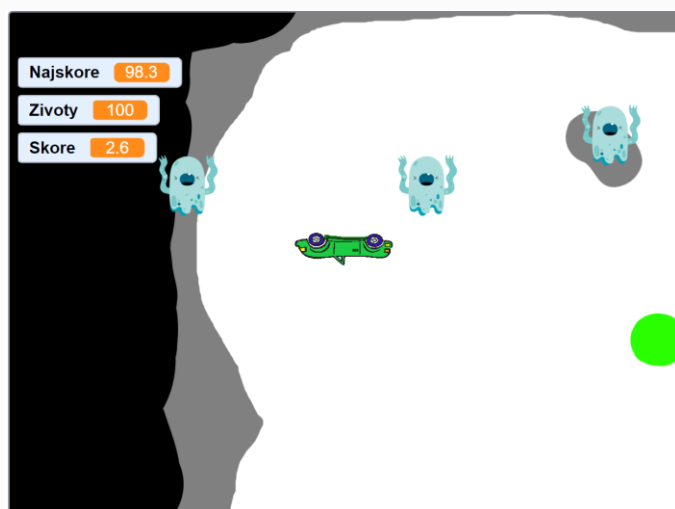
Keďže pozadie (scénu) sme zafarbili načierno a my chceme mať vnútro drahy biele, nástrojom štetec vytvoríme biele (farba 0, sýtosť 0, jas 100) obkruženie, ktoré vyfarbíme nástrojom výplň.



Už nám chýbajú iba malé "mláčky", ktoré uberajú životy. Sú 2 farby mláčok, šedá, akú sme už použili a zelená, ktorú si zvolíme. Mláčky nakreslíme jednoducho nástrojom štetec do plochy.



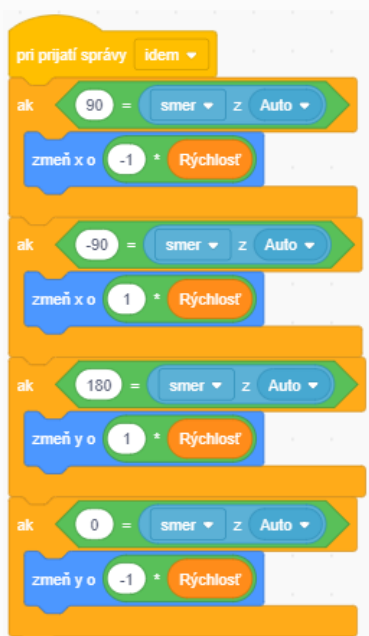
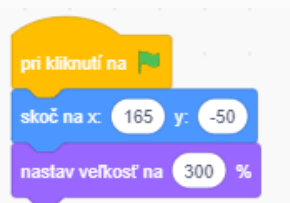
Ukážka aj s čiernym pozadím v scéne.



Pohyb obrazovky

Doteraz sme hýbali konkrétne s tou postavou, ktorú sme chceli posúvať. V tejto hre zvolíme iný prístup. Posúvať budeme postavu samotnej dráhy.

Toto je postava „Draha“, túto postavu zvážšime o 300%, tým pádom nebude celá rovno ukázaná, ale budeme sa musieť ňou posúvať a to vytvorí ilúziu pohybu auta, aj keď v skutočnosti posúvame pozadie.



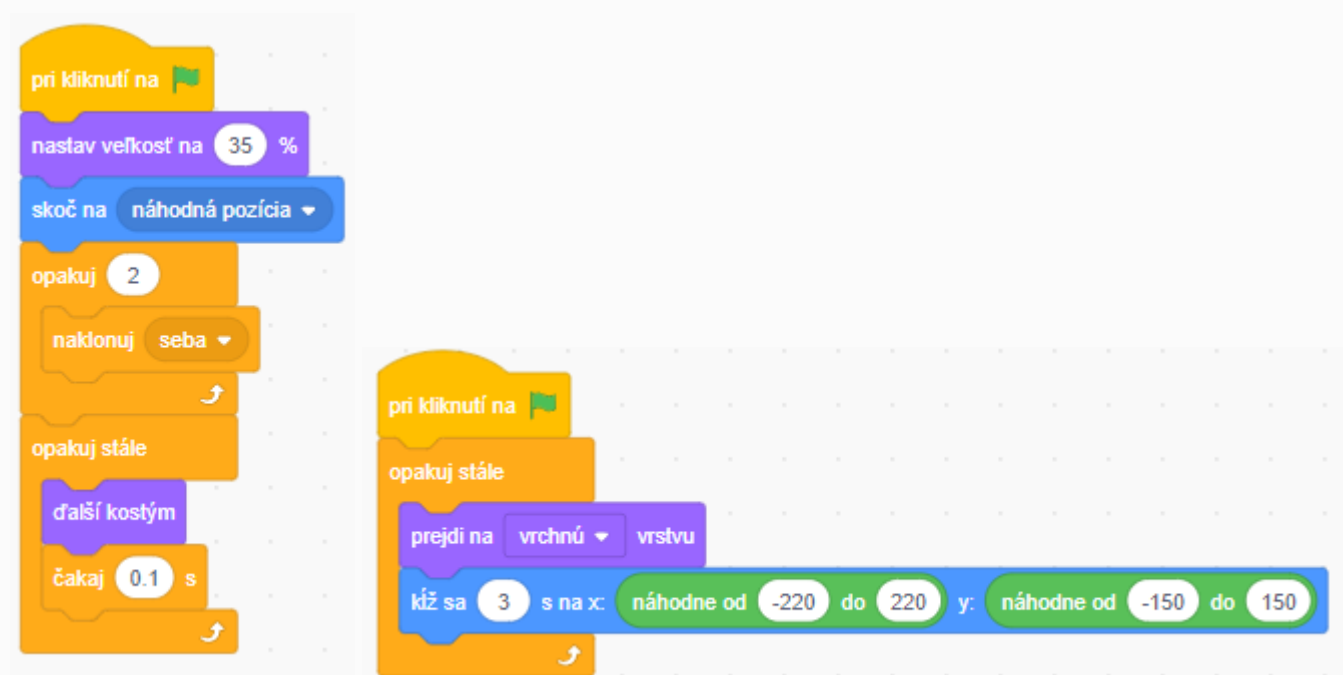
Veľmi **dôležitá** vec pri pohybe postavy „Draha“ je, že ju musíme posúvať opačným smerom, ako je smer, do ktorého chceme ísť autom.

Prečo? Znova ide o to ako vnímame keď sa hýbeme niekam nejakým smerom. Ak „Auto“ ide doprava, to znamená, že „Draha“ nám musí z ľavej strany miznúť preč a z pravej strany pribúdať. Takže „Drahu“ musíme posúvať smerom doľava cez súradnicu x. Rovnaký princíp je pre posun hore a dole. Z kategórie Zisťovanie, vieme použiť príkaz „smer z Auto“, ktorým vieme zistiť do akého smeru sa točíme bez ďalších správ navyše.

