

Jevčák Vít

KABOOM!
Zápočtový program

2023

Obsah

Úvod.....	3
Návod.....	3
Ovládání.....	3
Klávesové zkratky.....	3
Životy.....	3
Trofeje.....	3
Menu.....	4
Programátorská dokumentace.....	4
Struktura.....	4
Přehled objektů.....	4
Přehled funkcí.....	6
Přehled eventů.....	7

Úvod

Při vytváření mého zápočtového programu jsem se nechal inspirovat videohrou Kaboom! která vznikla v roce 1981 pro herní konzoli Atari 2600. Byla vytvořená společností Activision a její design vytvořil Larry Kaplan. Kaboom! byla později spustitelná i na konzoli Atari 5200 a osmibitových počítačích Atari. Cílem hry je chytání bomb pomocí plošiny a získání co nejvyššího skóre.

Návod

Ovládání

Hráč ovládá pouze plošinu, která se může pohybovat pouze do stran a k pohybu se využívají šipky doleva a doprava.

Klávesové zkratky

- S nebo VolumeMute – v jakémkoliv stavu hry vypne/zapne všechny zvuky.
- Escape – chová se rozdílně podle toho kde se nacházíme
 - o Zmáčknutí klávesy Escape v hlavním menu aplikaci vypne, k vypnutí může být také využito tlačítko Zavřít.
 - o Zmáčknutí Escape ve hře hru pozastaví a opětovné zmáčknutí hru opět spustí.
 - o Zmáčknutí Escape při prohře nás vrátí zpět do hlavního menu.

Životy

- Na začátku má hráč podle zvolení obtížnosti 3 nebo 1 život. Hráč přichází o životy vždy když nezachytí padající bombu. Nezachycená padající bomba vyústí v ztrátu jednoho života a pozastavení hry.

Trofeje

- Hráč může dosáhnutím určitého skóre získat po řadě bronzovou, stříbrnou, zlatou, emeraldovou a diamantovou trofej, přičemž emeraldovou může získat pouze v případě, že hraje o druhé obtížnosti a

více, diamantovou pouze u nejtěžší (třetí) obtížnosti. Při získání trofeje obdrží hráč, v případě, že nemá plný počet životů, jeden život navíc.

Menu

- Při prvotním spuštění hry se hráč ocitne v hlavním menu, kde si může zvolit obtížnost, ztišit zvuk, začít hrát či hru ukončit.

Programátorská dokumentace

Struktura

- Hlavní tělo programu je ve složce *Form1.cs*, *Funkce.cs* obsahuje pomocné funkce, *Form1.Designer.cs* obsahuje automaticky generovaný kód a *bin/Debug/net6.0-windows/Resources* obsahuje zdrojové obrázky a zvuky.

Přehled objektů

- Všechny objekty, které se vykreslují, jsou typu *PictureBox/Button* a veškerou logiku za fungováním obstarávají funkce či události těchto objektů (nejsem s tím úplně spokojený, ale potřeboval jsem najít funkční řešení vykreslování, které se nesešlo a vystačilo mi na vše co potřebuji, což *PictureBox* splňoval). Objekty mám teoreticky rozděleny na 7 druhů – plošina, Bob, bomby, srdce, trofeje, reproduktor a zbytek (*Button*, *Label*, *Timer* a *NumericUpDown*)
- Plošina – *PictureBox Paddle*
S plošinou jako jedinou pohybuji pomocí kláves doleva a doprava, nechávám ji viditelnou pouze ve stavech HRA a PAUZA.
- Bob – *PictureBox Bob*
Bob je “mad bomber“, pohybuje se vždy za novým náhodně vygenerovaným bodem, k pohybu Boba mám vytvořený *Timer BobPosun* a vždy si pomocí funkce *SpoctiRychlostBoba* spočítám interval časovače *BobPosun* a Boba posunuji vždy směrem k náhodně vygenerovanému bodu, kde by měl pustit bombu. Posunuji ho o číslo *RychlostBoba*, a pokud se Bob nachází v okolí nového bodu tak Boba neposunuji. Bob je viditelný pouze ve stavu HRA a PAUZA.

- bomby – *List<PictureBox> Bomby*
Bomb mám omezený počet a udržuji si je v listu *Bomby*, přičemž všechny akce na bombách dělám pouze pro bomby, které jsou viditelné. Bomby dělám viditelné pomocí *Timeru bombovac*, vždy při tiknutí. Pro pohyb bomb pak mám *Timer timer1*, který pohybuje se všemi viditelnými bombami a vždy při tiknutí kontroluji, jestli jsem náhodou bombu nechytíl nebo nespadla vedle. Pokud spadne vedle, tak bombě nastavím obrázek exploze, ozvučím výbuch a nastavím pauzu. Díky omezenému počtu si při ztěžování hratelnosti musím kontrolovat aby se nestávalo, že bomba nespadne a znovu se objeví na počátku, proto pak při ztěžování měním jen rychlost padání bomb a interval mezi bombami není nikdy menší než 250ms. Bomby generuji vždy podle umístění Boba a ne podle náhodně generovaného bodu *NextBomb*, aby to co nejvíce vypadalo, že bomby shazuje Bob. Bomby jsou viditelné vždy pouze ve stavu HRA a PAUZA.
- Srdce - *List<PictureBox> Zivoty*
Počet životů je nastavení nedovoluje více než 3 životy. Životy zobrazuji vždy jen ve stavech HRA a PAUZA. Počet životů si udržuji v proměnné *zivoty* a zobrazuji vždy jen počet podle aktuálního stavu životů.
- Trofeje - *List<PictureBox> Trofeje*
Trofeje si udržuji v listu trofejí a vždy je zobrazím jen pokud má hráč určité skóre + při získání určitého skóre také přidám život navíc pokud je to možné.
- Reproduktor - *PictureBox zvoucek*
Reproduktor je obrázek, který zobrazuje pouze zda je zvuk zapnutý, či ne, přičemž vždy při kliknutí na reproduktor se přepne na opačný režim a ve všech částech, ve kterých se spouští zvuk se nejdříve podívám na *bool HrajeHudba*. Reproduktor je viditelný ve všech částech programu.
- Zbytek - *Buttony, Label, Timery a NumericUpDown*
Tlačítka využívám pro přepínání mezi herním stavem a vypnutí programu. *Label Skore* používám na zobrazení aktuálního skóre. Timery využívám na pohyb bomb (*timer1*), Boba (*BobPosun*) a na generování bomb (*bombovac*). *NumericUpDown (Obtiznost)* využívám na zvolení obtížnosti (1-3).

