

Tetris

Hra Tetris bola vytvorená ako zápočtový program na predmet Programování I. Cieľom bolo vytvoriť funkčný program inšpirovaný klasickou hrou Tetris v programe Unity. Tento dokument popisuje hlavné časti programu, výber prostriedkov, funkciu použitých algoritmov a pod.

1 Popis hry

Hra má jednoduché pravidlá: program náhodne generuje útvary, ktoré má hráč za úlohu usporiadať čo najlepším možným spôsobom tak, aby bol zaplnený nejaký celý riadok. Hráč ich ovláda pomocou šípok, kde ľavá, pravá a dolná šípka posúvajú útvar do daného smeru a horná šípka ho otáča proti smeru hodinových ručičiek. Pokiaľ sa nejaký riadok zaplní, program ho odstráni a všetky kocky nad ním „spadnú“ o jedno políčko nižšie. Za každý odstránený riadok sa hráčovi pripočíta 1 bod, za 2 odstránené riadky 3 body, za 3 riadky 5 bodov, za 4 riadky 10 a za viac sa skóre zvýši o 3-krát počet odstránených riadkov. Pokiaľ už nie je možné ďalej ukladať kocky alebo nejaká presiahne horný okraj obrazovky, hra sa ukončí. Hráč má potom možnosť hru opakovať alebo z nej odísť.

2 Hracia plocha a herné objekty

Hra Tetris obsahuje tri scény: úvodnú scénu *Welcome*, scénu pri ukončení hry *GameOver* a samotnú hru *Level*.

Hlavným komponentom hry je tzv. mino, teda štvorec veľkosti 1×1 . Z jednotlivých mín sú vytvorené tetraminá, teda útvary zložené zo štyroch mín, ktoré tvoria jeden celok. V hre sa používa 7 typov tetramín. Z mín je vytvorená aj mriežka alebo *grid*, ktorá ohraničuje povolenú hraciu plochu.

Tetraminá aj mriežka sú objekty typu *GameObject*. Podstata posúvania tetramín je zabezpečená pomocou objektu *Transform*, čo je dieťa každého *GameObject*. *Transform* obsahuje hlavné vlastnosti prvku, napríklad pozíciu a rotáciu.

K mriežke je pripojený script *Game*, ktorý má na starosti ovládanie hry. Jednotlivé tetraminá sú spojené so scriptom *Tetramino*, ktorý ovláda ich pohyb. Úvodná a záverečná scéna majú pripojený script *MenuSystem*, ktorý ma za úlohu začať novú hru alebo hru úplne opustiť.

3 Herná logika

3.1 Začiatok hry

Po začatí hry sa náhodne vygeneruje nejaké zo siedmich typov tetramín a zobrazí sa na vrchu mriežky. Výber náhodného tetramina je implementovaná vo funkcii *GetRandomTetramino* v skripte *Game*. Táto funkcia používa vstavaný generátor náhodných čísel, ktoré vygeneruje číslo od 1 do 7 a každému číslu priradí nejaký typ tetramina. Zobrazenie tetramina v hre je zabezpečené funkciou *InstantiateTetromino*. Hráč ho odteraz môže ovládať pomocou šípiek.

3.2 Možnosť posunu na danú pozíciu

Pri pohybe tetramina je potrebné skontrolovať, či je možné sa posunúť na ďalšiu pozíciu. Treba rozlíšiť dve možnosti: tetramino opustí hraciu plochu alebo prekryje iné tetromino.

V prvom prípade pre každé jednotlivé mino tvoriace dané tetramino treba skontrolovať, či leží v povolenom priestore. Keďže ľavý dolný roh mriežky leží na pozícii [0, 0], dosiahne sa to jednoducho tým, že sa zistí, či súradnica x nepresahuje šírku mriežky a súradnica y nepresahuje jej výšku. Toto je implementované vo funkcii *IsInsideGrid* v skripte *Game*. Táto funkcia je ďalej použitá v skripte *Tetramino* vo funkcii *IsValidPosition*.