V postave „Auto“, ale aj tak musíme vytvoriť ovládanie, ktorým sa budeme posúvať. Pre „Auto“ nastavíme počiatočné nastavenia a tu využijeme funkciu blokov v Scratchi.

Duchovia – ovplyvňovanie životov

Klony postavy ducha sa taktiež budú prechádzať po dráhe a pokiaľ sa budú dotýkať auta, budú nám uberať životy.

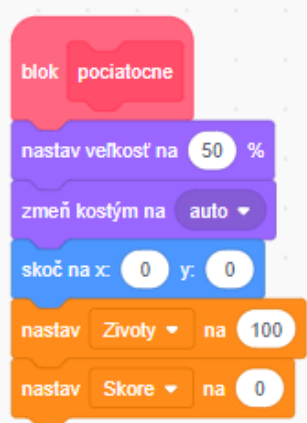


Duchov zmenšíme na 35% veľkosť, naklonujeme a meníme medzi kostýmami. Duchovia sa budú náhodne kľzať po ploche, ktorú vidíme. Zároveň ich však bude aj ovplyvňovať, to kam ideme s autom pomocou bloku „po prijatí správy idem“, ktorý je rovnaký ako bol v postave „Draha“.



Klony robia to isté ako hlavná postava (prvé príkazy).

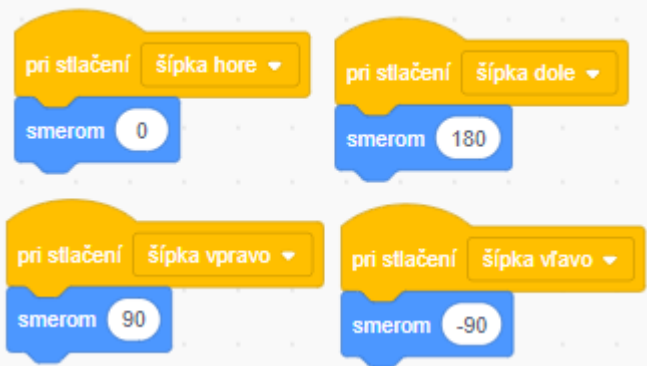
Auto



Do bloku môžeme vložiť ľubovoľné príkazy a všetky ich vieme vykonať hocikde pomocou bloku, ktorý sme si my nazvali „Pociatocne nastavenie“. Najviac užitočné sú bloky, pokiaľ chceme opakovať nejaké príkazy na rôznych miestach viackrát, ale sú veľmi nápomocné aj pre sprehládnenie všetkých príkazov.

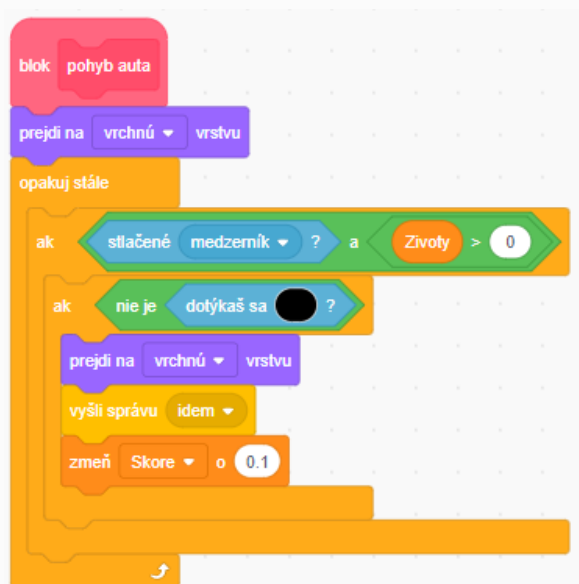
V bloku „Pociatocne nastavenie“ nastavujeme veľkosť kostýmu, konkrétny kostým, počiatocnú polohu a niekoľko premenných na počiatocné hodnoty. Účel premenných bude vysvetlený ďalej v úlohe.

Ovládanie „Auta“ vytvoríme klasickým spôsobom, ale iba pomocou príkazov smerom.



Blok „Príkaz pohybu auta“ nás dá do popredia a pokiaľ stlačíme medzerník, máme stále viac životov ako 0 a nedotýkame sa čierneho okraja dráhy vyšleme správu **idem** a pripočítavame skóre do premennej Skore.

Správu **idem** prijímame v postave „Draha“ a „Duch“.

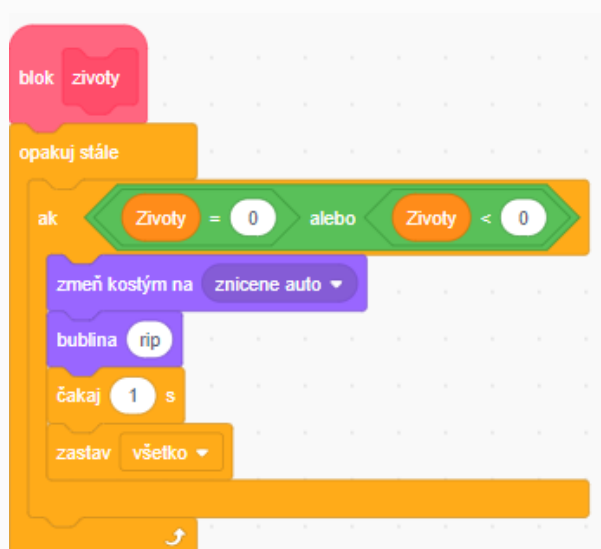


Životy auta a získavanie skóre

V bloku „Pociatocne nastavenie“ sme si nastavili životy na 100. Životy nám ubúdajú pokiaľ sa dotýkame, šedej plochy dráhy, zelenej kaluže na dráhe alebo duchov. Na detekciu čoho sa dotýkame si tiež vytvárame blok s názvom „Auto sa dotýka?“. Podľa toho, čoho sa dotýka odčítavame rôzny počet životov. Tento blok musíme následne aj využívať po zapnutí hry.



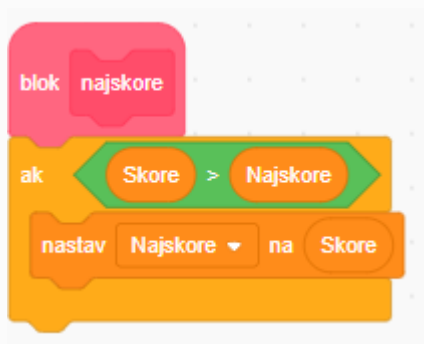
Musíme stále však kontrolovať aj či stále máme životy a pokiaľ nie, ukončiť hru. Vytvoríme si blok „Kontrola zivotov“.



