

Scratch: Chytání ovoce

Dnes se pokusíme ovládat človička pomocí šipek a chytat jím padající ovoce. Hru si můžete vyzkoušet zde: <https://scratch.mit.edu/projects/441005609/>

Postup vývoje

Spustíme si nový editor: <https://scratch.mit.edu/projects/editor/>

Tentokrát si musíme vytvořit 2 sprity, človička a ovoce.

Človiček

1. musí být umístěn na spodní části obrazovky
2. musí se otáčet vlevo-vpravo
3. pokud bude stisknutá šipka vpravo, nastavit na směr 90° a popojít o 5 kroků
4. pokud bude stisknutá šipka vlevo, nastavit směr na -90° a popojít o 5 kroků

Ovoce

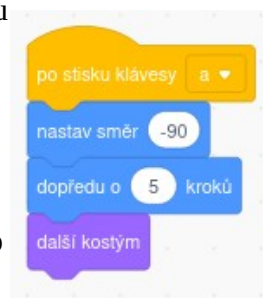
1. musí být umístěn v horní části a být skrytý
2. do nekonečna klonuj ovoce v náhodných intervalech
3. když startuje můj klon, tak skočím na náhodnou pozici úplně nahoře, ukážu se a pomaličku padám, dokud:
 1. nenarazím na zem – odečíst bod
 2. nedotýkám se človička – přičíst bod
4. snižuj interval klonování ovoce podle počtu bodů

Jak na to

Pro človička musím naprogramovat „po stisku na ZELENÁ_VLAJKA“ původní podmínky, to jest pozice, směr, styl otáčení a vynulování skóre.

Nyní musíme rozpohybovat človička. Samotný pohyb je jednoduchou sekvencí „nastav směr XXX“, „dopředu o XXX kroků“ a „další kostým“. Vpravo se posouváme pomocí směru 90°, vlevo o -90°. Počet kroků udává rychlost panáčka, hodnota kolem 5 kroků na stisk by měla stačit. Otázka je, jak zjistit stisknutí klávesy.

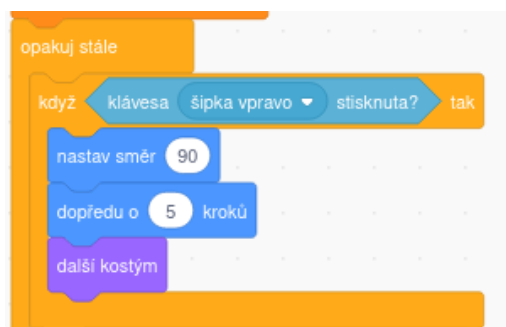
Nejprve můžeme vyzkoušet jednodušší způsob za pomoci bločku „po stisku klávesy XXX“. Funguje podobně, jako „po kliknutí na ZELENÁ_VLAJKA“, akorát reaguje na stisk klávesy a po každém stisku vykoná zadanou sekvenci. Na obrázku vidíme příklad kde po stisku tlačítka „a“ popojde človiček doleva.



kód jednoduchého pohybu vlevo

Po vyzkoušení zjistíte slabinu tohoto řešení.

Človíček popojde s každým stiskem vždy jen o 5 kroků. Pak musí uživatel klávesu pustit a opět ji stisknout. Alternativní řešení je za použití nekonečné smyčky a použití bločku „když XXX tak“ kde za podmínku použijeme „klávesa XXX je stisknuta?“ (z kategorie „Vnímání“). Výsledkem je nepřetržitý pohyb při držení klávesy.

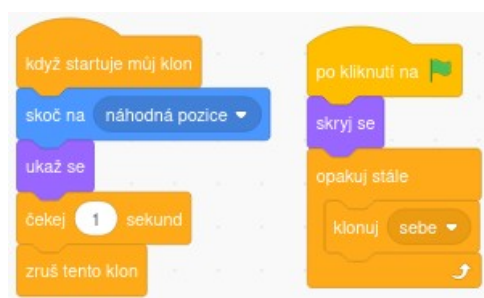


kód pohybu vpravo s opakováním

Dalším naším úkolem je rozpoohybovat ovoce.

Vzhledem k tomu, že nechceme vytvářet sprity pro každé ovoce zvlášť, můžeme využít funkci „klonuj XXX“, která nám umožňuje vytvořit nekonečně kopií našeho spritu. Ovládat takovéto kopie můžeme pomocí události „když startuje můj klon“.

Pro vyzkoušení se překlikněte na ovoce, umístěte jej do horní části a skryjte jej. Následně přidejte kód pro „když startuje můj klon“, kde skočíte na náhodné místo, zviditelníte se, počkáte chvíli a zrušíte klon pomocí „zruš tento klon“.



kód test generování ovoce

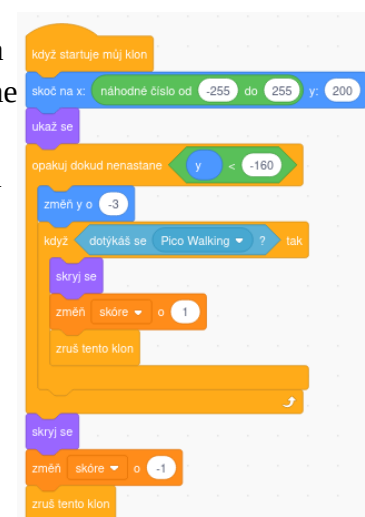
Tento kód sám o sobě neudělá nic, ale když si přidáme ještě kód „po kliknutí na ZELENÁ_VLAJKA“, kde do nekonečna budeme klonovat sebe, měli bychom vidět zobrazující se a mizící ovoce.

