Hra Puzzle

Dokumentácia k softwaru

Michal Kovaľ
1. ročník, skupina I33
Programování II, NPRG031
September 22, 2016

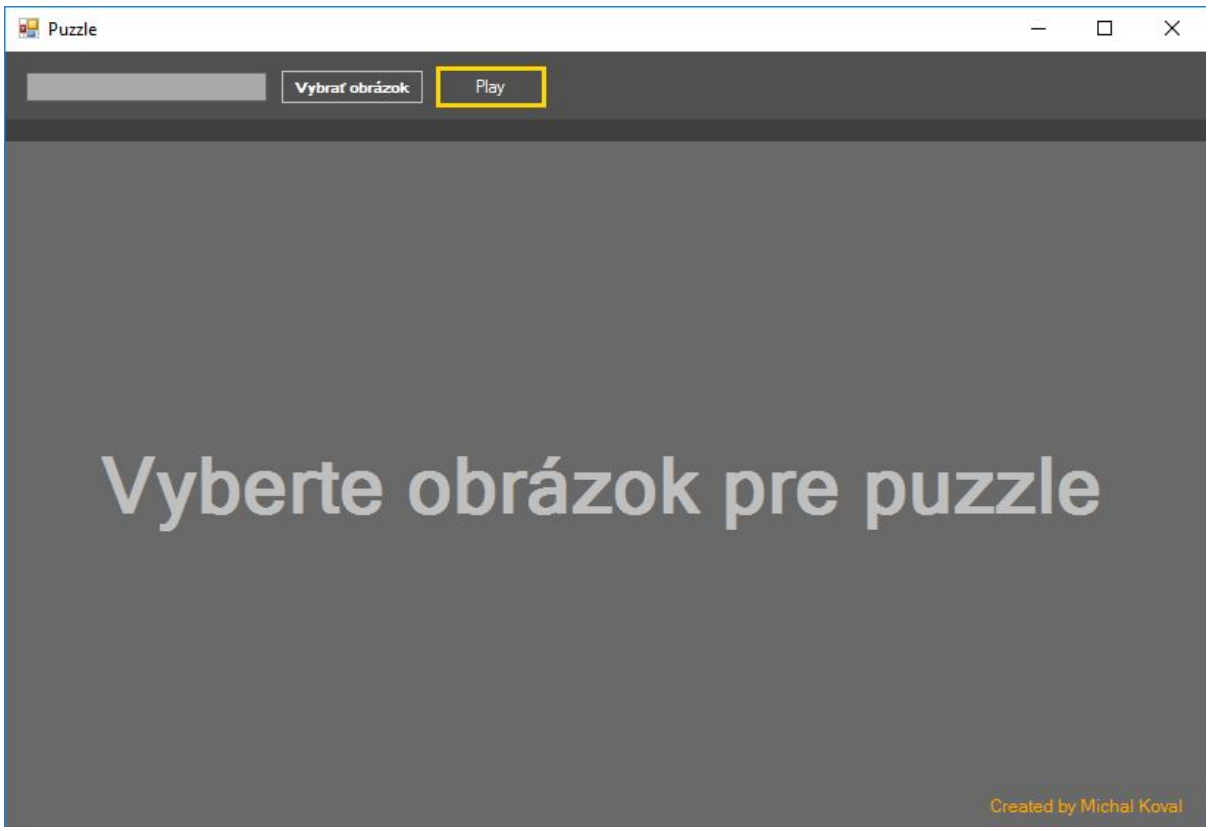
Úvod

Podstatou hry je vybrať si ľubovoľný obrázok, zvoliť si rozmery a miesto vystrihnutia z obrázka. Program následne sám obrázok rozstrihá na puzzle kúsky a náhodne ich rozmiestni po hracej ploche. Úlohou hráča je dané puzzle poskladať.

Kompilácia programu

Pre kompiláciu programu bolo použité Visual Studio 2015 Community.

