

6.2. Hra ovládaná dotykovými gestami

Počítačové hry patria k najpopulárnejším a aj komerčne najúspešnejším aplikáciám. Platí to aj pre mobilné platformy. V online aplikačných obchodoch nájdeme hry rôzneho žánru a kvality – od jednoduchších hier od individuálnych (aj amatérskych) vývojárov po profesionálne produkty veľkých vývojárskych tímov. Tablety a smartfóny sú v porovnaní so stolnými počítačmi špecifické vo viacerých ohľadoch. Ľahko sa prenášajú, podporujú dotykové aj hlasové ovládanie, sú vybavené rôznymi senzormi (napr. kamera, gyroskop, akcelerometer, lokalizačný senzor) a poskytujú prístup k internetu a jeho službám v princípe kdekoľvek a kedykoľvek. Tieto možnosti prirodzene vedú k vzniku hier s originálnym námetom aj spôsobom ovládania.

Otázky na zamyslenie

Aké žánre počítačových hier poznáte?

Aké hry máte nainštalované vo svojom tablete alebo v smartfóne?

Malo by zmysel hrať ich aj na stolnom počítači?

Ktorú mobilnú hru považujete za najoriginálnejšiu, najzaujímavejšiu?

Kľúčové slová

hra, dotykové gestá, komponent Clock, grafické objekty, zvukové efekty, viac úrovní, viac obrazoviek

Čo sa naučíme a čo si precvičíme

- navrhne a naprogramujeme vlastnú hru s viacerými úrovňami ovládanú dotykovými gestami,
- použijeme statické aj pohybujúce sa grafické objekty a zvukové efekty,
- precvičíme si prácu s viacerými obrazovkami,
- komponent *Clock* z kategórie *Sensors* zužitkujeme novým spôsobom: na riadenie zmeny stavu sprajtov, resp. ako prostriedok na plánovanie udalostí v hre,
- vytvorenú hru zverejníme v *Galérii* prostredia MIT AI2.

Akú zaujímavú aplikáciu môžeme vytvoriť?

Naším cieľom bude **vyvinúť** (t. j. vymyslieť a naprogramovať) **vlastnú arkádovú hru typu „plošinovka“**. Arkádové hry sú typické jednoduchým, ale zábavným a pútavým konceptom. Obsahujú viacero úrovní (levelov) so stupňujúcou sa náročnosťou. O úspešnosti hráča rozhoduje najmä dobrý postreh a zručnosť v ovládaní vstupného zariadenia, ale často tiež schopnosť koncentrovať sa, logicky myslieť, predvídať a rýchlo sa rozhodovať. V plošinových hrách prekonáva hlavný hrdina rôzne nástrahy virtuálneho sveta - skáče cez prekážky, vyhýba sa nepriateľom alebo s nimi bojuje. Popritom zbiera rôzne predmety, ktoré mu môžu pomôcť alebo za ktoré získava body.

Ako budeme postupovať pri tvorbe aplikácie?

Pri tvorbe vlastnej plošinovej hry bude potrebné:

- zvoliť vhodný námet,
- premyslieť si pravidlá hry a nadväznosť levelov,
- navrhnuť grafické používateľské rozhranie hry, pripraviť obrázky a zvuky (príp. iné dáta potrebné pre jednotlivé levely),
- vyriešiť problém generovania grafických objektov (napr. pozadí, prekážok, nepriateľov) v hernom svete,
- vyriešiť problém ovládania hlavného hrdinu (beh, výskok, strieľanie a pod.),
- vyriešiť problém kolízie hráča s inými grafickými objektami.

Po otestovaní hry môžeme výsledok svojej práce zverejniť v *Galérii* prostredia MIT AI2, aby sme získali spätnú väzbu aj od iných používateľov.

V tejto kapitole budeme **riešenia vybraných problémov ilustrovať na príklade jednoduchej vzorovej hry**, v ktorej je hlavným hrdinom zajac bežiaci po lúke. Zajaca budeme ovládať dotykom prsta. Nad hlavou mu lietajú mrkvy, ktoré má zbierať, pod nohami sa mu gúľajú kapusty, ktoré má preskakovať. Keď zajac zakopne o kapustu, stráca život.

