

PV178 - FI Keyboard Hero

1 Prológ

Táto úloha je zameraná na precvičenie generiky, udalostí a rozširujúcich metód. Veľkú časť tejto úlohy ale tvorí aj práca s už rozpracovaným projektom, pochopenie jeho dokumentácie a následné využitie niekoľkých predprogramovaných metód, o ktorých v súčasnosti väčšina z vás asi nevie, ako presne interne fungujú.

V prípade nejasností môžete využiť príslušné vlákno v rámci predmetového diskusného fóra, prípadne napísať autorovi úlohy na **410452@mail.muni.cz**.

2 Zadanie

Vitajte v našej FI firme. Vaším prvým projektom bude vytvoriť konzolovú aplikáciu, ktorá bude umožňovať užívateľovi hrať na virtuálnom klavíri pomocou klávesnice. Naš klavír bude oveľa zjednodušený, bude obsahovať iba tóny C dur stupnice štvrtej oktávy (bez tónu C piatej oktávy). Každý tón má svoju frekvenciu, priradenú klávesu na klávesnici a svoj názov. Presné údaje sú znázornené v tabuľke.

Klávesa na klávesnici	Názov tónu	Frekvencia
a	C	261 Hz
s	D	293 Hz
d	E	330 Hz
f	F	349 Hz
g	G	392 Hz
h	A	440 Hz
j	H	494 Hz

Súčasťou tejto úlohy bude taktiež implementácia hry FI Keyboard Hero¹. V tejto hre si hráč vyskúša hranie niektorých pesničiek na klavíry podľa daného vzoru a bude sa snažiť zahrať tóny pesničky čo najpresnejšie, aby získal čo najviac bodov.

¹Táto hra je inšpirovaná hrou Guitar Hero. Tí čo ju nepoznáte, tu je z nej [ukážka](#)

2.1 Popis hry

Každý vzor pesničky je uložený v textovom dokumente, kde sú za sebou zapísané znaky, ktoré zodpovedajú klávesám na klávesnici. Medzi nimi môžu byť medzery (predpokladáme, že vzor nikdy nekončí medzerou). Ak je vo vzore znak *a*, hráč musí v danom momente stlačiť klávesu **a**. Ak tam je medzera, hráč v danom momente nemôže stlačiť žiadnu klávesu.

Pozícia znaku určuje, kedy hráč musí stlačiť danú klávesu. Znak na pozícii *i* treba stlačiť v pevne danom časovom intervale (300ms) pred znakom na pozícii *i + 1*. Napríklad vzor “**as f**” nám vraví, že:

1. hráč má 300 ms na stlačenie klávesy **a**
2. následne má 300 ms na stlačenie klávesy **s**
3. následne má 300 ms počas ktorých nesmie stlačiť žiadnu klávesu
4. následne má 300 ms na stlačenie **f**

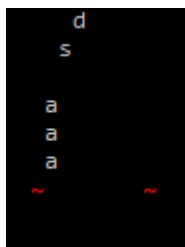
Ak nestihne stlačiť požadovanú klávesu alebo sa v hociktovej pasáži pomýli a stlačí inú, než je treba, strhne sa mu bod.

Hráč začína hru s maximálnym možným počtom bodov, ktorý zodpovedá dĺžke pesničky, čiže dĺžke jej vzoru. V tomto prípade by teda hráč začínal so 4 bodmi. Predstavme si situáciu, keď hráč ide hrať túto našu pesničku so vzorom (“**as f**”):

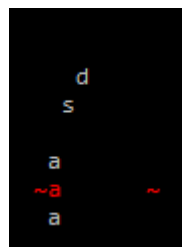
1. prvých 300 ms stlačí omylom klávesu **s**
2. počas ďalších 300 ms nestihne stlačiť klávesu **s**
3. následne počas 300 ms nestlačí nič
4. následne počas 300 ms stihne stlačiť klávesu **f**

Na konci pesničky má teda 2 body.