Užívateľská časť

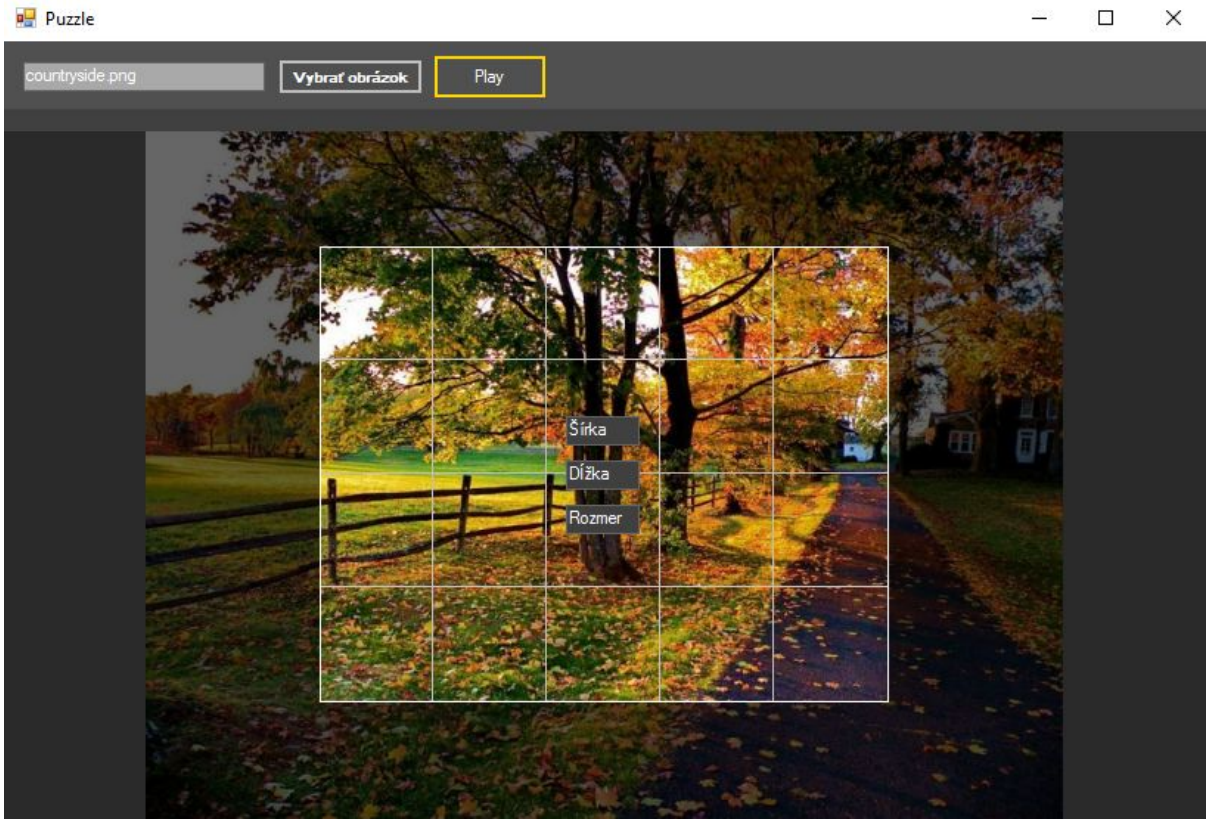


obrázok 1.

Na začiatku si užívateľ vyberie obrázok pomocou tlačidla “**Vybrať obrázok**”. Otvorí sa dialógové okno s priečinkom. Užívateľ si zvolí obrázok a klikne na “ **Otvoriť** ”. Hra sa následne spúšta pomocou tlačidla “Play”. Ak obrázok nebol zvolený, či už pri stlačení “**Cancel**” v dialógovom okne alebo pri nestlačení tlačidla “Vybrať obrázok”, program užívateľa upozorní správou “Nebol vybraný obrázok.”

Následne sa nám zobrazí obrázok s mriežkou pre vystrihnutie. (obrázok 2.) Užívateľ stlačením ľavého tlačidla myši a posunom posúva obrázok, aby si zvolil oblasť obrázka, ktorá mu najviac vyhovuje.

Zmeniť **šírku**, **dĺžku** hracieho pola alebo **rozmer** samotného kúska je možné pomocou troch editovacích textových okienok tak, že kurzorom myši sa musíme nachádzať napr. nad editovacím okienkom “Šírka”, klikneme naň a zmeníme požadovanú rozmer na iný.
Ak by daný rozmer presahoval veľkosť obrázka, text v editovacom textovom okienku sa zmení na červený. Program zabezpečí, aby mriežka nevychádzala mimo obrázok.



obrázok 2.