My ale nepotřebujeme ovoce jen zobrazit, potřebujeme, aby padalo. Jak na to? Jednoduše. Stačí ovoce zobrazit kdekoli v horním okraji obrazovky a s každým krokem měnit polohu „y“ o nějakou hodnotu (doporučuji „-3“). Nyní to těžké, kdy přestat. K tomu nám pomůže bloček „opakuji dokud nenastane XXX“ kde za podmínku dosadíme polohu země. K podmínkám zpravidla využíváme kategorii „Operátory“ a využijeme k tomu „XXX < YYY“, které je splněno, když hodnota nalevo je menší, než hodnota napravo. V mém případě je to „y < -160“. Uvnitř tohoto konečného cyklu budeme padat, to jest měnit hodnotu o „-3“. A co se stane, když podmínka nastane? To znamená, že jsme spadli na zem, snížíme skóre o 1 a zrušíme tento klon (už spadl, není nám k ničemu).



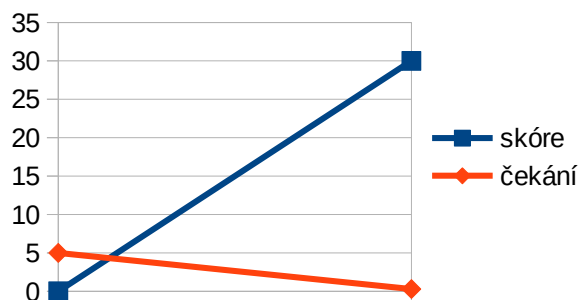
Figure 1: kód generování ovoce s padáním

Ovoce nám již padá, ovšem my bychom rádi i ovoce sbírali. Proto kromě „změň y o -3“ využijeme další podmínky „když XXX“, kde za XXX dosadíme podmínku „dotýkáš se XXX“ (človíček). Toto budeme kontrolovat při každém posunu a pokud takováto situace nastane, zvýšíme skóre a opět zrušíme klon (ovoce jsme sebrali, zmizelo, klon již k ničemu nepotřebujeme).



kód generování ovoce s dotykem

K dokonalosti nám chybí už jen zvyšovat rychlost klonování s přibývajícím body. Budeme tedy potřebovat hodnotu „skóre“ a využijeme jej pro čekání v cyklu, kde klonujeme ovoce. Možností je opět mnoho, v našem případě využijeme jednoduché lineární mapování ve tvaru „ $ax + y = z$ “. Při skóre 0 chceme čekat 5s, když si dosadíme „ $a*0 + y = 5$ “, získáme „ $y = 5$ “. Pro 30s chceme čekat



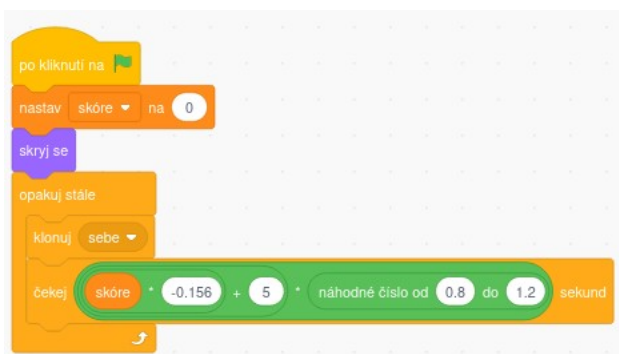
lineární mapování

0,3s, opět dosadíme „ $a*30 + 5 = 0,3$ “ a získáme „ $a = -0,156666$ “. Nyní jen potřebujeme tuto rovnici zapsat pomocí „Operátorů“ do našeho programu a sledovat, jak se zrychluje klonování ovoce s přibývajícím počtem bodů.



kód klonování s lineárním mapováním

Jako třesinka na dortu je pak možnost vynásobit tuto naši čekací dobu náhodným koeficientem, který vnese do hry určitou dynamiku. Násobení je krásné v tom, že násobení číslem 1 nezmění výsledek, násobení číslem menším jej zmenší (0.5 na polovic), násobení větším číslem jej zvětší (2 na dvojnásobek). Pro lehkou dynamiku je vhodné zvolit rozsah 0.8 – 1.2. Čísla v tomto rozsahu zkrátí či prodlouží rozdíl o znatelný, ale jen drobný okamžik (vizualizace viz obrázky se smajlíkem), což zpravidla působí pozitivně na atmosféru hry při zachování hrátelnosti.



kód generování ovoce s náhodným koeficientem



0,8 1 1,2

Závěr

Příště použijeme události, abychom do hry přidali zvuky a také další podmínky, abychom mohli určit konec a začátek hry.