

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

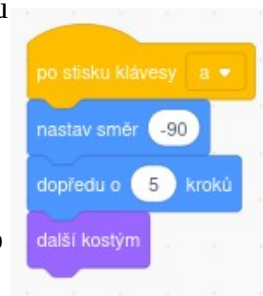
1. musí být umístěn v horní části a být skrytý
2. do nekonečna klonuj ovoce v náhodných intervalech
3. když startuje můj klon, tak skočím na náhodnou pozici úplně nahoře, ukážu se a pomaličku padám, dokud:
 1. nenarazím na zem – odečíst bod
 2. nedotýkám se človička – přičíst bod
4. snižuj interval klonování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

Jak na to

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

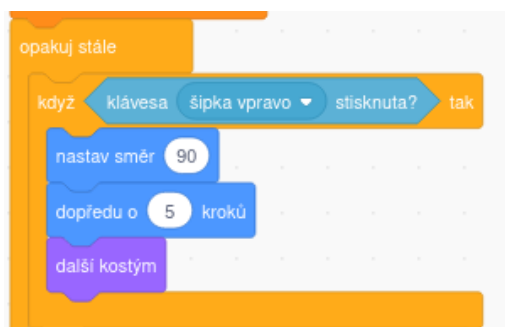
Nyní musíme rozpořehovat človička. Samotný pohyb je jednoduchou sekvencí „nastav směr XXX“, „dopředu o XXX kroků“ a „další kostým“. Vpravo se posouváme pomocí směru 90°, vlevo o -90°. Počet kroků udává rychlost panáčka, hodnota kolem 5 kroků na stisk by měla stačit. Otázka je, jak zjistit stisknutí klávesy.

Nejprve můžeme vyzkoušet jednodušší způsob za pomoci bločku „po stisku klávesy XXX“. Funguje podobně, jako „po kliknutí na ZELEŇÁ_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 rozpochybovat 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

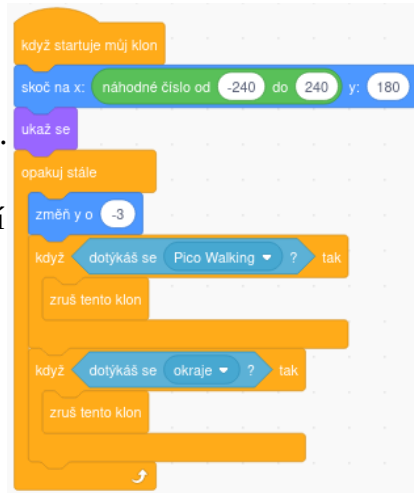
Tento kód sám o sobě neudělá nic, ale když si přidáme ještě kód „po kliknutí na ZELENÁ_VLAJKA“, kde do nekonečna budeme klonovat sebe, měli bychom vidět zobrazující se a mizící ovoce.

My ale nepotřebujeme ovoce jen zobrazit, potřebujeme, aby padalo. Jak na to? Jednoduše. Stačí ovoce zobrazit kdekoli na horním okraji obrazovky a s každým krokem měnit polohu „y“ o nějakou hodnotu (doporučuji „-3“). Výsledek je ukázán na obrázku „kód padání ovoce 1“.



kód padání ovoce 1

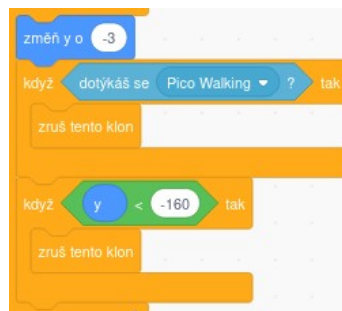
Všimněte si, že jsme odstranili bloček „zruš tento klon“ a také nám všechno ovoce zůstává ležet na zemi. Co s tím? Použijeme podmínky „když XXX tak“ abychom definovali, co se stane, když ovoce spadne na zem, případně když se dotkne našeho panáčka. Tělo tohoto bločku se vykoná pouze tehdy, když v tu chvíli, kdy je bloček „když XXX tak“ vykonáván je podmínka pravdivá. Z toho plyne, že jej musíme pravidelně kontrolovat. Proto jej můžeme vložit do bločku „opakuji stále“ ihned pod naši implementaci gravitace (změň y o -3). Pro vyzkoušení můžeme využít zjednodušených bločků „Vnímání“, kde najdeme „dotýkáš se XXX“ a vybereme „Pico Walking“ (náš človíček), respektive „okraje“ (když ovoce spadne). Pokud je podmínka splněna, chceme aby ovoce zmizelo. K tomu nám poslouží bloček „zruš tento klon“. Výsledek je ukázán na obrázku „kód padání ovoce 2“.



kód padání ovoce 2

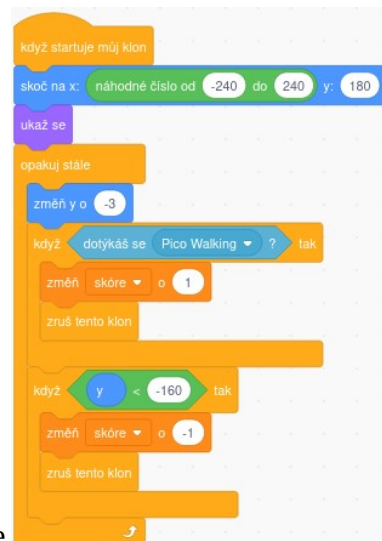
Nyní by již mělo ovoce zmizet jak při dotyku s okrajem, tak při dotyku s panáčkem. Problém je, že ovoce se může dotýkat i ostatních okrajů, proto bude vhodnější postavit druhou podmínku na

poloze „y“ tohoto klonu. Pokud bude hodnota „y < 160“, prohlásíme ho za spadlé a zrušíme jej. Porovnání „XXX < YYY“ najdeme v „Operátorech“ a hodnotu „y“ současného spritu v „Pohybu“. Upravená podmínka je zobrazena na obrázku „kód padání ovoce 3“.



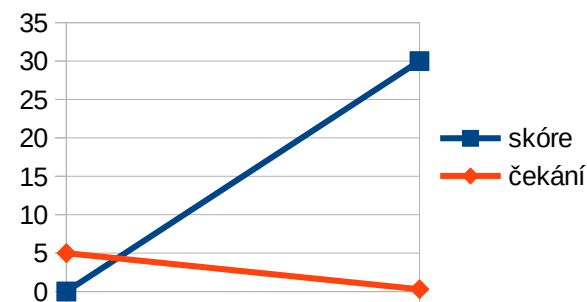
kód padání ovoce 3

Aby hra mohla mít nějaký začátek a konec, budeme potřebovat počítat skóre. Pro tyto účely si v „Proměnné“ vytvoříme novou proměnnou „skóre“ a využijeme bločku „změň XXX o YYY“ pro přidání „1“ když panáček chytne ovoce (první podmínka) a „-1“ pokud spadne na zem (druhá podmínka). Celý kód padání ovoce je na obrázku „kód padání ovoce 4“.



kód padání ovoce 4

K dokonalosti nám chybí už jen zvyšovat rychlost klonování s přibývajícimi 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 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. Mapování je znázorněno na obrázku „lineární mapování“



lineární mapování



kód klonování s lineárním mapováním

Jako třešinka 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

Závěr

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