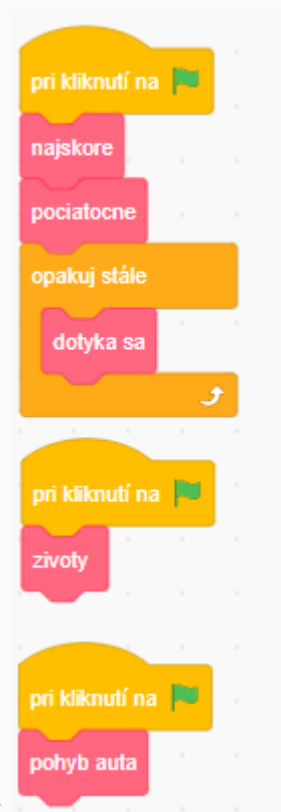
Ak máme 0 alebo menej životov, zmeníme na kostým „newcar2“ (zničené auto), vypíšeme, že sa nám rozbilo auto a zastavíme celú hru.

Najlepšie skóre

Posledný detail hry bude ukazovanie a ukladanie najlepšieho skóre. Vytvoríme blok „Nastavenie najlepšieho skóre“.



Chceme toto skóre ukázať a následne skontrolovať, keď skončí hra, či aktuálne skóre je lepšie ako naše doteraz uložené najlepšie skóre. Ak áno, uložíme si naše skóre do premennej „Naj skóre“.



Všetky bloky v postave Auto využívame následovne:

Pinball

Táto úloha využíva všetky znalosti, ktoré sme sa doteraz naučili a je pomerne jednoduchá, preto dostanete tento postup bez ukážky príkazov, ktoré používame.

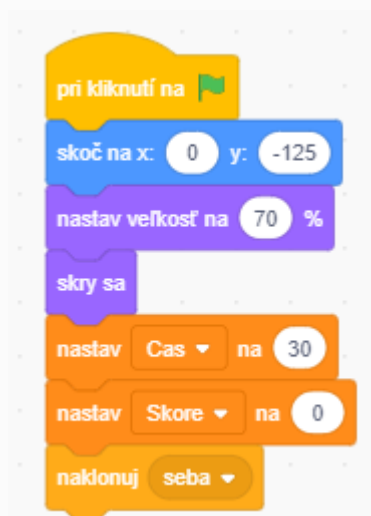


Plošinou, ktorú ovládame myškou odrážame loptu, aby nespadla dole. Máme čas 30 sekúnd, ak zoberieme loptou zvonček, máme o 1 sekundu viac. Ak lopta zoberie motýľa, dostávame +5 skóre.

Zhrnutie funkcií jednotlivých postáv:

- Postava **plosina** má za úlohu sa hýbať zo strany na stranu pomocou pohybu myši
- Postava **lopta** bude odrážaná od plošiny a je to aj jej jediná funkcia
- Postava **zvonec** má za úlohu pri dotyku s loptou pripočítať 5 skóre
- Postava **motyl** má za úlohu pri dotyku s loptou pripočítať 1 sekundu k času
- Postava **skore+** má za úlohu ukázať „+5“ a vykonať jednoduchú animáciu
- Postava **cas+** má za úlohu ukázať „+1“ a vykonať jednoduchú animáciu

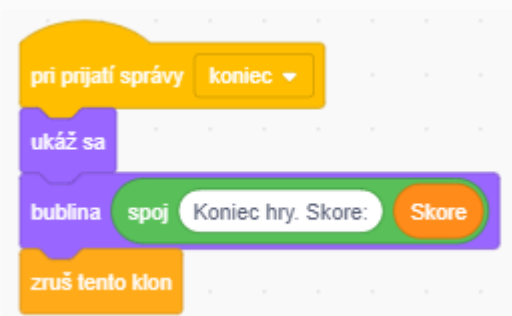
Plošina



Budeme nastavovať počiatočné nastavenia pri stlačení na zelenú vlajočku. Počiatočná pozícia x: 0, y: -125, veľkosť postavy 70%, skryť sa, nastaviť premennú Cas na 30, nastaviť premennú Skore na 0 a naklonovať sa.

Pri štarte klonu sa ukážeme a opakujeme stále nastavenie súradnice x postavy podľa súradnice x myšky (sekcia Zisťovanie). V ďalšom bloku pri štarte klonu opakujeme stále príkazy čakaj 1 sekundu, zmeníme premennú Cas o -1 a vložíme podmienku ak je premenná

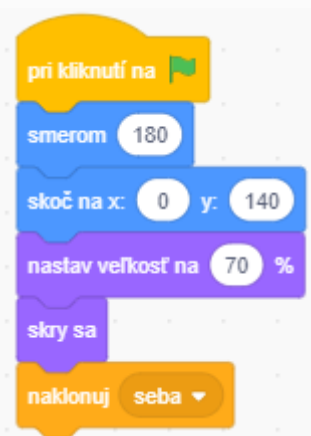
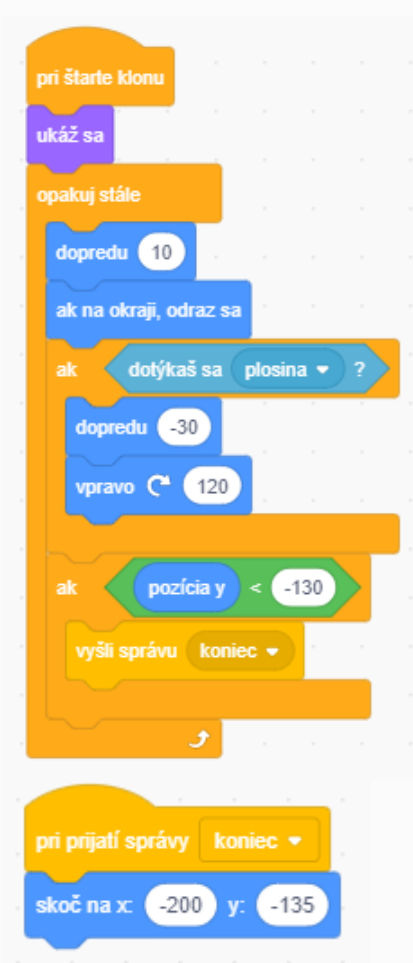
Cas menej ako 0, vyšli správu „koniec“. Tento druhý blok „pri štarte klonu“ nám slúži ako časovač hry.



Pri prijatí správy „koniec“ sa ukážeme, vypíšeme cez bublinu správu o konci hry + aké skóre sme dosiahli (sekcia Operácie, príkaz spoj) a zrušíme tento klon.

Lopta

Klasicky začneme s počiatočnými nastaveniami pri stlačení na zelenú vlajočku. Otočíme loptu smerom 180, skočíme na x: 0, y: 140, nastavíme veľkosť na 70%, skryjeme sa a naklonujeme sa.