Na postup do nasledujúceho levelu musí hráč chytiť predpísaný počet mrkvičiek. Základné parametre hry (napr. počet životov) bude možné nastaviť na osobitnej obrazovke. Na pozadí sa má prehrávať rytmická hudba, pri zrážkach sprajtov zase vhodné zvukové efekty. Hra by sa mala dať v priebehu hrania aj pozastaviť.

Niektoré komponenty, ktoré použijeme, ste už spoznali v úvodných etudách alebo pri práci na iných projektoch:

- vizuálne komponenty a správcovia rozvrhnutia,
- komponenty *Canvas* a *ImageSprite*,
- komponenty *Sound* a *Player*,
- komponent *Clock*,
- lokálna databáza *TinyDB*, komponent *Screen* (viac obrazoviek v aplikácii).

Úloha 1

V projekte **pmz_6_2_zajacovka_ver0.aia** nájdete nultú verziu vzorovej hry s grafickými a zvukovými súbormi a základnými komponentami tvoriacimi používateľské rozhranie aplikácie:



Obr. 6.2.1 Používateľské rozhranie základnej verzie hry Zajacovka

Všimnite si, že:

- obrazovka má nastavenú orientáciu na šírku (*Landscape*)
- na plátne sú umiestnené 3 komponenty typu *ImageSprite* (sprajty) s obrázkami zajaca, mrkvy a kapusty,
- horný pás nad plátnom obsahuje 2 statické obrázky (komponenty typu *Image*), nápisy s aktuálnym stavom počítačiel a tlačidlá pre spustenie novej hry a ukončenie práve bežiackej hry (vhodné umiestnenie komponentov sme dosiahli vložením jedného komponentu *HorizontalArrangement* do druhého),
- pre zajaca je k dispozícii 8 obrázkov s názvami *zajac1.png*, *zajac2.png*, ..., *zajac8.png* (ide o rôzne fázy animácie behu),
- v časti *Media* sú pripravené už aj zvukové súbory (budeme ich potrebovať neskôr).

Zajac sa zatiaľ nepohybuje, nebeží. Pomocou komponentu *Clock* zabezpečte, aby bol zajac animovaný (t. j. aby vyzeral, že beží).

Pomôcky

Komponent *Clock* (časovač) sme už na generovanie udalostí v pravidelných intervaloch používali. Fázy animačného cyklu budeme striedať v reakcii na udalosť *Clock.Timer* (teda v okamihu, keď časovač „tikne“). Na uloženie poradového čísla aktuálnej fázy využijeme globálnu premennú.

V tomto prípade je vhodné nastaviť vlastnosť *Clock.TimerEnabled* tak, aby zajac začal bežať až po odštartovaní novej hry. Hodnota vlastnosti *Clock.TimerInterval* by mala byť dostatočne malá na to, aby animácia pohybujúcich sa nôh pôsobila plynulo.

Po spustení novej hry už zajac beží na mieste. Zatiaľ však nevie vyskočiť, naučíme ho to:

Úloha 2

Pomocou ďalšieho časovača zabezpečte, aby zajac vyskočil nahor *primerane rýchlo* a *dostatočne vysoko*. Výskok zajaca ovládajte dotykom prsta v ľubovoľnom miesta plátna (t. j. lúky, po ktorej zajac beží).

Vysvetlíme si

Výskokom zajaca rozumieme plynulé posúvanie sa zajaca z jeho základnej pozície smerom nahor a následné plynulé zostúpenie smerom nadol.

Sprajty sú schopné (ak sú aktívne) pohybovať sa určeným smerom aj samé (každých *ImageSprite.Interval* milisekúnd sa posunúť v smere *ImageSprite.Heading* o *ImageSprite.Speed* pixelov). Tento prístup sa nám ale teraz nehodí. Pohyb zajaca budeme riadiť vlastným nezávislým časovačom (do projektu vložíme ďalší komponent typu *Clock*).