Hráme hru: (obrázok 3. a 4.) Pri prvom spustení hry sa užívateľovi ponad hraciu plochu objaví okienko s inštrukciami pre hranie hry (toto okienko sa už neobjaví pri kliknutí na tlačidlo “Nová hra”). Informačné okienko skryjeme kliknutím na krížik v rohu okienka.

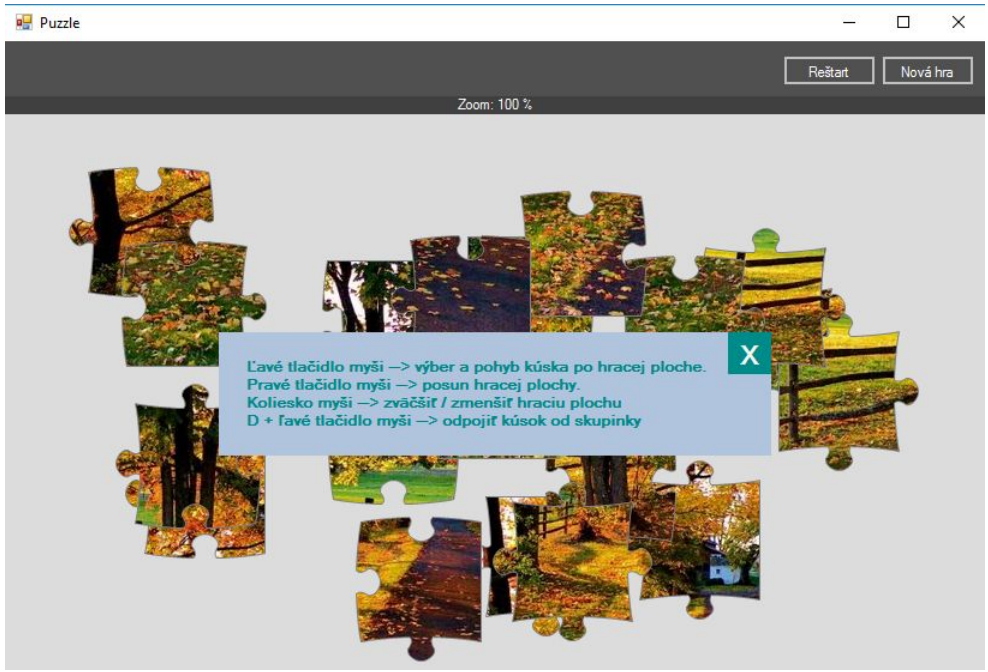
Hra sa hrá pomocu myši (**ľavé tlačidlo** - výber kúska a jeho posun, **pravé tlačidlo** - posun hracej plochy, **koliesko** myši - zoom) a klávesy “**D**” (**detach**, ak chceme kúsok odpojiť od skupinky už spojených kúskov, **D + ľavé tlačidlo myši**).

Pre výber kúska sa **kurzor musí nachádzať vo vnútri kúska**, kurzor zaberá presne na tvar kúska aký má (nestane sa teda, aby sme zvoli kúsok bez toho, že je kurzor graficky vo vnútri kúska).

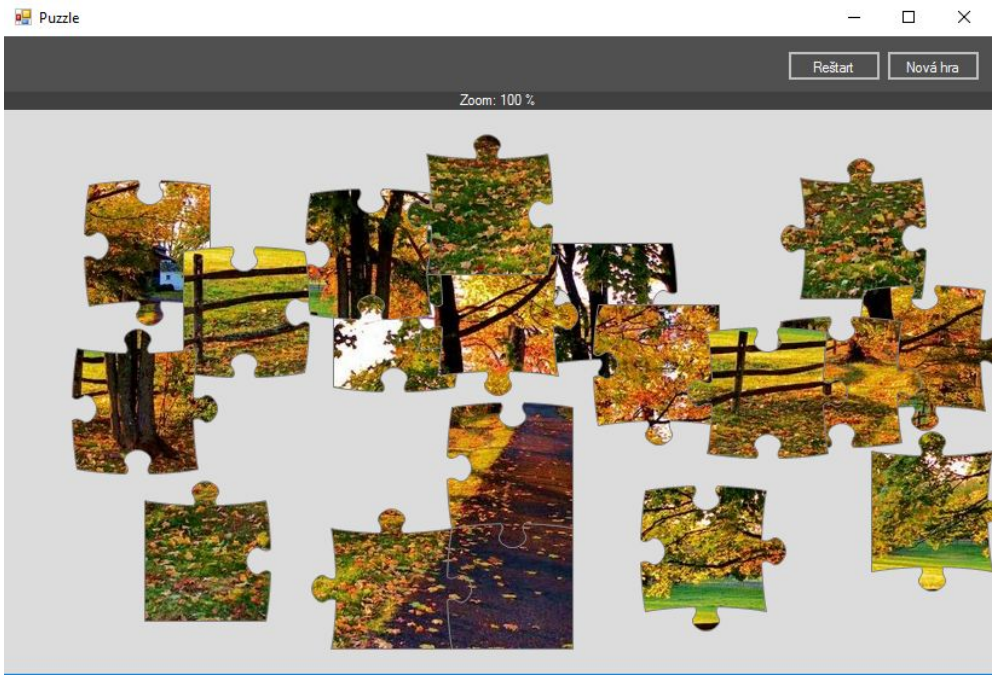
Po **uvoľnení ľavého tlačidla myši** (ak sme mali vybraný kúsok) sa kúsok zacvakne do iného kúska, ak je v jeho dostatočnej blízkosti. Kúsky do seba zacvaknú, ak im to dovolia ich zúbky. Zacvaknúť do seba môžu aj kúsky, ktoré sa graficky nezhodujú.