Ak chceme zistiť, či dané tetramino neprekrýva nejaké iné, musíme mať uložený stav jednotlivých políček v mriežke. Na to používame premennú *grid*, čo je dvojrozmerné pole obsahujúce hodnoty typu *Transform*. Pokiaľ je dané políčko prázdne, *grid* na tejto pozícii obsahuje hodnotu *null*. Ak je naopak obsadené, *grid* obsahuje *Transform* mino, ktoré sa tam nachádza. Funkcia *GetTransformAtPosition* v skripte *Game* vráti hodnotu typu *Transform*, ktorá sa potom použije v skripte *Tetramino* vo funkcii *IsValidPosition* na kontrolu toho, či mino nachádzajúce sa na tejto pozícii je súčasťou nejakého iného tetramina, čo by znamenalo, že tetramino sa sem nemôže posunúť.

3.3 Ovládanie tetramín

Hráč môže jednotlivé tetraminá ovládať pomocou šípok. Posun doľava, doprava a dole je implementovaný ako pričítanie trojrozmerného vektoru k vlastnosti *position* objektu *transform*, kde jednotlivé súradnice sú buď 0 alebo 1 (resp. -1) podľa toho, kam sa má mino pohnúť. Otočenie proti smeru hodinových ručičiek stlačením šípky hore je implementované podobne, ale namiesto k *position* sa pričítava vektor (0, 0, -90) k vlastnosti *rotation*.

Pri každom pohybe sa najskôr tetramino posunie do daného smeru, až potom sa skontroluje, či je nová pozícia platná volaním funkcie *IsValidPosition*. Pokiaľ nie je, tetramino sa vráti do pôvodného stavu pričítaním opačného vektoru k tomu, akým sa dostal do tejto neplatnej pozície. Pokiaľ sa tetramino do daného smeru môže pohnúť, je potrebné ešte aktualizovať údaje v premennej *grid*, a to zavolaním metódy *UpdateGrid* skriptu *Game*. Táto metóda najskôr všetky políčka v *grid* pôvodne zabrané tetraminom nastaví na *null* a potom políčkam, na ktorých sa tetramino nachádza po posunutí, priradí *transform* jednotlivých mín.

3.4 Dopadnutie tetramina a vznik nového

V prípade, že sa tetramino nemôže pohnúť viac dole, teda dopadlo na dno mriežky alebo na iné tetramino, metóda *CheckUserInput* skriptu *Tetramino* v časti kontrolujúcej stlačenie šípky

dole nastaví premennú *enabled* na false. Premenná *enabled* je súčasťou každého *GameObject* a podľa jej hodnoty je alebo nie je možné ju ovládať zo strany používateľa. Pri vzniku *GameObject* je jej hodnota true, teda sa objekt dá ovládať. Nastavením na false zafixujeme jej pozíciu a stláčanie šípok nebude mať ďalší vplyv na jej pohyb.

Nové tetramino rovnako ako na začiatku hry vygenerujeme pomocou funkcií *GetRandomTetramino* a *InstantiateTetramino*.

3.5 Odstraňovanie zaplnených riadkov

Pokiaľ je nejaký riadok plný, chceme ho odstrániť a riadky nad ním posunúť o jeden nižšie. Toto zabezpečíme viacerými funkciami v skripte *Game*.