Pri štarte klonu sa ukážeme a opakujeme stále nasledovné príkazy: dopredu 10 krokov, ak na okraji, odraz sa, vložíme podmienku kde sa pýtame či sa dotýkame postavy plosina. Pokiaľ áno ideme dopredu o -30 krokov a otočíme sa vpravo o 120 stupňov. Loptu preto posúvame hneď o -30 krokov, pretože ináč by ostala zaseknutá na plošine. Po tejto podmienke vkladáme ďalšiu podmienku, kde sa pýtame či je pozícia y našej postavy menej ako -130. Ak áno, vyšleme správu koniec.

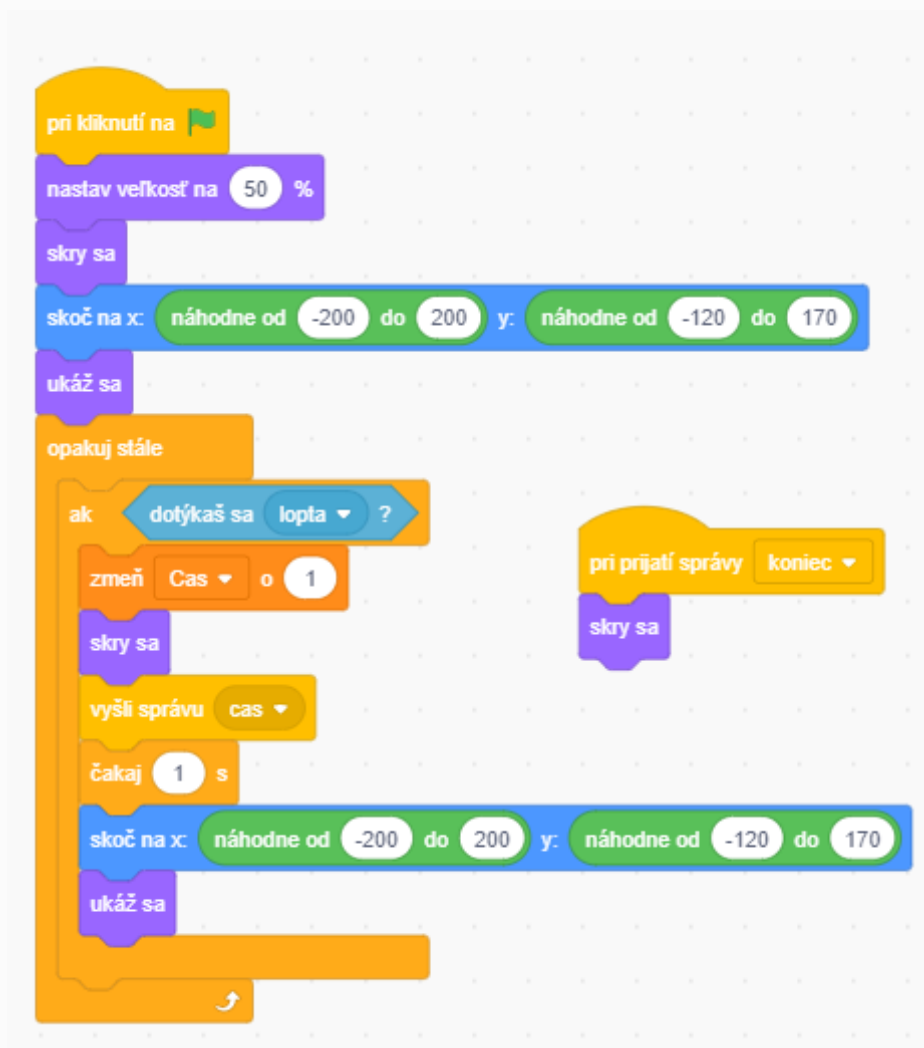
Pri prijatí správy koniec iba skočíme na x: -200, y: -135.

Zvonec

Pri stlačení na zelenú vlajočku nastavujeme veľkosť na 50%, skryjeme sa, skočíme na náhodne x od -200 do 200 a na náhodne y od -120 do 170 a ukážeme sa.

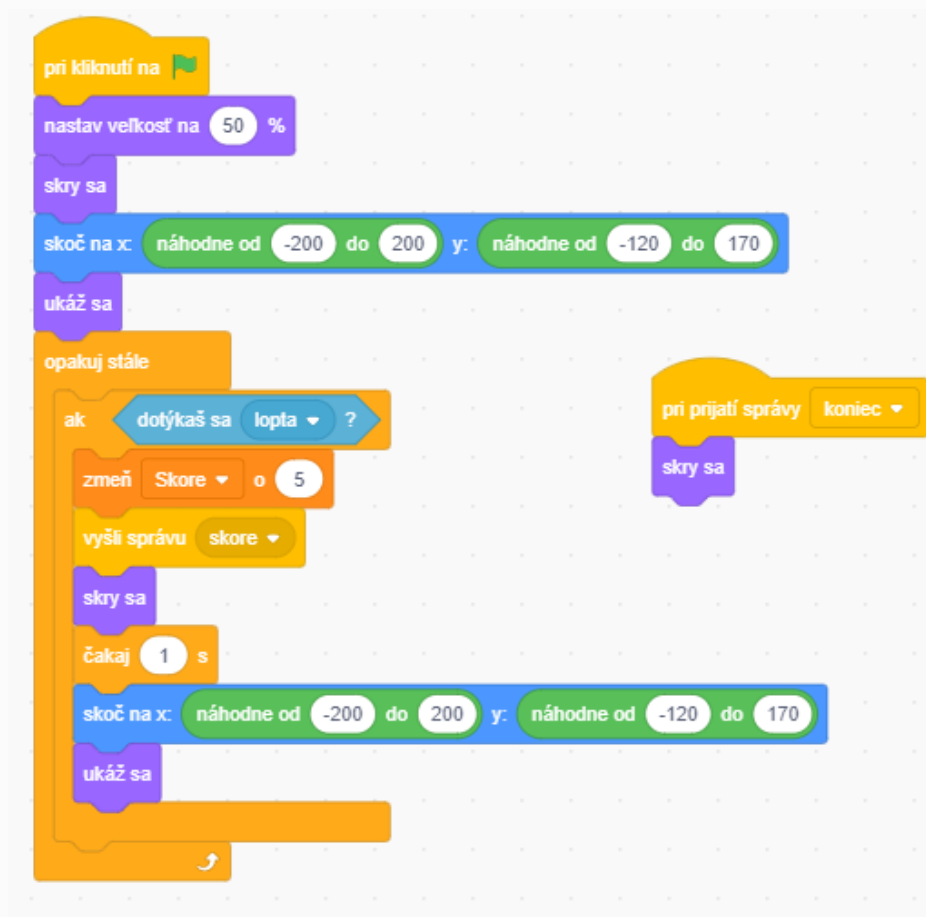
Následne opakujeme stále nasledovné príkazy: vložíme podmienku či sa dotýkame postavy lopta. Ak áno, zmeníme premennú Cas o 1, skryjeme sa, vyšli správu „cas“, počkáme 1 sekundu, skočíme na náhodne x od -200 do 200 a na náhodne y od -120 do 170 a ukážeme sa.

