

## 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.

### Príprava na výučbu

Prerekvizity: udalosti súvisiace s dotykovým ovládaním (etuda 2.1), komponent *ImageSprite/Ball* (etuda 2.3, 2.4), komponent *Clock* (etuda 2.3 a 2.4), komponent *TinyDB* (etuda 2.4), viac obrazoviek (etuda 2.5), komponent *Spinner* (etuda 2.6).

Súbory obrázkov a zvukov, ktoré sú použité vo vzorovej aplikácii, sa nachádzajú v archíve **media.zip**.

Prílohou kapitoly sú tiež 4 verzie riešenia. Projekt **pmz\_6\_2\_zajacovka\_ver0.aia** je východiskovým projektom pre riešenie úloh uvedených v kapitole, je určený pre žiakov.

Ďalšie projekty predstavujú vzorovú aplikáciu v stave po vyriešení Úloh 1 až 6 (**pmz\_6\_2\_zajacovka\_ver1\_R.aia**), Úlohy 7 (**pmz\_6\_2\_zajacovka\_ver2\_R.aia**) a vybraných námetov na vylepšenie a rozšírenie aplikácie (**pmz\_6\_2\_zajacovka\_ver3\_R.aia**).

### 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 mrkiev. 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 počítadiel a tlačidlá pre spustenie novej hry a ukončenie práve bežiackej hry (vhodné umiestnenie komponentov sme dosiahli vložení 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).

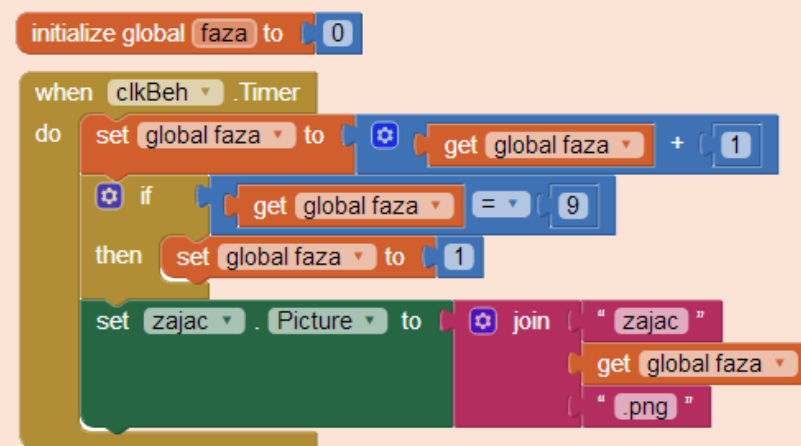
#### Pomôcky

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.

### Pomoc k riešeniu úlohy

Do projektu vložíme časovač a naprogramujeme reakciu na jeho „tiknutie“ (t. j. na udalosť `Clock.Timer`). Hodnotu premennej `faza` postupne zvyšujeme. Po dosiahnutí poslednej fázy nastavíme do globálnej premennej opäť hodnotu 1:



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

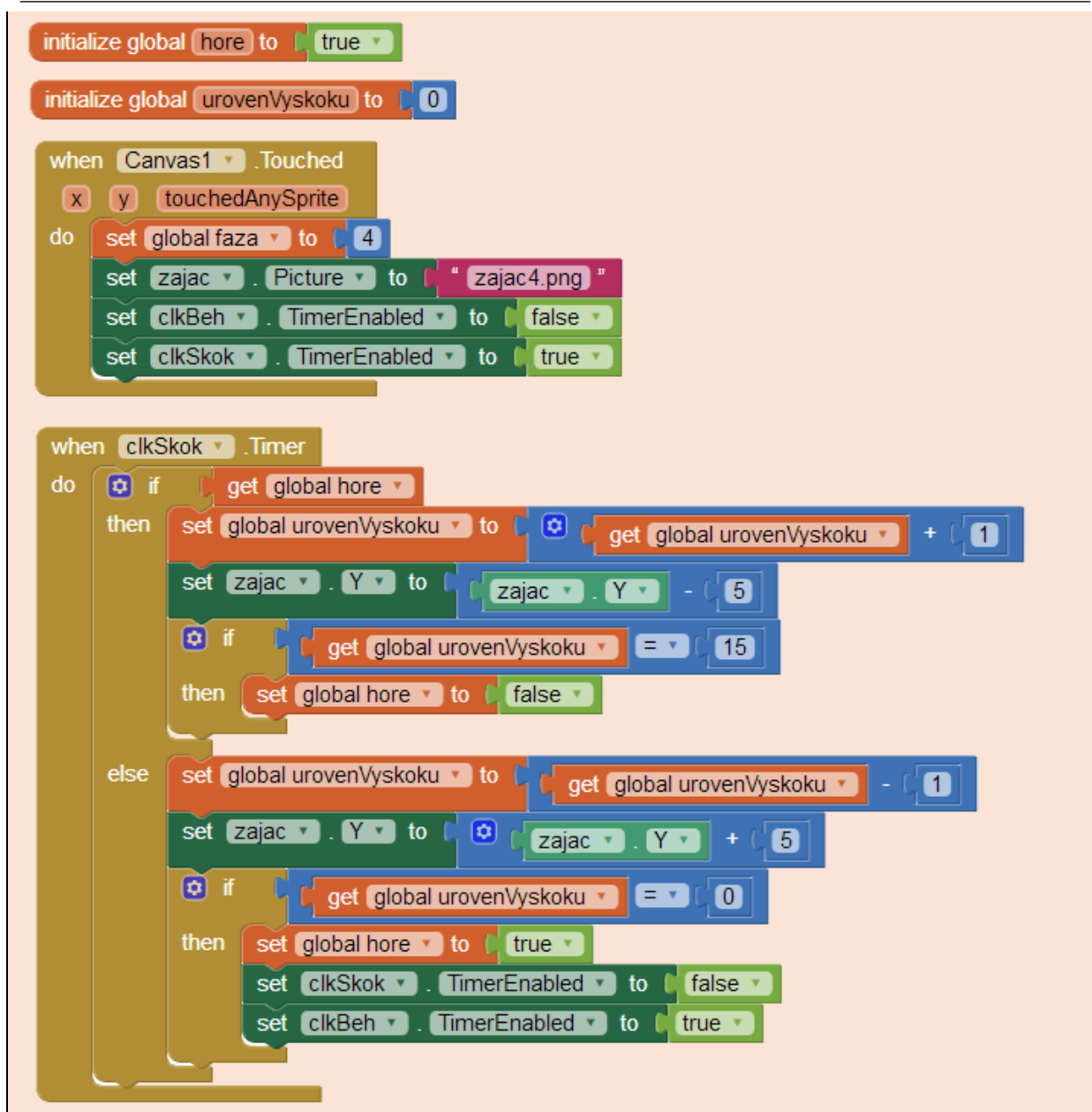
Výskokom zajaca rozumieme plynulé posúvanie sa zajaca z jeho základnej pozície smerom nahor a následné plynulé zostúpenie smerom nadol.

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 zlaďiť 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 časovať riadiaci animáciu behu. Počas skoku môže mať zajac nastavenú fázu 4 naznačujúcu skok.

### Pomoc k riešeniu úlohy

V ukážke nižšie je vidno, ako 15 opakovaniami a znižovaním súradnice `y` o 5 pixelov dosiahneme výskok zajaca o 75 pixelov nahor. Po 15. posune zmeníme smer pohybu nastavením logickej globálnej premennej `hore` na `false`:



```

initialize global hore to true
initialize global urovenVyskoku to 0

when Canvas1 Touched
  x y touchedAnySprite
  do
    set global faza to 4
    set zajac . Picture to "zajac4.png"
    set clkBeh . TimerEnabled to false
    set clkSkok . TimerEnabled to true

when clkSkok . Timer
  do
    if get global hore
    then
      set global urovenVyskoku to get global urovenVyskoku + 1
      set zajac . Y to zajac . Y - 5
      if get global urovenVyskoku = 15
      then
        set global hore to false
    else
      set global urovenVyskoku to get global urovenVyskoku - 1
      set zajac . Y to zajac . Y + 5
      if get global urovenVyskoku = 0
      then
        set global hore to true
        set clkSkok . TimerEnabled to false
        set clkBeh . TimerEnabled to true
  
```

### Úloha 3

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

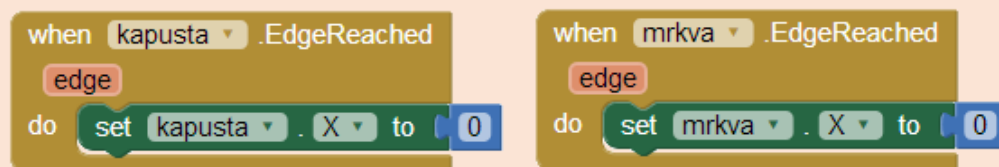
#### Pomôcky

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.