Metóda *DeleteRows* odstráni všetky riadky, ktoré sú plné a ostatné posunie na správnu pozíciu. Najskôr iteruje hodnotami 0 až po výšku mriežky a pre každý riadok volá funkciu *IsRowFull*. Táto funkcia pre daný riadok zistí, či každé pole v grid v tomto riadku nie je null, teda či každé pole je zabrané nejakým minom. Pre každý riadok, pre ktoré to platí, *DeleteRow* volá najskôr metódu *DeleteMinosInRow* a potom *MoveAllRows*. *DeleteMinosInRow* najskôr v danom riadku dealokuje pamäť všetkých mín a potom v grid nastaví pre každý stĺpec tohto riadka hodnotu null. *MoveAllRows* zasa všetky riadky vyššie nad práve odstráneným posunie dole pomocou metódy *MoveRowDown*. *MoveRowDown* pre daný riadok nastaví v grid riadok o jeden nižšie na rovnaké hodnoty ako súčasný riadok, v súčasnom riadku všetky hodnoty vymaže a nakoniec posunie samotné miná na správnu pozíciu.

3.6 Koniec hry

Hra skončí vtedy, keď nejaké tetramino presiahne vrch vyhradenej plochy a už nemá možnosť posunúť sa. Toto sa kontroluje pre každé tetromino potom, ako dopadne, pomocou funkcie *CheckIfAboveGrid*, ktorá zistí, či súradnica y ľubovoľného mina presahuje hodnotu *gridHeight*. Pokiaľ táto situácia nastane, zavolá sa metóda *GameOver*, ktorá zobrazí scénu konca hry.

3.7 Počítanie skóre

Po každom odstránenom riadku sa hráčovi pripočíta určitý počet bodov. Ten závisí na tom, koľko riadkov naraz odstránil. Metóda `DeleteRow` okrem toho, že odstráni všetky plné riadky, takisto upraví skóre. Po každom odstránenom riadku sa inkrementuje premenná `rowsDeleted`, ktorá je potom argumentom metódy `UpdateScore`. Táto funkcia zasa v závislosti na počte riadkov zvýši hodnotu globálnej premennej `score`. Hráčovi sa skóre zobrazí ako dynamicky generovaný text.

Skóre sa zobrazí aj po skončení hry. Keďže po opustení scény sa stratia všetky jej premenné, je výsledné skóre uložené v Player Preferences. Uloží ho metóda `SaveScore`, ktorá je volaná metódou `GameOver`, teda metódou, ktorá zmení scénu z hernej na konečnú.

3.8 Prechádzanie medzi scénami

Skript `MenuSystem` obsahuje dve metódy: `Play` a `Quit`. `Play` načíta scénu `Level` a `Quit` úplne opustí aplikáciu. Obidve metódy sú použité v scénach `Welcome` a `GameOver`. Zo scén `Welcome` a `Quit` je teda možné prejsť na novú hru alebo skončiť program.

4 Spustenie hry

Zdrojový kód je dostupný na adrese <https://github.com/kszabova/Tetris>. V prípade lokálnej inštalácie Unity je možné ho importovať na svoj počítač a otvoriť ako Unity projekt.

Na adrese https://drive.google.com/open?id=1yXcT5J_W_9-sIWTAIo2A_r73ZUozzEIE je dostupný ZIP súbor obsahujúci spustiteľný súbor na Windows, teda samotnú hru, a súbor `UnityPlayer.dll`, čo je nevyhnutná súčasť na spustenie hry. Po stiahnutí a spustení súboru `Tetris.exe` sa na obrazovke objaví dialógové okno obsahujúce niekoľko možností.

Najdôležitejšie je pole *Screen*, kde sa dá vybrať rozlíšenie obrazovky. Na hru sú svojím pomerom strán najvhodnejšie rozlíšenia 1280x720 a 1920x1080. Po výbere stačí kliknúť na tlačidlo *Play!* a hra sa spustí.

5 Nástroje a zdroje

Hra bola vytvorená v programe Unity v.2017.3. Miná boli vytvorené v grafickom programe Gimp. Obrázok na pozadí hry je prevzatý z adresy <https://unsplash.com/photos/3YrppYQPoCI>, jeho autorom je Guillaume Galtier. Font použitý v hre (DK Lemon Yellow Sun) od autora Davida Kerkhoffa je dostupný na adrese <https://www.dafont.com/dk-lemon-yellow-sun.font>.