Pri prijatí správy koniec sa skryjeme.



Motýľ

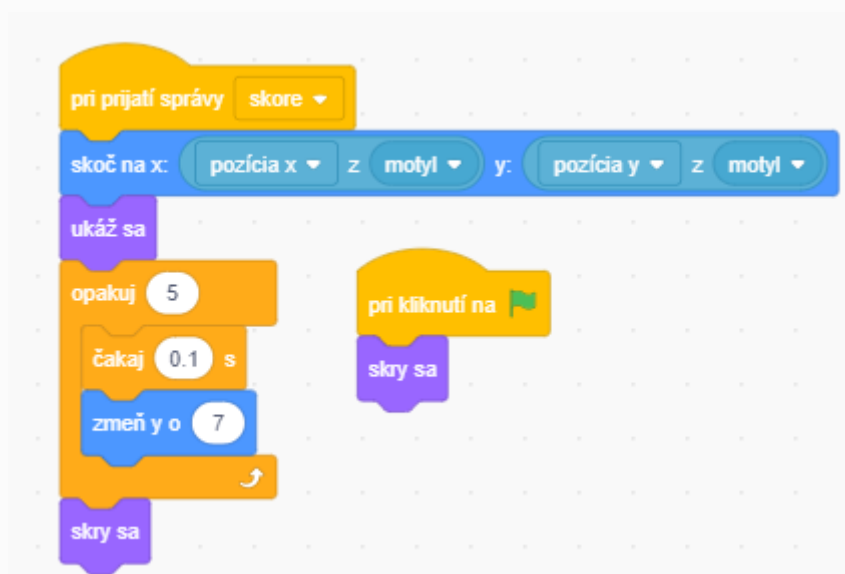
Motýľ má presne rovnakú funkčnosť ako zvonec, menia sa iba 2 veci. Meníme premennú Skóre o 5 a vysielame správu „skóre“. Ostatné príkazy sú presne identické so Zvoncom.



Postava „+5 skóre“

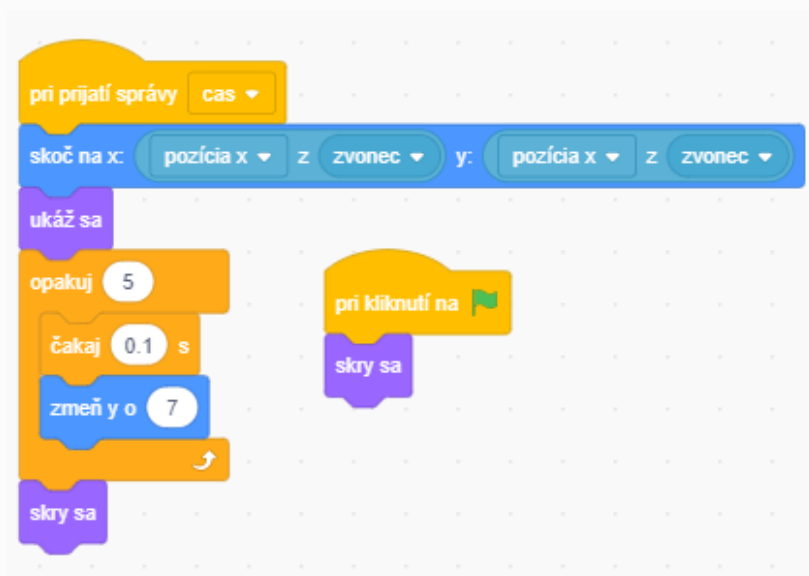
Pri kliknutí na zelenú vlajočku sa iba skryjeme.

Pri prijatí správy „skóre“ skočíme na súradnice x, y postavy motýľa (sekcia Zisťovanie), ukážeme sa a 5 opakujeme príkaz čakaj 0,1 sekúnd a zmeň y o 7. Tento malý cyklus nám vytvorí jednoduchú animáciu stúpania. Po cykle sa skryjeme.



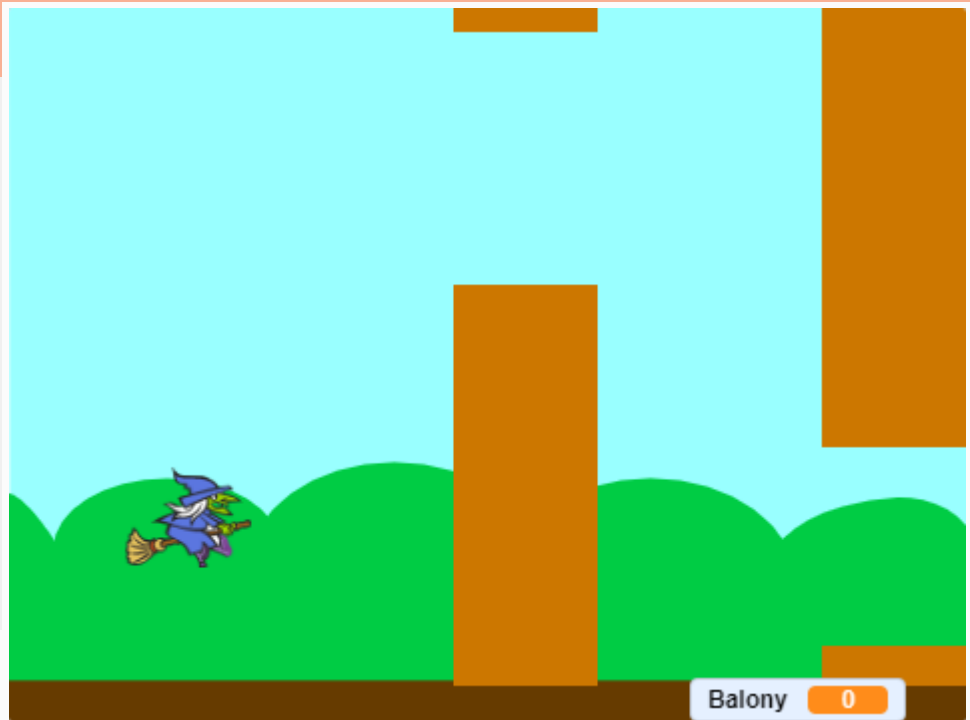
Postava „+1 cas“

Postava „+1 cas“ má presne rovnakú funkcionálnosť ako postava „+5 skóre“. Jediný rozdiel je, že skočíme na súradnice x, y postavy zvonca a nie motýľa.