## Pomoc k riešeniu úlohy

Riešenie tejto úlohy je triviálne, najviac času zaberie žiakom testovanie aplikácie s rôznymi nastaveniami rýchlostí pohybu (vlastnosť `ImageSprite.Interval` v kombinácii s vlastnosťou `ImageSprite.Speed`) pre kapustu a mrkvu. V oboch prípadoch chceme, aby sa sprajty pohybovali zľava doprava, vlastnosť `ImageSprite.Heading` preto nastavíme na 0.

Kapuste aj mrkvu ešte naprogramujeme reakciu na náraz do pravého okraja obrazovky:



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

## Úloha 4

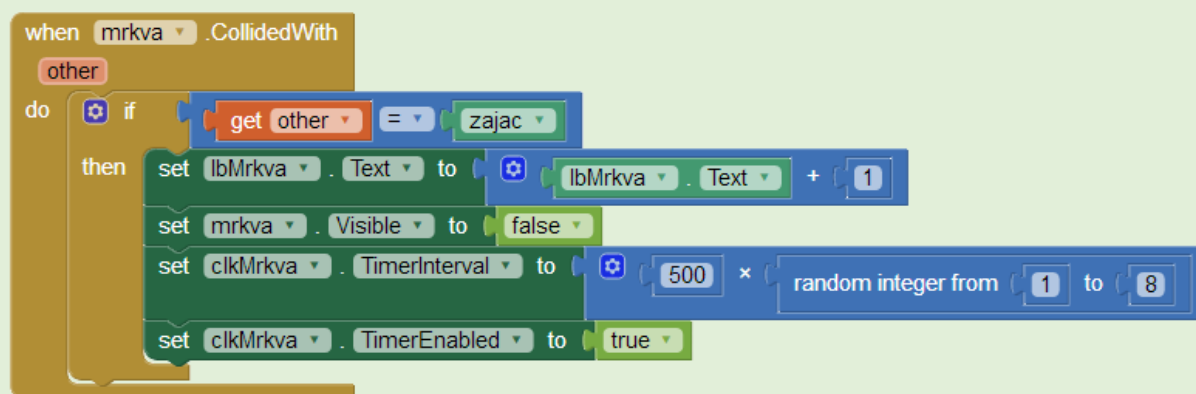
Upravte riešenie tak, aby sa ďalšia mrkva po zrážke so zajacom objavila na ľavom okraji obrazovky o *náhodne zvolený počet sekúnd* a letela vpravo *náhodne zvolenou rýchlosťou*. Kapusta by sa v prípade zrážky so zajacom mohla začať guľať ihneď (t. j. v jej prípade nebudeme rozlišovať medzi nárazom na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

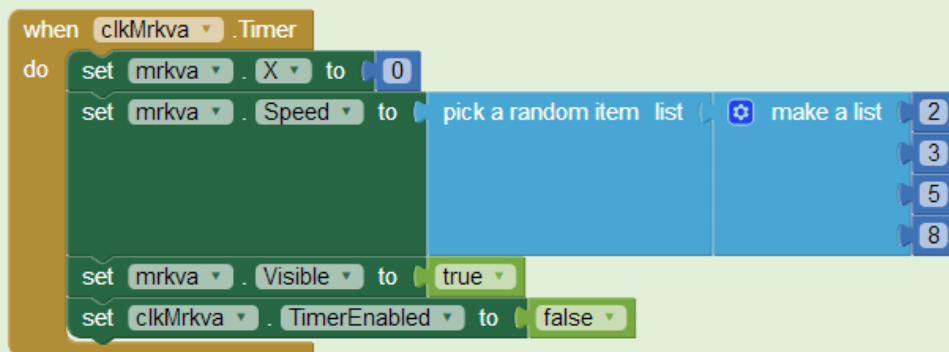
## Vysvetlíme si

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 mrkvu z ľavej strany po predchádzajúcej zrážke so zajacom:





Naplánovať udalosť „objavenie sa mrkvy zľava“ znamená nastaviť pri zrážke príslušnému časovaču vlastnosti tak, aby tikoľ 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`.

### Pomoc k riešeniu úlohy

Porovnanie parametra `other` s objektom reprezentujúcim zajaca nebolo nutné, keďže s iným sprajtom sa mrkva ani kapusta v našej verzii hry nezrážajú. Tento zápis je však všeobecnejší a zjednoduší prípadné neskoršie úpravy či rozširovanie aplikácie.