Přehled funkcí

- *EnableButton(Button button, bool OnOff)*
Dle *boolu OnOff* zviditelní či nechá zmizet tlačítko předané parametrem. Nevrací nic.
- *NastavStav(Stav NaJaky)*
Funkce nastaví stav na požadovaný stav předaný v parametru, zviditelní/zneviditelní objekty podle stavu (switch statement). Funkce vrací void.
- *HudbaZmena()*
Volaná z kliknutí reproduktoru, podle aktuálního stavu hudby změni na stav opačný. Nic nevrací.
- *KontrolaSkore()*
Funkce na odměňování, volána při zvyšování skóre, kontroluje jestli hráč dosáhl určitých met a podle nich zviditelňuje trofeje a volá funkci na přidání života. Nevrací nic.
- *JeVRozmezi(Point first, Point second, int range, int polovinaSire)*
Kontroluje, je-li střed bodu *first* v mezích objektu o umístění levého horního rohu *second* a šířky *range*. Vrací true pokud ano, false jinak.
- *HudboHrej(bool hraje, SoundPlayer zvukar)*
Z *boolu hraje* určí jestli zapnout *Soundplayer.PlayLooping()* či ne. Nevrací nic.
- *SpoctiRychlostBoba(int posunBoba, int celkovyCas, int pozice1, int pozice2)*
Z daných parametrů spočte interval, kterým se má Bob pohybovat při jeho zadané rychlosti aby se za určitý čas přesunul o vzdálenost $|pozice1 - pozice2|$. Vrací výsledek tohoto výpočtu, nebo 1 pokud se výsledek rovná 0 (což by nemělo nikdy nastat v aktuálním stavu programu, jelikož se to může stát jen pokud buď *posunBoba* či *celkovyCas* je roven 0, což se již nemůže stát, ale z možnosti chyby nechávám.
- *VratRozdilAbs(int prvni, int druhy)*
Vrací absolutní hodnotu rozdílu dvou čísel.
- *ZmenInterval(System.Windows.Forms.Timer casovac, int oKolik)*
Mění interval *Timeru casovac*, pokud by změnilo do záporných hodnot, pak interval zmenší na polovinu, limitně se interval ukotví na hodnotě 1, program počítá s tím, že v takové rychlosti to nemůže být lidsky možné.

- *ZmenaZivotu(int pocet)*
Vrací buď *pocet + 1* pokud by nebyl překročen maximální počet životů, jinak *pocet*.

Přehled eventů

- *Form1()*
Načte hlavní okno, načte ho na pozici (0,0), zviditelní to co má být zviditelněno, nastaví stav Menu, zapne hudbu.
- *ProcessCmdKey(ref Message msg, Keys keyData)*
Zpracovává zmáčkнутé klávesy, jmenovitě Escape, Left, Right, VolumeMute a S.
- *BReset_Click(object sender, EventArgs e)*
Nastaví stav Hra, nechá zmizet to co nepatří na začátek hry, vynuluje skóre, obnoví životy, zapne hudbu přes funkci *HudboHrej()*.
- *BHraci_Click(object sender, EventArgs e)*
Nastaví stav Hra, nastaví životy dle obtížnosti, plošinu nastaví na počáteční pozici, nechá zmizet trofeje.
- *BMenu_Click(object sender, EventArgs e)*
Nastaví stav Menu, nastaví skóre na 0, zapne hudbu *HudboHrej()* a nechá zmizet trofeje.
- *BPokracovat_Click(object sender, EventArgs e)*
Nastaví stav Hra a zapne hudbu.
- *timer1_Tick(object sender, EventArgs e)*
Všechny viditelné bomby posune a zkontroluje zda nejsou chyceny či nemají bouchnout. Při výbuchu nechá zaznít výbuch, ubere život a zkontroluje jestli nenastala prohra. Při kontrole chytání inkrementuje skóre a volá *KontroluSkore()*.
- *bombovac_Tick(object sender, EventArgs e)*
Nechá zviditelnit právě jednu bombu, její pozici nastaví dle pozice Boba, nechá vygenerovat bod *NextBomb*, zavolá funkci *SpoctiRychlostBoba()* a kontroluje, jestli se nebudou měnit intervaly mezi bombami či rychlost padání bomb (obtížnost hry)
- *pictureBox1_Click(object sender, EventArgs e)*
Zavolá funkci *HudbaZměna()*, která hudbu vypne, resp. zapne, dle aktuálního režimu hudby.
- *BobPosun_Tick(object sender, EventArgs e)*
Pokud se Bob nenachází v okolí bodu *NextBomb*, tak posune Boba směrem k tomu bodu.
- *Ukoncit_Click(object sender, EventArgs e)*
Ukončí aplikaci.