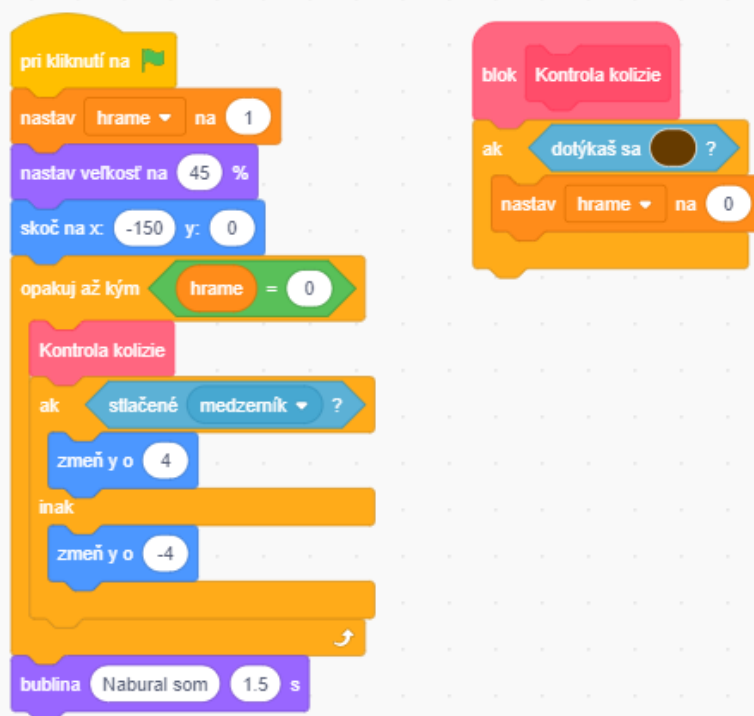


Flappy Witch

Dnešná úloha bude napodobenina slávnej hry Flappy Bird. Čarodejnicou budeme stúpať držaním medzerníka. Musíme sa vyhýbať prekážkam a zároveň nepadnúť na zem.



Čarodejnica

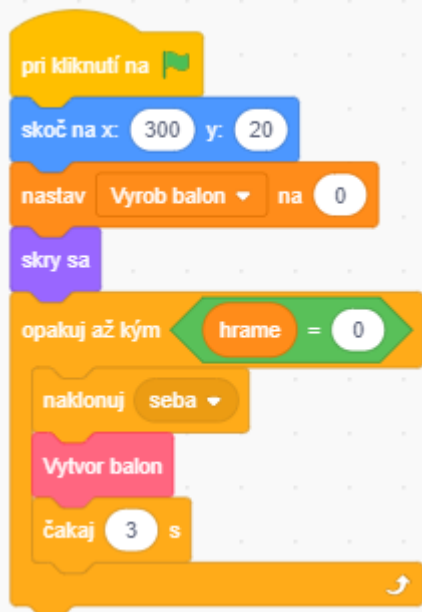


Počiatkové nastavenia sú nasledovné. Premennú „hrame“ nastavíme na 1. Táto premenná nám riadi či hra stále beží alebo ju ukončíme. Keď je 1 hráme, keď 0 hra skončila. Nastavíme veľkosť na 45%, skočíme na x: -150, y: 0.

Po nastaveniach vložíme cyklus typu opakuj až kým sa premenná „hrame“ nebude rovnať 0. Vo vnútri používame vytvorený blok „Kontrola kolízie“. Tento blok kontroluje či sa dotýkame zeme. Ak áno nastavíme premenné hrame na 0 a hra končí. V cykle pokračujeme zloženou podmienkou ak je medzerník stlačený => zmeň y o 4, inak zmeň y o -4. Takže pokiaľ držíme medzerník stúpame, keď ho pustíme klesáme.

Keď z cyklu vyjdeme, vieme, že hra skončila tak vypíšeme bublinu s textom „Nabural som“.

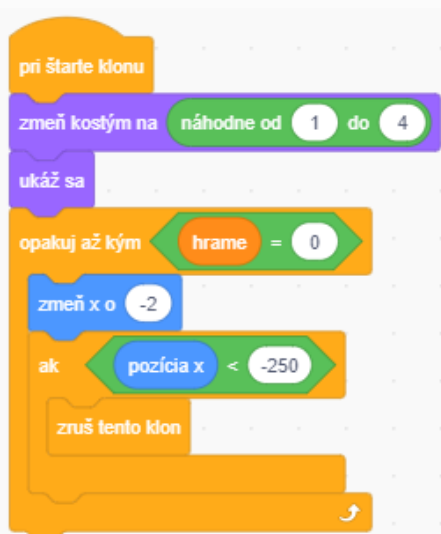
Stena



Počiatkové nastavenia budú skok na x: 300, y: 20, nastavíme premennú Vyroba balon na 0 a skryjeme sa. Ďalej používame cyklus typu opakuj až kým premenná hrame nebude rovná 0. V jeho vnútri klonujeme seba, voláme blok Vytvor balon.

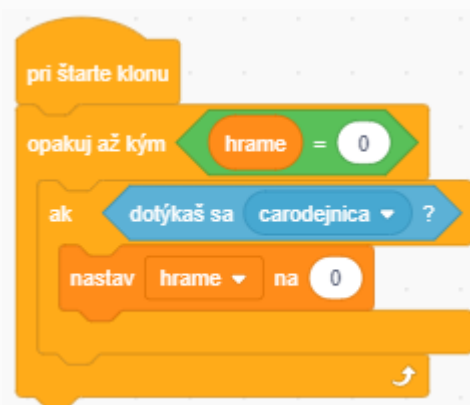
V bloku Vytvor balon zvýšime o 1, vložíme podmienku ak je premenná Vyroba balon rovná 3, naklonujeme postavu balon a nastavíme premennú Vyroba balon na 0. Tento blok nám zabezpečuje, že vygenerujeme balón každú 3. stenu.

Po bloku Vytvor balon v cykle čakáme 3 sekundy.



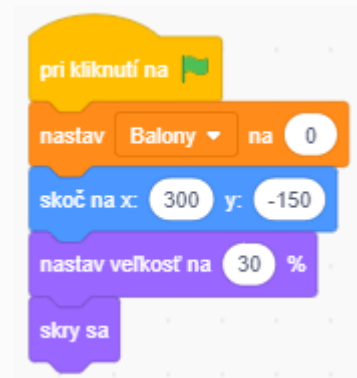
Vytvárame 2 bloky pri štarte klonu. V jednom používame opakuj až kým premenná hrame nebude rovná 0 a v ňom kontrolujeme podmienkou či sa dotýkame postavy carodejníka. Ak áno, nastavíme premennú hrame na 0.