Ak sme vytvorili **skupinky puzzle kúskov** a chceme ich **spojiť**, je potrebne uchytiť ten kúsok zo skupinky, ktorý je najviac na kraji skupinky a prisunúť ho k druhej skupinke na miesto kde ho chceme zacvaknúť.

Pre **rozpojenie skupiniek** použijeme vyššie zmienení spôsob pre rozpojenie kúska (D + ľav. tlač. myši). Ak odoberieme kúsok, ktorý ako jediný spájal dve skupinky. Skupinky sa stanú samostatnými.



obrázok 3.



obrázok 4.

Tlačidlo “**Reštart**” slúži na opätovné náhodne rozhádzanie kúskov toho istého zvoleného obrázka”. Tlačidlo “**Nová hra**” zahodí starú hru a vráti užívateľa na začiatok programu pre nového výber obrázka.
V strede na hornom panely sa nachádza **informácia o aktuálnom priblížení** hracej plochy (zobrazí sa len pri hraní hry).
Kúsky v rámci hracej plochy **nemôžu presahovať horný ľavý roh hracej plochy** (z dôvodu implementácie, program kúsky nepustí za túto hranicu). Kúsky však **môžu byť umiestené** ľubovoľne ďaleko **smerom dole a napravo**, hracia plocha sa automaticky natiahne.

Programátorská časť

Program pozostáva z týchto hlavných tried: Form1.cs, PuzzlePiece.cs, Gameboard.cs, GridLayer.cs, PuzzleGameData.cs, PuzzleGameUtilities.cs, PictureEditor.cs, PieceArrangement.cs, PuzzleBezierovaKrivka.cs.

Pri implementácií hry som musel vyriešiť viaceré problémy:

1. Ako reprezentovať kúsok,
2. Ako vystrihnúť kúsky z obrázka,
3. Ako zistiť, či sa kurzor dotýka daného kúska,
4. Ako prekresľovať kúsky pri pohybe kúska,
5. Ako spojiť ľubovoľné dva kúsky,
6. Ako vytvárať skupinky s puzzle kúskami,
7. Ako meniť zobrazovaciu veľkosť a pozíciu hracej plochy.

Ako reprezentovať kúsok

Každý kúsok je reprezentovaný ako objekt, obsahujúci metódy, vlastnosti, a premenné.
Objekt kúsok (trieda PuzzlePiece.cs) obsahuje referenciu na **Bitmap s vystrihnutým kúskom** z obrázka, obsahuje svoje **súradnice**, **veľkosť** (veľkosť kúska bez zúbkov a so zúbkami), **oblasti** reagujúce na **pripojenie** sa ku kúsku, svojich pôvodných susedov (hranovo) a taktiež **rozloženie zúbkov.** Rozloženie zúbkov je vždy **vygenerované náhodne** pri každej novej hre.
(*trieda PieceArrangement.cs a metóda GeneratePiecesArrangement() v triede PuzzleGameUtilities()*).

Ako vystrihnúť kúsky z obrázka?