Po zapnutí časovača sa zajac presunie niekoľkokrát nahor (počet opakovaní treba zladiť s počtom pixelov posunu, dosiahnuť optimálnu rýchlosť a výšku). Následne zmeníme smer pohybu na opačný a postupne zajaca vrátíme naspäť do východiskovej pozície. Po návrate zajaca nadol

časovač pre riadenie skoku vypneme a zapneme znovu časovač riadiaci animáciu behu. Počas skoku môže mať zajac nastavenú fázu 4 naznačujúcu skok.

Úloha 3

Podľa zadania by zajacovi mali ponad hlavu lietať mrkvy a pod nohami sa mu gúľať kapusty. Naprogramujte automatický pohyb týchto sprajtov.

Pomôcky

Experimentuje s rôznymi hodnotami vlastností `ImageSprite.Interval`, `ImageSprite.Speed` a umiestnením sprajtov vzhľadom na pozíciu zajaca. Dokáže zajac kapustu preskočiť? Má možnosť dočiahnuť na mrkvu? Keď zajac kapustu preskočí, bude pokračovať v pohybe ďalej až po pravý okraj obrazovky. Podobne, keď zajac mrkvu nechytí, bude mrkva letieť ďalej k pravému okraju. Oba tieto sprajty sa po dosiahnutí pravého okraja obrazovky majú objaviť vľavo.

Zajac už vie skákať, ale pri zrážke s mrkvou ani s kapustou sa nič neudeje. Pohyb kapusty aj mrkvy je pravidelný a preto ľahko predvídateľný.

Úloha 4

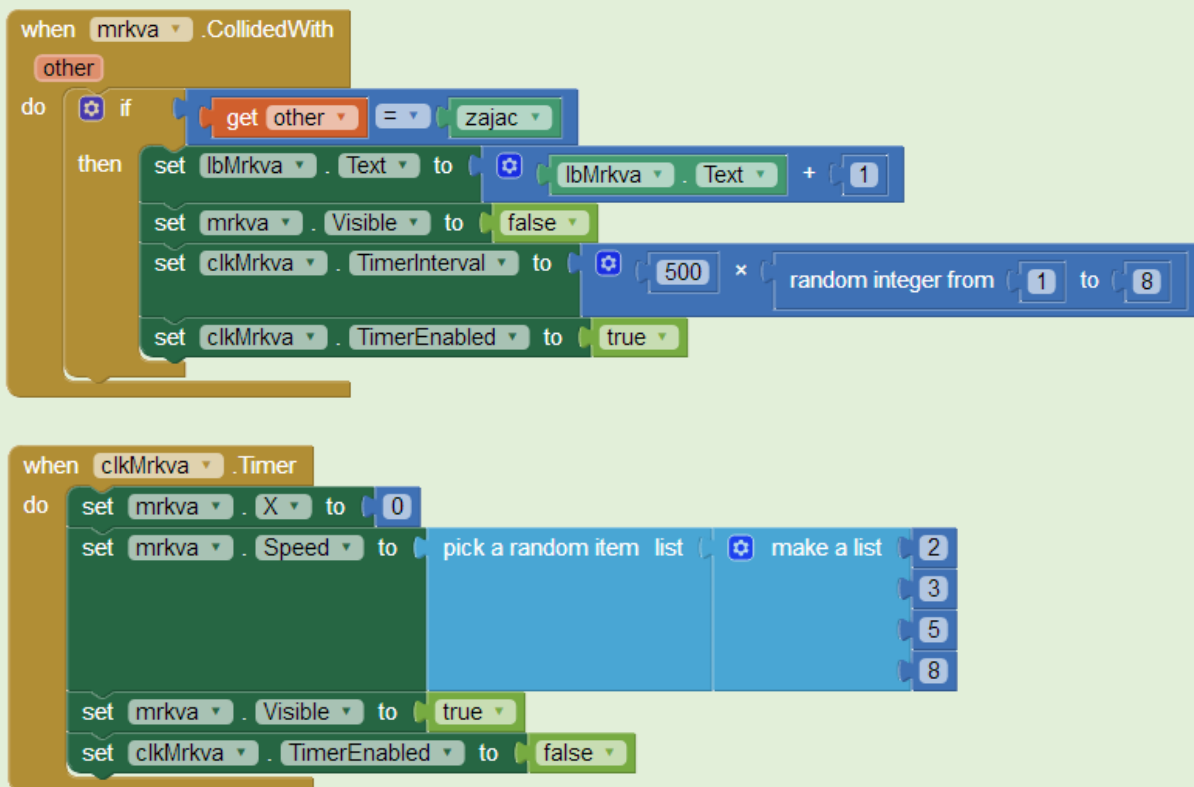
Upravte riešenie tak, aby sa ďalšia mrkva po zrážke so zajacom objavila na ľavom okraji obrazovky *o náhodne zvolený počet sekúnd* a letela vpravo *náhodne zvolenou rýchlosťou*. Kapusta by sa v prípade zrážky so zajacom mohla začať gúľať ihneď (t. j. v jej prípade nebudeme rozlišovať medzi nárazom na zajaca a nárazom na pravý okraj obrazovky).