V druhom meníme kostým náhodne od 1 do 4. Pretože máme 4 rôzne druhy stien. Ukážeme sa, vložíme cyklus typu opakuj až kým sa premenná hrame nerovná 0. V jej vnútri meníme x o -2 a používame podmienku, ak x súradnica steny je menšia ako -250, tak zrušíme tento klon.



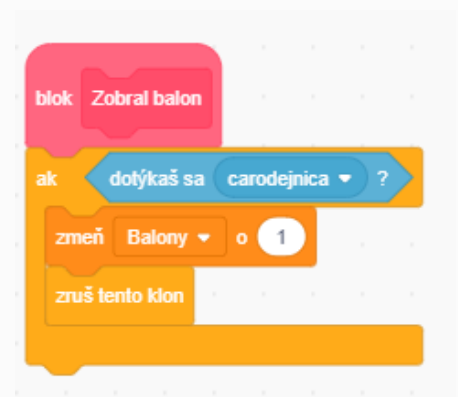
Balón

Pri kliknutí na zelenú vlajočku nastavíme premennú Balony na 0, skočíme na x: 300, y: -150, nastavíme veľkosť na 30% a skryje sa.



Vytvoríme blok Zaciatok, v ňom cyklus typu opakuj až kým sa nedotýkam farby steny a v ňom opakujeme príkaz zmeň y o 10. Po ňom zmeníme y o 50 a ukážeme sa.

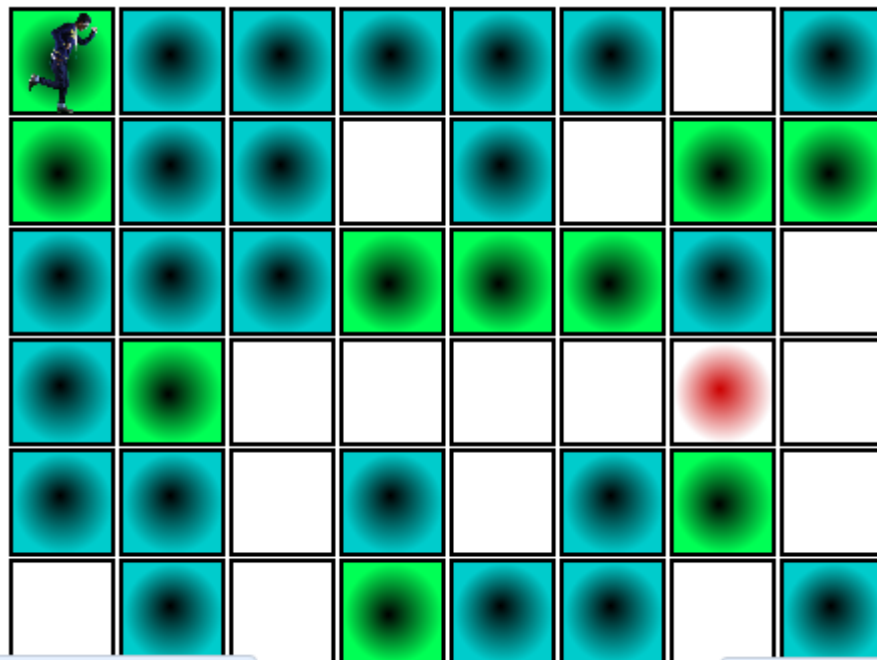
Vytvoríme blok Zobral balon na detekciu dotyku. Tu dáme podmienku s argumentom ak sa dotýkaš čarodejnice, zmeň premennú Balony o 1 a zruš tento klon.



Vytvoríme blok Mimo obrazovku na detekciu kedy už balón nemôžeme zobrať lebo je mimo obrazovky. Tu dávame podmienku ak je x súradnica menšia ako -220 zrušíme tento klon.

Políčka

Naša úloha je pohybom panáčka vyčistiť celé pole, tzn. všetky políčka budú biele a dôjsť do cieľa v čo najmenšom počte ťahov. Po vstupe postavy na políčko nám políčko zmení farbu. Máme 3 farby a ich postupnosť je biela -> modrá -> zelená.



Cistých ťahov

0

Ťahov

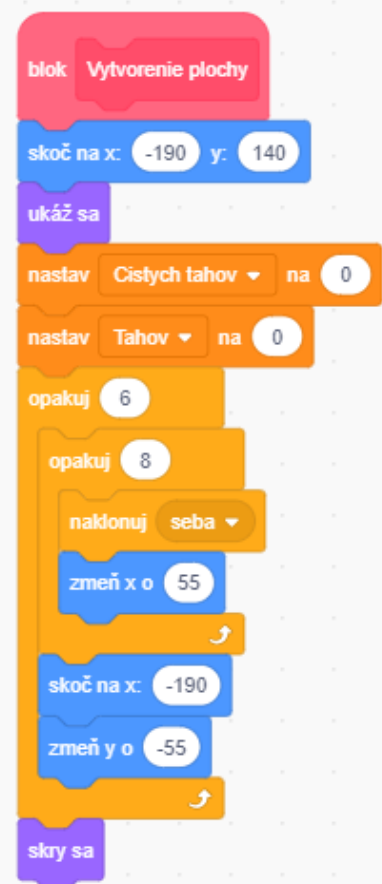
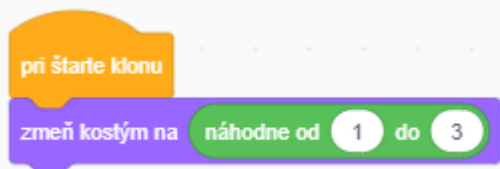
0

Vytvorenie hracieho pola

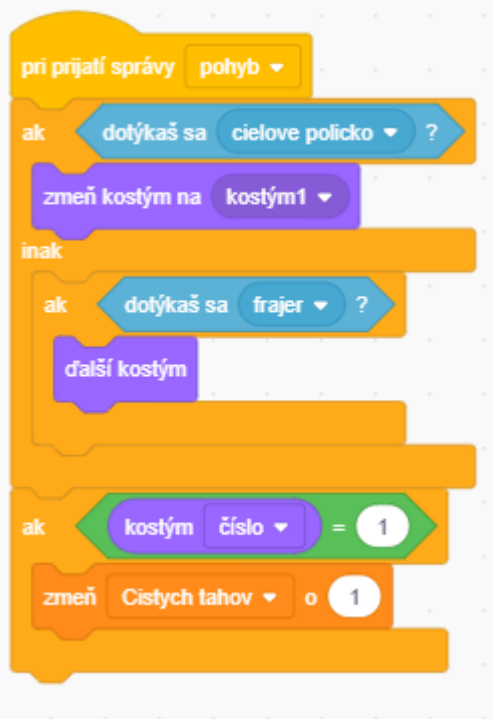
Máme postavu „policko“, túto postavu budeme klonovať niekoľko krát, aby sme vytvorili hracie pole. Klonovanie sme sa naučili na predošlých hodinách, ale ako vytvoríme takto usporiadané hracie pole? Využijeme vnorené cykly a v nich budeme klonovať políčka.