Prv som si obrázok rozstrihal na štvorčeky, kde každý štvorček bol vačší ako zvolený rozmer kúska. Je to kvôli tomu, že kúsky majú zúbky, ktoré môžu vychádzať smerom von z kúska, a tie chceme aby tiež obsahovali časť obrázka.
Následne som pomocou Bezierových kriviek vytvoril jedinú hranu so zúbkom (krivku). Túto hranu (krivku) je možné ľubovoľne otáčať a spájať s ostanými hranami(krivkami) kúska. Vytvorené štyri hrany (krivky) som spojil do jednej **uzavretej krivky reprezentujúcej tvar kúska**. Pomocou tejto krivky som vystrihol zo štvorčeka požadovaný tvar pomocou nastavenia **Clip** oblasti(všetko vo vnútri tejto oblasti ostane vykreslené). Inou metódou by bolo použitie takzvanej masky, kde by som porovnával zodpovedajúce pixely vo štvorčeku s pixelmi v maske. Avšak riešenie s použitím clip oblasti bolo jednoduchšie na implementáciu.

Ako zistiť, či sa kurzor dotýka daného kúska?

Najjednoduchším spôsobom by bolo prejsť každý kúsok v zozname kúskov, ale ak by tých kúskov bolo dostatočne mnoho, rádovo 500 - 2000, tak by bolo dobré si hraciu plochu virtuálne rozdeliť na takzvané "**buckets**" (**košíky**), kde v každom košíku by bola referencia na kúsky, ktoré sa nachádzaju v blízkosti košíka.
Následne pri hľadaní najbližších kúskov ku kurzoru myši by sme si vyrátali súradnice košíka, a prechádzali iba kúsky vrámci košíka.

Pri kliknutí myšou na určitú oblasť hracej plochy sa vyrátajú súradnice košíka, prejde sa zoznam kúskov v košíku a **vyberie sa kúsok**, ktorý je v rámci košíka **čo najvyššie**. Vybraný kúsok sa stane najvyšším a už **nie je členom košíka**, v ktorom pred kliknutím bol.

Ako vybrať kúsok z košíka? Ako som spomenul vyššie, každý kúsok má svoj tvar.
Aby sme zistili, či sa kurzor nachádza v danom tvare kúska, porovnáme **pixel Bitmap-u** kúska, ktorý sa nachádza priamo pod súradnicami kurzory. Ak je jeho **alfa** kanál rovný 0, daný pixel je **priehľadný** a tým pádom ukazujeme mimo kúsok, Ak je **alfa > 0** ukazujeme na pixel určitej farby a teda ukazujeme na kúsok. Na obrázku 5., sivá oblasť reprezentuje priehľadnú časť Bitmap-u. Červené ohraničenie obrázka naznačuje reálnu veľkosť vykresľovaného z časti priehľadného kúska, pomocou priehľadného Bitmap-u.

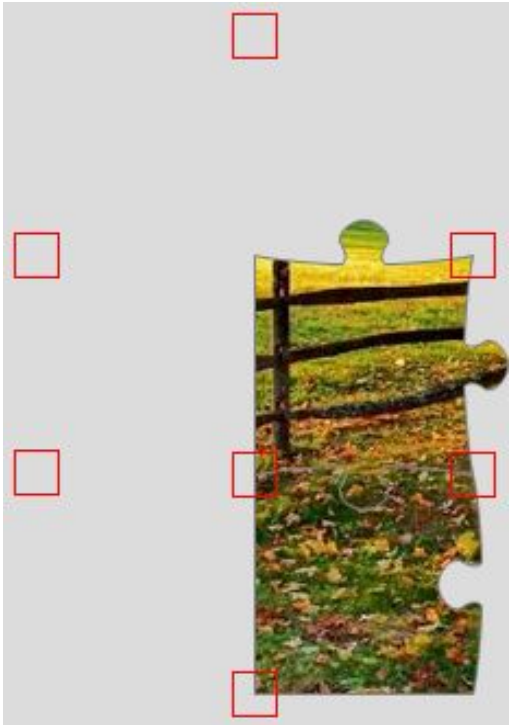
obrázok 5.

Ako prekreslovať kúsky pri pohybe kúska ?