Pri experimentovaní s hrou môže vysvitnúť, že zrážky vznikajú aj v situáciách, keď sa sprajty očividne nedotýkajú (napr. zajacovi sa podarí mrkvu chytiť aj vtedy, keď ju má už za ušami). Dôvodom je fakt, že v App Inventore sa pri vyhodnocovaní kolízií berú do úvahy obdĺžniky, do ktorých sú obrázkové tvary sprajtov vpísané. Obdĺžnik s obrázkom zajaca a obdĺžnik s obrázkom mrkvy sa môžu prekryvať aj svojimi „prázdny“ plochami, čo z pohľadu hráča nie je vhodné považovať za zrážku.

### 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.

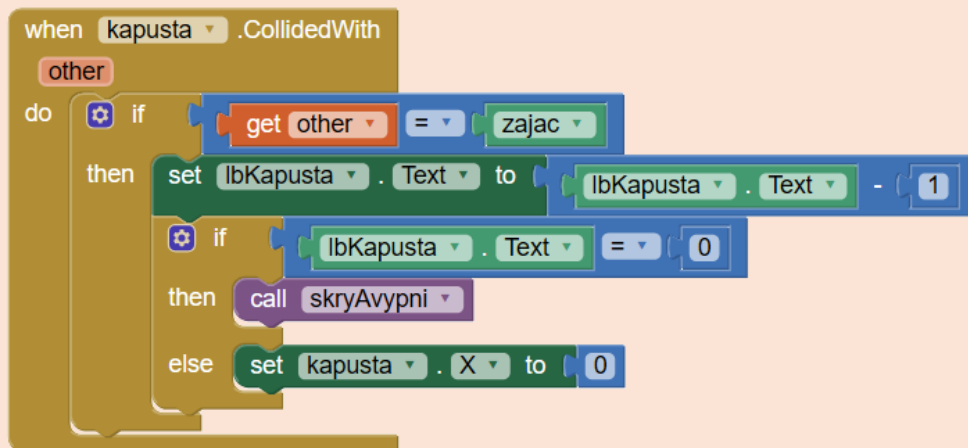
### Pomôcky

Počítadlo súvisiace s gúľajúcimi 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.



### Pomoc k riešeniu úlohy

V udalostnom bloku, v ktorom reagujeme na zrážku kapusty so zajacom, ošetríme príkazom vetvenia situáciu, keď hráč príde o všetky životy: v procedúre `skryAvypni` skryjeme všetky sprajty a vypneme časovače. Prebiehajúcu hru týmto ukončíme:



Žiaci môžu vymyslieť aj krajší spôsob ukončenia (napr. zobrazíť skrytý text alebo sprajt s gratuláciou).

### 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

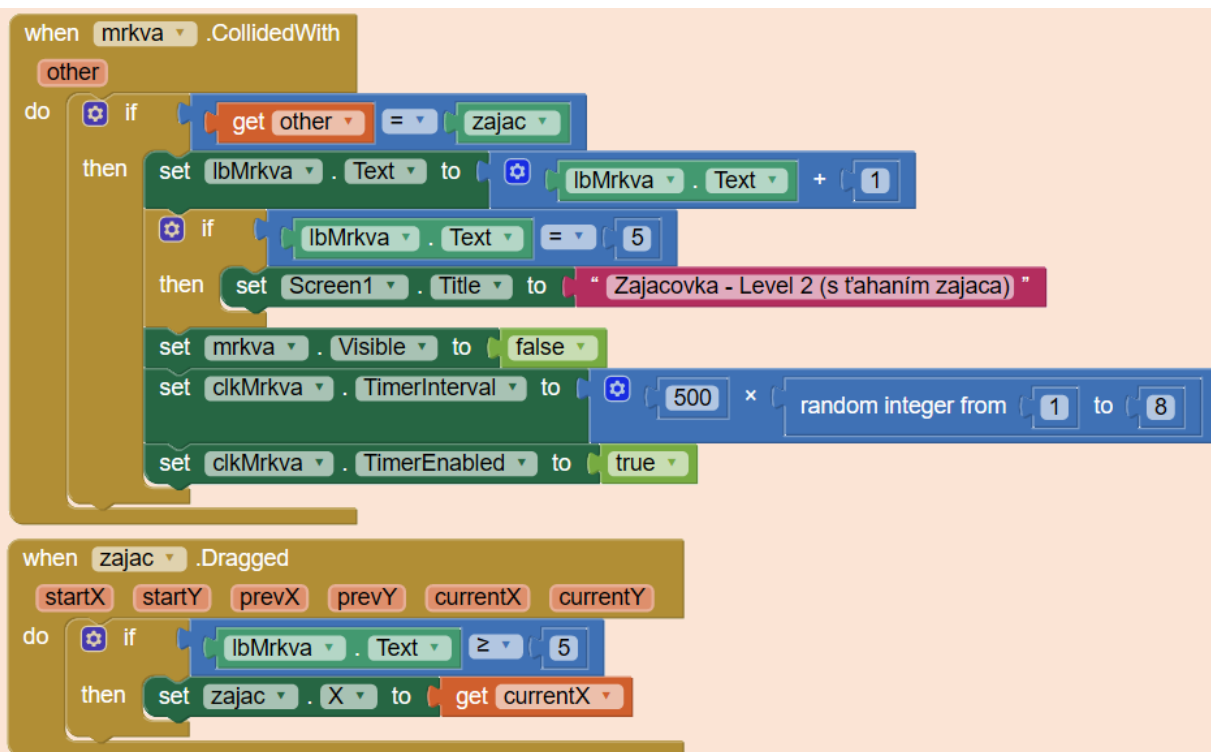
Zamyslite sa, čo znamená posúvať zajaca ťahaním vpred alebo vzad v horizontálnom smere. Ktorá súradnica sprajtu sa pri ťahaní bude meniť? Ktorý udalostný blok budeme potrebovať?