Prvý cyklus opakujeme 6 krát, to bude počet riadkov a v jeho vnútri cyklus opakujeme 8 krát, to bude počet stĺpcov. Keď je cyklus takto naskladaný v sebe, stále ideme pekne po poradí zhora dole, ale musíme si uvedomiť, že vnútorný cyklus sa vykoná všetkých 8 krát a až potom ideme ďalej. Každé políčko posúvame o dané hodnoty x a y, aby sa nám vytvorilo hracie pole. Na konci je príkaz skry sa, aby sme ukryli pôvodnú postavu „policko“, pomocou ktorej sme naklonovali celé hracie pole.

Ďalej chceme, aby pole bolo už od začiatku viac farebné, preto pri vytvorení klonu nastavíme náhodný kostým.



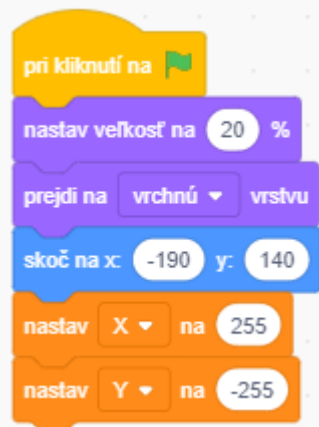
V postave „policko“ bude taktiež logika, ktorá vyhodnotí, či je políčko vyčistené a prirába ho do premennej „Cistych tahov“.



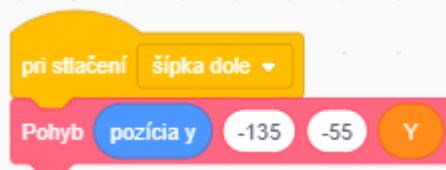
Vždy keď sa postavou „frajer“ pohneme, odosielame taktiež správu pohyb. Tu ošetrujeme, aby cieľové políčko bolo vždy čisté. Ak na cieľovom políčku nie sme, tak nastavujeme políčku ďalší kostým, čiže ďalšiu farbu. Ak je farba políčka biela (takže má kostým číslo 1), pripočítame si čistý ťah. Čistý ťah sa pripočítava za každé biele políčko, takže po prvom pohybe sa nám nastaví na hodnotu všetkých čistých políčok na ploche.

Pohyb postavou po ploche

Postave musíme samozrejme nastaviť nejaké počiatočné nastavenia. Pôjde o veľkosť, prejdienie na vrchnú vrstvu a počiatočnú pozíciu. Vidíme tu tiež nastavenie pomocných premenných X a Y. Tie budú vysvetlené v ďalšom kroku.

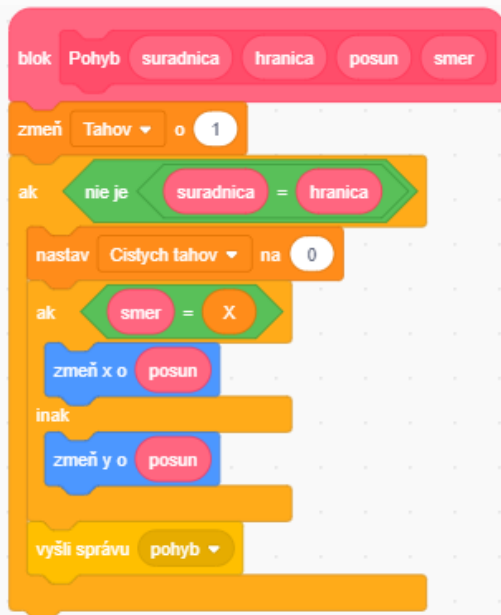


Pre pohyb do strán máme vytvorený blok „Pohyb“. Doteraz sme bloky použili iba ako nejaké spojenie nemienciach sa príkazov, ktoré používame viac krát. Bloky, ale majú aj zlepšenú funkcionlitu, že vieme do nich zapísať veci, ktoré chceme použiť v ich vnútri.



V programovacích jazykoch by sme blok Pohyb nazývali funkciou a pozícia y, -135, -55 a Y by boli jej **argumenty**. Argumenty nám sú nápomocné tým, že ich vieme využiť vo vnútri bloku v konkrétnych príkazoch a vždy môžu byť odlišné. Funkcie sú nutnosťou pri programovaní v akomkoľvek jazyku.

V definícii bloku Pohyb máme tieto argumenty pomenované ako **suradnica**, **hranica**, **posun** a **smer**. Tieto argumenty vieme využívať v príkazoch. Argument **suradnica** je aktuálna súradnica postavy, tú porovnáваме s argumentom hranica. Argument **hranica** je zadaná hodnota okraja hracieho

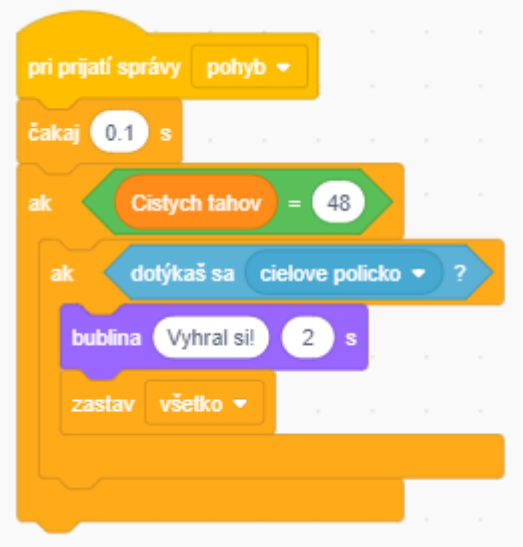


sme mohli hru ukončiť.

poľa podľa toho, do ktorej strany ideme. Napríklad keď ideme hore, tak je hranica vrchný riadok, aby sme nemohli ísť vyššie.

Argument **posun** je o koľko sa posúvame na x alebo y osi. Do argumentu **smer** vkladáme ešte na začiatku spomínané premenné X alebo Y. Vďaka nim vieme či sa máme hýbať hore/dole alebo vpravo/vľavo. Následne posielame správu pohyb.

V postave „frajer“ pri prijatí správy pohyb kontrolujeme či sme už vyčistili celé hracie pole a sme na cieľovej pozícii, aby



Umiestnenie cieľového políčka

Postava „cieľove políčko“ má jednoduché príkazy. Skočí na počiatočnú pozíciu, dáme ho na vrchnú vrstvu, ukážeme a následne ho presunieme na náhodnú pozíciu pomocou jednoduchého matematického vzťahu.



Takýmto vzťahom zabránime tomu, aby bolo cieľové políčko na poslednom riadku a stĺpci.