Vysvetlíme si

Komponent `Clock` môžeme používať ako časovač generujúci udalosti v pravidelných intervaloch. Po zapnutí časovača časovač „tiká“ opakovane, až kým ho nevypneme. Na každé tiknutie môžeme zareagovať v udalostnom bloku `when Clock.Timer`.

Niekedy sa časovať hodí na naplánovanie vzniku jednorazových udalostí dôležitých pre riadenie behu aplikácie, zmenu stavu a pod. Časovaču môžeme nastaviť interval, po uplynutí ktorého sa vykonajú príkazy uvedené v jeho udalostnom bloku. K ďalšiemu tiknutiu časovača už nedôjde, keďže časovač na záver udalostného bloku vypneme.

Jednorazovou udalosťou (teda takou, ktorá nenastáva opakovane v pravidelných intervaloch) je v našej hre pokračovanie letu mrkvy z ľavej strany po predchádzajúcej zrážke so zajacom:



Naplánovať udalosť „objavenie sa mrkvy zľava“ znamená nastaviť pri zrážke príslušnému časovaču vlastnosti tak, aby tikol o požadovaný počet milisekúnd.

V ukážke zdrojového kódu vyššie vidíme, že sme časovaču `clkMrkva` nastavili vlastnosť `TimeInterval` na hodnotu, ktorá je výsledkom vyhodnotenia výrazu s náhodným číslom (možno pôjde o pol sekundy, možno až o 4 sekundy). Časovač sme vzápätí zapli nastavením jeho vlastnosti `Enabled` na `true`.

Druhý udalostný blok obsahuje príkazy na presun mrkvy z miesta zrážky na ľavú stranu obrazovky a nastavenie jej rýchlosti na náhodnú výberom niektorej hodnoty z pripraveného zoznamu možností. Posledným príkazom v tomto udalostnom bloku je vypnutie časovača `clkMrkva`.

Úloha 5

V komponentoch typu *Label* v hornom páse nad plátnom zobrazujte aktuálny stav počítadiel (počet chytených mrkiev, počet životov). Zabezpečte ukončenie hry a umožnite jej spustenie odznovu.

Pomôcky

Počítadlo súvisiace s guľajúcimi sa kapustami slúži ako počítadlo životov. Po každej zrážke zajaca s kapustou sa jeho hodnota zníži o 1 (hráč stratí život). Po strate všetkých životov (napr. po desiatom zakopnutí o kapustu) sa hra skončí. Príkazy, ktoré je potrebné vykonať pri skončení hry alebo pri spúšťaní novej hry je vhodné umiestniť do samostatných procedúr.

Úloha 6

Po nazbieraní istého počtu mrkiev (na testovacie účely postačí napr. 5) bude možné zajaca kedykoľvek v priebehu hry (teda aj počas výskoku) ťahaním presúvať vpred alebo vzad po lúke. Vďaka tomu bude hráč schopný lepšie plánovať a realizovať výskoky a bude úspešnejší. Získanie schopnosti presúvať sa aj v horizontálnom smere budeme v našej vzorovej hre považovať za nový (druhý) level.