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

### Pomoc k riešeniu úlohy

V priebehu testovania môžeme ako hranicu na postup do druhého levelu zvoliť malú hodnotu, napr. 5 chytených mrkiev. Ťahanie zajaca má fungovať len vtedy, ak sme v druhom leveli. Túto podmienku vidíme v udalostnom bloku `when zajac.Dragged`. V reakcii na ťahanie presúvame zajaca v horizontálnom smere. Presnejšie: meníme jeho súradnicu X podľa aktuálnej pozície prsta, ktorým ho po plátne ťaháme:





Projekt **pmz\_6\_2\_zajacovka\_ver1\_R.aia** predstavuje verziu hry po vyriešení Úlohy 1 až 6.

### Úloha 7

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

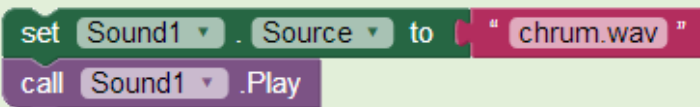
Dovoľte tiež nastavovať niektoré parametre hry používateľovi (napr. počet životov a počet mrkvičiek, ktoré sú potrebné na prechod do ďalšieho levelu).

### 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:



Na prehrávanie krátkych zvukových efektov použijte 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:

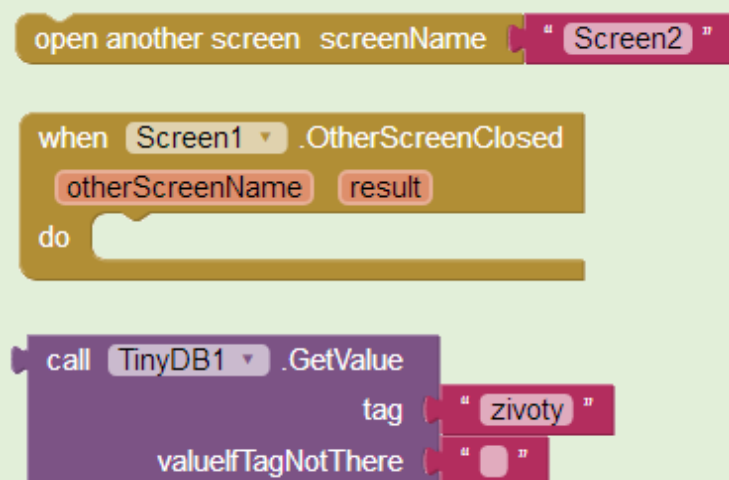


Nastavovanie parametrov hry by sa malo realizovať na samostatnej obrazovke. Pridajte preto do projektu ďalší komponent typu `Screen`, navrhňte vzhľad tejto obrazovky. Používateľovi

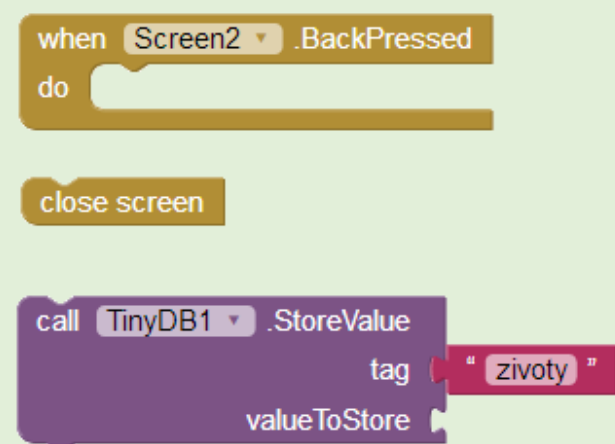
môžete dať na výber rôzne hodnoty parametrov s využitím komponentu `Spinner`. Na hlavnú obrazovku sa môžete vrátiť stlačením tlačidla potvrdzujúceho zmenu alebo zabudovaným tlačidlom pre návrat späť.