Pri presúvaní kúska sa v skutočnosti prekresľuje **iba oblasť pod kúskom**. Obrázok 5. zároveň naznačuje oblasť, ktorá sa prekreslí. K prekresleniu dôjde iba pri zdvihnutí a následne až po položení kúska na miesto. Počas pohybu ostáva pozadie s kúskami ako jeden Bitmap a ponad neho pohybujeme vybraným kúskom (Bitmap-om). Počas pohybu teda prekresľujeme iba dva Bitmap-y: pozadie a vybraný kúsok. Aby som vytvoril čo **najplynulejší pohyb bez zasekávania** sa počas prekresľovania, použil som takzvanú **Buffer Graphics**, ktorá vykresľovanú scénu ukladá do dvojitej vyrovnávajcej pamäte a až následne, pomocou metody BufferGraphics.Render() vykreslíme scénu na obrazovku.
Oblasť pod kúskom prekresľujem tak, že si zhromaždím **všetky košíky s kúskami**, do ktorých **zasahujú súradnice štyroch rohov** daného kúska. Všetky kúsku z košíkov vykreslím a na ne nakoniec vykreslím aj samotný kúsok.
Následne z takto vytvoreného Bitmap-u vystrihnem oblasť iba oblasť najbližsiu kú kusku. Túto vystrihnutú oblasť vykreslím ponad hraciu plochu do presne stanovených súradníc. To vytvára efekt, že sa prekreslí iba oblasť, tesne pri kúsku. Užívateľ navonok nič nespozoruje.

Ako spojiť ľubovoľné dva kúsky ?

Pri spájaní dvoch kúskov som vytvoril štyri oblasti, v ktorých ak sa ocitne súradnica rohu kúska, tak sa daný kúsok pripojí ku kúsku. Každá zo štyroch oblastí reprezentuje hranu, ku ktorej to chceme pripojiť. (obrázok 6.)



obrázok 6.

Ako vytvárať skupinky s puzzle kúskov ?

Pri každom pripojení nového kúska k skupinke zistím, či sa spájané hrany zhodujú v tvare (teda či do seba zúbky zapadnú). (metóda SnapPiece v triede Gameboard.cs). Pri následnom uvoľnení ľavého tlačidla(event MouseUp) sa zavolá metóda SnapPiece, a ak bol kúsok v niektorej zo zacvakavacích oblastí, vytvorí sa efekt zacvaknutia posunutím suradníc kúska k zacvakavanému kúsku.

Ak chcem **presúvať skupinku kúskov:** Pri každom zdvihnutí kúska, navštívim aj všetkých podsusedov (ak existujú) kúska pomocou prehľadávanie do hĺbky. Rekurzívne sa vnáram do jednotlivých podsusedov, spýtam sa ich, či som ich už nenavštívil a ak nie tak daného suseda pridám do **zoznamu najdených susedov** (List<PuzzlePiece> Neighbours). Daného suseda vyhlásim za navštíveného, aby som ho pri dalšom rekurzívnom vnáraní znova nenavštivil a pozriem sa na jeho podsusedov. **Zoznam najdených susedov** použijem neskôr pri hromadnom pohybe kuskov (zmene suradníc každého kúsku v zozname) a hromadnom prekreslovaní a zacvakovaní kuskov.

Ako meniť zobrazovaciu veľkosť a pozíciu hracej plochy ?

Pre **zväčšenie / zmenšenie** som použil takzvaný “scaling factor”, teda double číslo reprezentujúce aktuálne zväčšenie / zmenšenie voči originálnej veľkosti. Dané číslo

používam pri prekresľovaní kúskov, prepočítaním rozmerov. Kúsok ako taký, fyzicky zostáva rovnaký. Má rovnaké súradnice rovnakú veľkosť Bitmap-u, len je vykreslený v inom rozmere pomocou Graphics.DrawImage(). Je samozrejme potrebné prerátať **reálne súradnice kurzora** na súradnice určene zmenou veľkosti, aby naďalej, kurzor správne ukazoval na kúsky, nad ktorými sa nachádza.
Hracia plocha **zoom-uje v bode**, kde je kurzor myši. To znamená, že potrebujeme hraciu plochu pri zmene veľkosti posunúť spať pod kurzor, ako tomu bolo pred zmenou veľkosti.

Pri **posune hracej plochy** meníme súradnice hracej plochy a samozrejme tento posun potrebujeme odčítať od pozície kurzora myši, aby kurzor správne ukazoval na jednotlivé kúsky.

Záver

Čo by som chcel v budúcnosti doimplementovať je, aby si užívateľ mohol overiť, či je puzzle zložené správne.