Na vizualizáciu toho, kedy a akú klávesu má hráč stlačiť, sa využíva trieda **Display**, ktorú už máte implementovanú a netreba ju nijak pozmeňovať. Funguje tak, že do konzolovej aplikácie postupne vypisuje znaky zo vzoru pesničky a oni sa každých 300 ms posunú o jednu úroveň nižšie. Ak sa dostanú do farebne znázornenej oblasti, znamená to, že hráč má 300 ms na to, aby danú klávesu stlačil (pretože za 300 ms sa posunie nižšie a už nebude vo farebnej oblasti). Ak sa posledný znak zo vzoru dostane do pasáži, tak hra po 300 ms skončí a hráčovi sa vypíše počet získaných bodov.



hra v čase t



hra v čase $t + 600$ ms

2.2 Solution

Našťastie nemusíte celú hru programovať od začiatku. Jeden programátor už dostal túto úlohu, lenže sa mu ju nechcelo dokončiť (tak dostal 0 bodov a padáka). Solutiona obsahuje adresár **Songs**, kde sú nachystané vzory nejakých známych pesničiek. Obsahuje aj adresár **Utils** v ktorej sú nasledovné triedy.

2.2.1 Sounder

Táto trieda rieši prehratie zvuku po stlačení klávesy od užívateľa. Metóda *MakeSound* zahrá zvuk takým spôsobom, aby to nerušilo načítavanie ďalších kláves (o tom, ako to funguje sa budete učiť za pár týždňov). Prakticky vám stačí vedieť, že tejto metóde dáme frekvenciu danej noty, ktorú má zahráť a ona ju zahrá pomocou metódy *Console.Beep*.

2.2.2 Reader

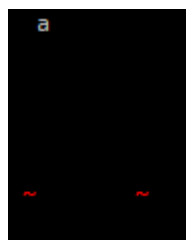
Trieda **Reader** je zodpovedná za načítanie pesničky zo súboru a spracovanie vstupu od užívateľa pri hraní. Využíva síce taktiež prvky jazyka C#, ktoré sa budete učiť trochu neskôr, ale pasáže kódu, ktoré s nimi pracujú, stihol náš bývalý programátor predpripraviť. Pre vás je dôležité pozrieť si dokumentáciu daných metód a z nej zistiť, ako sa s touto triedou pracuje.

Nanešťastie niektoré metódy tejto triedy nie sú dokončené, treba ich spraviť. Metódy *OnKeyPressed* a *OnKeyNotPressed* by mali vyvolávať udalosti, na ktoré môže zareagovať trieda využívajúca náš **Reader**. Nezabudnite, že udalosti musia obsahovať aj informácie, ktoré dostávajú tie dve metódy v argumentoch. Metóda *GetInput* je naprogramovaná chybné, triede **Sounder** predáva vždy iba frekvenciu 400, my by sme ale chceli, aby jej predávala frekvenciu, ktorá odpovedá načítanej klávese.

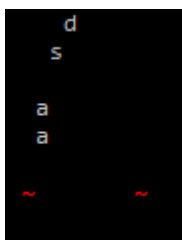
2.2.3 Displayer

Táto metóda je zodpovedná za výpis informácií o stave hry do konzoly. Túto triedu využíva **Reader** a jej fungovanie už bolo spomenuté vyššie. Na čo chceme

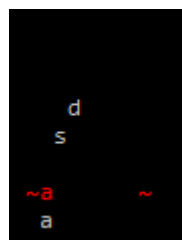
ale upozorniť je, že v stave, keď má hráč stlačiť prvú klávesu zo vzoru, je argument *position* metódy *ActualDisplay* rovný 6 (je to len z toho dôvodu, aby sa hráč mohol pripraviť na začiatku hry pred tým, ako začne pesnička). Pre istotu udávame zopár príkladov, ako sa táto metóda zachová pri vzore “aa sd” s rôznymi hodnotami argumentu *position*.