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)



Obrazovka `Screen2` s ovládacími prvkami na výber hodnôt parametrov



### Pomoc k riešeniu úlohy

Projekt `pmz_6_2_zajacovka_ver2_R.aia` predstavuje verziu hry po vyriešení Úlohy 7.

Pri spúšťaní hry (v udalostnom bloku `Screen.Initialize`) je potrebné skontrolovať, či už sú v databáze zapísané nejaké hodnoty parametrov. Ak ide o prvé spustenie a databáza je prázdna, treba do nej uložiť východiskové hodnoty. Aktuálne hodnoty parametrov získaných z databázy kvôli prehľadnosti ukladáme do globálnych premenných.

### 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 gúľajú 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).

#### Pomoc k riešeniu úlohy

Projekt **pmz\_6\_2\_zajacovka\_ver3\_R.aia** predstavuje verziu hry s niektorými vylepšeniami (pozastavenie hry, zapínanie/vypínanie zvuku).

Tlačidlá *Nový hra* a *Koniec hry* sme v tejto verzii nahradili jediným tlačidlom `btnOvladanie`, na ktorom meníme nápis podľa toho, v akom stave sa hra nachádza. Hra môže bežať (vtedy sa dá zastaviť), byť zastavená (vtedy sa v nej dá pokračovať) alebo nebeží (vtedy sa dá spustiť nová hra). Informáciu o tom, či hra beží, obsahuje globálna premenná. Pri zastavovaní bežiacej hry si musíme zapamätať, v akom stave sme hru prerušili (Akou rýchlosťou letí mrkva zobrazená na obrazovke? Je zajac práve vo výskoku?)

Zapínanie a vypínanie prehrávania zvukov sa dá vyriešiť napr. pridaním komponentu typu `Checkbox` do horného pásu nad plátnom. Na tie miesta v zdrojovom kóde, kde sa má zvuk prehrávať v závislosti od voľby používateľa, stačí pridať príkaz vetvenia.

Odporúčaný priebeh výučby		
Činnosť učiteľa	Činnosť žiaka	Poznámky
<i>1. hodina – Úvod, Úlohy 1, 2 a 3</i>		
<p>Uvedenie témy projektu: učiteľ diskutuje so žiakmi o počítačových hrách.</p> <p>Cieľom tohto individuálneho projektu je vytvorenie vlastnej mobilnej hry typu „plošinovka“.</p> <p>Učiteľ usmerňuje žiakov pri skúmaní východiskového projektu s návrhom vzhľadu a základnej štruktúry vzorovej aplikácie.</p> <p>Nasleduje riešenie konkrétnych úloh.</p> <p>Učiteľ nechá žiakov, aby navrhli riešenie jednotlivých problémov, moderuje diskusiu.</p> <p>Úloha 1: Zadanie o striedaní fáz v animačnom cykle behu.</p> <p>Úloha 2: Zadanie o výskoku zajaca.</p> <p>Úloha 3: Zadanie o automatickom pohybe mrkvy a kapusty.</p>	<p>Žiaci reagujú na otázky, hovoria o vlastných skúsenostiach.</p> <p>Žiaci importujú východiskový projekt z počítača do App Inventora a preskúmajú ho.</p> <p>Žiaci najprv diskutujú o riešení problému, slovne sformulujú hlavnú myšlienku riešenia, programujú už samostatne, testujú svoje riešenia na tabletoch.</p>	<p>Námety na otázky pre žiakov sme uviedli v úvode kapitoly.</p> <p>Diskusia by mala smerovať k uvažovaniu o špecifikách mobilných hier v súvislosti s vlastnosťami mobilných zariadení.</p> <p>Učiteľ pripraví pre žiakov projekt <b>pmz_6_2_zajacovka_ver0.aia</b> na stiahnutie.</p> <p>Učiteľ upriami pozornosť žiakov na všetky dôležité prvky rozpracovaného riešenia (vymenovali sme ich v zadaní Úlohy 1).</p> <p>S komponentom <b>Clock</b> už žiaci pracovali. Novou skúsenosťou bude pre nich použitie časovača na naplánovanie jednorazovej udalosti v Úlohe 3.</p> <p>V projekte bude potrebné použiť viacero komponentov typu <b>Clock</b>, učiteľ by mal žiakov upozorniť na to, aby ich vhodne pomenovali.</p>
<i>2. hodina – Úloha 4 a Úloha 5</i>		
<p>Úloha 4: Zadanie o zrážkach mrkvy a kapusty so zajacom.</p>	<p>Žiaci pokračujú v programovaní aplikácie.</p>	<p>Žiaci by mali naprogramovať do konca vyučovacej hodiny riešenie, ktoré je funkčné, hrateľné.</p>