Otázky na zamyslenie

Zamyslite sa, čo znamená posúvať zajaca ťahaním vpred alebo vzad v horizontálnom smere. Ktorá súradnica sprajtu sa pri ťahaní bude meniť? Ktorý udalostný blok budeme potrebovať?

V titulkovom pruhu môžeme zobrazovať, v ktorom leveli sa hráč práve nachádza.

Úloha 7

Doprogramujte v hre prehrávanie rytmickej hudby na pozadí a pridajte zvukové efekty pri zrážkach mrkvy a kapusty so zajacom.

Dovoľte tiež nastavovať niektoré parametre hry používateľovi (napr. počet životov a počet mrkiev, ktoré sú potrebné na prechod do ďalšieho levelu).

Pomôcky

Na prehrávanie hudby na pozadí je vhodné použiť komponent `Player`. V režime *Design* nastavíme vlastnosť `Player.Source` na názov súboru s rytmickou hudbou (napr. `podmaz.mp3`). V zdrojovom kóde doplníme na vhodné miesta príkazy na prehrávanie a zastavenie prehrávania:

`call Player1 .Start` `call Player1 .Stop`

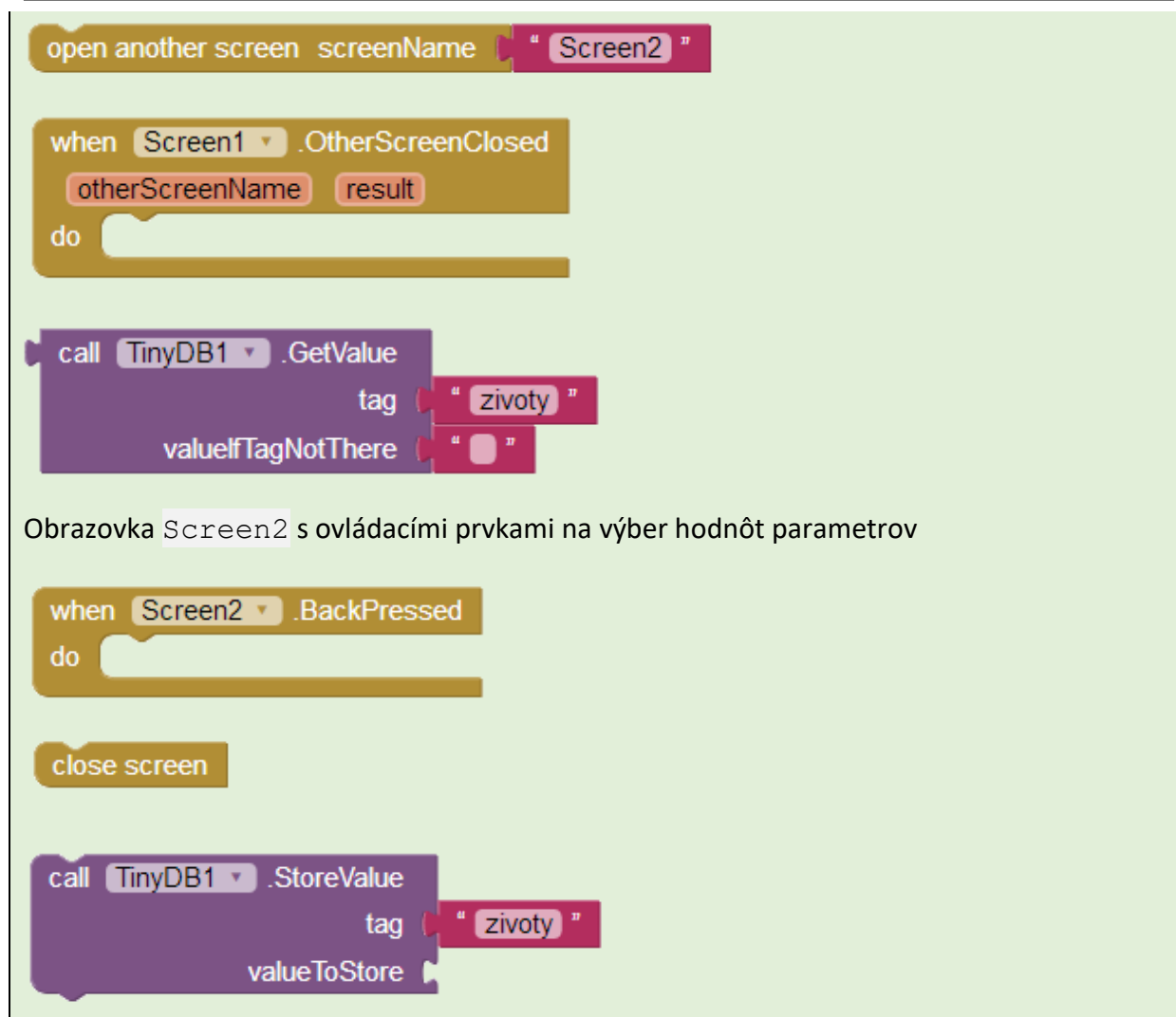
Na prehrávanie krátkych zvukových efektov použite komponent `Sound`. Keďže chceme akusticky odlíšiť zakopnutie o kapustu od chytenia/zjedenia mrkvy, pri každej kolízii najprv nastavíme správny názov zdrojového súboru so zvukom a potom ho prehráme:

`set Sound1 .Source to "chrum.wav"`
`call Sound1 .Play`

Nastavovanie parametrov hry by sa malo realizovať na samostatnej obrazovke. Pridajte preto do projektu ďalší komponent typu `Screen`, navrhňte vzhľad tejto obrazovky. Používateľovi môžete dať na výber rôzne hodnoty parametrov s využitím komponentu `Spinner`. Na hlavnú obrazovku sa môžete vrátiť stlačením tlačidla potvrdzujúceho zmenu alebo zabudovaným tlačidlom pre návrat späť.

Aktuálne hodnoty parametrov môžeme uchovávať v databáze `TinyDB`, ktoré obe obrazovky zdieľajú. Na implementovanie tejto funkcionality budete potrebovať aj tieto bloky:

Obrazovka `Screen1` (hlavná obrazovka aplikácie)



Úloha 8

Porozmýšľajte nad námetom pre vlastnú hru alebo zrealizujte niektoré z nižšie uvedených námetov na vylepšenie základnej verzie hry.

Ako vylepšiť či rozšíriť našu aplikáciu?

Vzorovú hru by sme mohli vylepšovať ďalej. Uvádzame niekoľko nápadov, ktoré vás môžu inšpirovať aj pri tvorbe vlastnej hry:

- po spustení aplikácie nech sa zobrazí dialógové okno s informáciou o pravidlách a ovládaní hry,
- po každej x-tej mrkve, ktorú chytíme, získame pre zajaca jeden život späť,
- pridáme viac levelov, v každom ďalšom leveli sťažíme hráčovi situáciu (mrkvy začnú lietať rýchlejšie, kapusty a guľajú častejšie, pribudnú aj iné grafické objekty a pod.),
- zajac môže mať aj ďalšie schopnosti (zatiaľ s ním vieme skákať a presúvať ho v horizontálnom smere ťahaním, mohol by sa vedieť chvíľu vznášať vo vzduchu alebo zničiť kapustu vystrelením kalerábu),
- hra môže mať časový limit,
- pozadie za zajacom by sa mohlo v priebehu hry meniť,

- výsledky hráčov sa zapíšu do databázy a budú sa dať zobrazíť na samostatnej obrazovke,
- v rozhraní v hornom páse bude k dispozícii tlačidlo na pozastavenie hry, aby ju hráč mohol prerušiť a dohrať neskôr,
- prehrávanie hudby na pozadí a zvukových efektov sa bude dať vypnúť,
- detegovanie kolízie zajaca a mrkvy bude presnejšie (nech zajac nechytí mrkvu aj vtedy, keď ju už má za ušami).