position = 0



position = 4



position = 7

3 Vaša robota

1. Implementujte nedorobené metódy triedy **Reader** – *KeyPressed* a *KeyNotPressed*.
2. Vytvorte triedu, ktorá implementuje rozhranie *IGame*. Na začiatku hry sa hráča spýta, akú pesničku chce hrať – musí zadať presný názov textového súboru uloženého v adresári Songs. Následne sa využije trieda **Reader**², nech načíta vzor pesničky z daného súboru a nech zapne hru. S vašou triedou bude komunikovať výhradne formou udalostí – v oboch prípadoch (či už hráč v daný okamih stlačil klávesu alebo nie) sa vo vašej triede overí, či hráč nespravil chybu. Na záver sa vypíše, koľko bodov hráč získal. Ak ste všetko správne naprogramovali, hra by vám už mala ísť spustiť a fungovať.
3. Hra nám síce funguje, ale všetky klávesy vydávajú ten istý tón. Navrhnite preto kolekciu, do ktorej sa budú dať efektívne uložiť dáta o tónoch zobrazených v tabuľke na 1. strane. Požadujeme, aby nám táto kolekcia umožnila efektívnu prácu s týmito dátami. Síce v našej zjednodušenej verzii je názov tónu vždy iba jeden znak, musíme predpokladať, že túto našu kolekciu budú chcieť ľudia používať v budúcnosti aj na ukladanie tónov, ktoré ako názvy používajú čísla, reťazce znakov (napríklad tón C#) alebo hocičo iné. Chceme mať teda túto kolekciu generickú.

Taktiež chceme, aby táto kolekcia podporovala metódu *AddRange*³, ktorá do nej pridá danú kolekciu záznamov (aby sme nemuseli pridávať do nej tóny po jednom) a metódu *GetInfo*, ktorá vráti informáciu o tóne, ktorý zodpovedá danej klávese na klávesnici (jeho názov a frekvenciu).

²Treba brať na vedomie, že **Reader** implementuje **IDisposable**

³Je dosť nepravdepodobné, že dostatočne efektívna kolekcia bude mať túto metódu už implementovanú

4. Vytvorte triedu **Piano**, ktorá bude využívať vašu kolekciu. Pri vytvorení objektu sa vždy do tejto kolekcie pridajú údaje znázornené v tabuľke. Teraz už nebude problém upraviť metódu *GetInput* triedy **Reader** tak, aby trieda **Sounder** vydávala zvuky so správnou frekvenciou, podľa stlačenej klávesy.
5. Teraz by sme mali byť v stave, kedy hra veselo funguje. Vedenie našej firmy sa však pozrelo do zdrojákov vášho projektu a nepáči sa im, koľko riadkov kódu je nutných po vytvorení objektu **Reader**. Preto máte za úlohu spraviť rozširujúcu metódu triedy **Reader**, ktorá nejakým spôsobom zníži počet riadkov v kóde po vytvorení objektu tejto triedy⁴.
6. Nachádzame sa v poslednej časti tejto domácej úlohy. Po pár týždňoch od vydania našej hry nám chodia spätné väzby od hráčov, že sa im nepáči zvuk nášho klavíru. Vraj je priam otrasný. Vedenie sa preto rozhodlo, že zvuky v hre vylepšia, ale sprístupnia ich iba hráčom, ktorý si za to zaplatia.

Preto máte za úlohu stiahnuť si zvuky klavíru z učebných materiálov a upraviť triedu **Sounder**. Pridajte do nej metódu *MakeCoolSound*, ktorá podľa stlačenej klávesy prehrá zvuk, ktorý odpovedá tejto klávese⁵. Tiež treba upraviť metódu **Reader**, aby podporovala túto platenú funkciu. Čo sa týka detekovania, či hráč zaplatil, pre jednoduchosť úplne stačí, keď pridáte do konštruktoru argument typu bool. V triede **Game** si už danú informáciu zistíte podľa vášho uváženia (môžete sa napríklad spýtať užívateľa, či už zaplatil).

⁴Hodnotí sa, či viete vytvoriť a použiť rozširujúcu metódu

⁵Rada: využite triedu v mennom priestore `System.Media`