<p>Úloha 5: Zadanie o počítadlách, ukončení hry a spustení novej hry.</p>		<p>Učiteľ žiakov navedie na použitie procedúr, ktoré sa zavolajú pri spúšťaní novej hry, resp. pri ukončení bežiackej hry.</p> <p>Jednotlivé verzie aplikácie by si žiaci mali priebežne ukladať ako „checkpointy“ projektu. V prípade problémov sa tak budú môcť vrátiť k predošlej, ešte funkčnej verzii projektu.</p>
<p><i>Domáca úloha</i></p>		
<p>Úloha 6: Zadanie o ťahaní zajaca v horizontálnom smere.</p> <p>Úloha 7: Zadanie o prehrávaní hudby na pozadí a o zvukových efektoch.</p> <p>Pri zadávaní domácej úlohy učiteľ pripomenie žiakom komponenty <code>Player</code> a <code>Sound</code>. Uistí sa tiež, že žiaci rozumejú zadaniu o ťahaní zajaca v horizontálnom smere.</p>	<p>Žiaci pokračujú v programovaní doma.</p>	<p>Pridaním možnosti ťahať zajaca v horizontálnom smere je hra interaktívnejšia, zaujímavejšia.</p> <p>Učiteľ môže žiakov motivovať ukážkou tej verzie aplikácie, v ktorej je už funkcionálna z Úlohy 6 a 7 implementovaná.</p>
<p><i>3. hodina – Úloha 7, Individuálny projekt</i></p>		
<p>Učiteľ skontroluje (príp. ohodnotení) vypracovanie domácej úlohy, pýta sa žiakov na problémy, ktoré pri programovaní mali.</p> <p>Učiteľ ešte raz špecifikuje požiadavky na výsledný produkt (rozdá ho žiakom aj v písomnej podobe):</p> <ul style="list-style-type: none"> <li>- hlavný hrdina je ovládaný dotykovými gestami (dotyk, ťahanie, švihnutie - <i>Flung</i>),</li> </ul>	<p>Žiaci prezentujú svoje riešenia, porovnávajú si ich navzájom.</p> <p>Žiaci premýšľajú nad vlastnou hrou, konzultujú svoje nápady s učiteľom.</p>	<p>Je možné, že žiaci doma vylepšili aplikáciu implementovaním vlastných nápadov. Žiakom necháme dostatočný priestor na to, aby sa mohli medzi sebou podeliť o skúsenosti, pochváliť sa. Úsilie navyše by mal učiteľ primerane oceniť.</p> <p>Riešenie Úlohy 7 s ďalšou obrazovkou na nastavovanie parametrov hry (projekt <b>pmz_6_2_zajcovka_ver2_R.aia</b>), môže učiteľ poskytnúť žiakom ako študijný materiál.</p>

<ul style="list-style-type: none"> <li>- prechádza postupne aspoň 3 levelmi so zvyšujúcou sa náročnosťou,</li> <li>- zbiera predmety, za ktoré má body,</li> <li>- vyhýba sa nepriateľom, pri kolízii s nimi stráca životy,</li> <li>- <b>aspoň 1 ďalšia funkcionálna</b> podľa vlastného uváženia,</li> <li>- aplikácia musí byť funkčná a zverejnená v Galérii projektov prostredia App Inventor.</li> </ul> <p>Cieľom je vytvoriť funkčný prototyp, ktorý má zmysel zverejniť pre ostatných používateľov.</p> <p>Na nasledujúcej hodine by mali mať žiaci premyslený vlastný námet na hru a pripravené aspoň mediálne súbory, ktoré použijú.</p>		<p>Námet na hru môže byť ľubovoľný, nie však násilný alebo necitlivý voči niekomu inému.</p> <p>Žiaci sa môžu inšpirovať prezeraním si hier dostupných v online obchode Google Play alebo na webových herných portáloch. Pri úvahách o vlastnej hre sa môžu inšpirovať aj námetmi na vylepšenie hry <i>Zajacovka</i> uvedené v závere kapitoly.</p> <p>Ak chcú žiaci použiť vo svojej hre grafické alebo zvukové súbory z internetu, mali dbať na to, aby neporušili autorské práva.</p> <p>Popis aplikácie pri publikovaní projektu v <i>Galérii</i> by mal byť stručný ale výstižný, prehľadný, aby ním oslovili potenciálneho záujemcu. Ak žiaci ešte v <i>Galérii</i> projekt nezverejňovali, učiteľ im ukáže postup, tiež príklad vhodného a nevhodného popisu projektu na aplikácii vybranej z <i>Galérie</i>.</p>
<b>4. hodina - Práca na projekte</b>		
<p>Učiteľ je v roli konzultanta. Kontroluje však aj napredovanie žiakov a usmerňuje ich.</p>	<p>Žiaci programujú v App Inventore, testujú riešenie na mobilnom zariadení.</p> <p>V závere vyučovacej hodiny by mal každý zo žiakov ukázať učiteľovi aktuálny stav svojej aplikácie a naznačiť, na čom plánuje pracovať ďalej doma.</p>	<p>Komunikácia a vzájomná pomoc žiakov je vítaná, učiteľ zasahuje vo chvíľach, keď si žiaci riešením problému nevedia dať rady.</p> <p>Niektorí žiaci budú chcieť možno rozvíjať námet zo vzorovej hry, prípadne adaptujú pôvodnú verziu úpravou grafiky, zvukových efektov a pravidiel. Aj to je prijateľný prístup.</p>
<b>5. hodina – Práca na projekte</b>		
<p>Učiteľ pomáha žiakom pri hľadaní a odstraňovaní chýb. Zaujíma sa o</p>	<p>Žiaci ďalej pracujú na svojich projektoch, v prípade potreby využijú</p>	<p>Ak má žiak problémy s implementáciou svojho námetu (precenil svoje sily, príp. ho</p>

<p>aktuálny stav riešenia, oceňuje nápady žiakov a pokrok v práci.</p> <p>V závere vyučovacej hodiny žiakom pripomenie spôsob odovzdania a prezentácie projektu.</p>	<p>prítomnosť učiteľa, spolužiaka.</p>	<p>v dosiahnutí cieľa objektívne obmedzuje samotný App Inventor), učiteľ by mal žiakovi navrhnúť obmenu pôvodného námetu, ktorá bude pre žiaka zvládnuteľná.</p>
<p><b>6. hodina – Hodnotenie projektov</b></p>		
<p>Učiteľ hodnotí projekty žiakov.</p> <p>Po prezentácii žiaka položí žiakovi doplňujúcu otázku týkajúcu sa niektorého z riešených podproblémov.</p> <p>Učiteľ dá priestor na otázky aj ostatným žiakom.</p> <p>V závere hodiny učiteľ ocení nápady a úsilie, ktoré pri ich realizácii žiaci vynaložili. Vyzve žiakov, aby zhodnotili, ako sa im na projekte pracovalo, či sú s výsledkom svojej práce spokojní.</p>	<p>Každý žiak ukáže stránku svojej aplikácie v Galérii, spustí ju v mobilnom zariadení, predstaví jej možnosti.</p> <p>Žiaci si môžu aplikáciu stiahnuť do svojho tabletu či smartfónu a vyskúšať si ju.</p> <p>Žiak odpovedá na otázku učiteľa a spolužiakov.</p>	<p>Pri hodnotení projektu by mal učiteľ zobrať do úvahy:</p> <ul style="list-style-type: none"> <li>- či žiak splnil požiadavky,</li> <li>- tvorivý prístup k riešeniu,</li> <li>- dôslednosť pri realizácii námetu (napr. zmysel pre detail, úroveň grafického spracovania).</li> <li>- popis projektu v Galérii,</li> <li>- prezentáciu vytvorenej hry pred spolužiakmi,</li> <li>- odpoveď na doplňujúcu otázku.</li> </ul> <p>Môže tiež prebehnúť anonymné hlasovanie o najlepšej aplikácii.</p> <p>Vyučovacia hodina venovaná hodnoteniu projektov by mala prebehnúť 1 týždeň (najviac 2 týždne) po poslednej projektovej hodine v škole, aby mali žiaci čas aplikáciu dokončiť podľa svojich predstáv.</p>