HANGMAN

Dokumentace

Duong Xuan Anh

Letní semestr 2020/2021

Programování II NPRG031

Obsah

[Zadání 3](#_Toc73042182)

[Komponenty aplikace 3](#_Toc73042183)

[Menu 3](#_Toc73042184)

[Hra 3](#_Toc73042185)

[Práce s programem 3](#_Toc73042186)

[Start programu 3](#_Toc73042187)

[Průběh hry 5](#_Toc73042188)

[Konec hry 6](#_Toc73042189)

[Programovací část 7](#_Toc73042190)

[Technologie 7](#_Toc73042191)

[Komponety programu 7](#_Toc73042192)

[Funkce jednotlivých komponentů 8](#_Toc73042193)

[Myšlenka, jak hru naprogramovat 8](#_Toc73042194)

[Průběh práce 10](#_Toc73042195)

[Závěr 10](#_Toc73042196)

# Zadání

Úkolem je vytvořit kompletní hru šibenice.

# Komponenty aplikace

## Menu

Okno s menu se bude spouštět jako první. Bude obsahovat položky „Nová hra“, „Nastavení“ a „Konec“.

* Tlačítkem „Nová hra“ se bude spouštět okno s hrou dle nastavených parametrů
* Tlačítkem „Nastavení“ se zobrazí nabídka k nastavení některých parametrů hry
  + Nastavení obtížnosti – nastavení obtížností na snadná, střední nebo náročná
* Tlačítkem „Konec“ se hra ukončí

## Hra

Okno s hrou bude obsahovat samotnou hru šibenice. Ovládání hry bude pomocí vytvořených tlačítek pro všechna písmena bez diakritiky. Při spuštění hry se vyhledá náhodné slovo ze souboru dle zadané obtížnosti.

Místo všech písmen hledaného slova se hráči zobrazí pouze pomlčky. Když hledané slovo bude obsahovat písmeno s diakritikou, doplní se písmeno ve správném tvaru, ale odhalí se tlačítkem písmena bez diakritiky.

Příklad: Slovo čolek obsahuje písmeno „č“, které se odhalí stiskem tlačítka „c“. Do odhaleného textu se však napíše „č“, nikoliv „c“.

Hráč se snaží trefit písmena, která jsou v hledaném slově. V případě, že se trefí, odhalí se všechny výskyty zvoleného písmene. V opačném případě se vykreslí část šibenice. Šibenici (konstrukci) je předkreslená, bude se na ni jen dokreslovat panáček (nejdříve hlava, pak tělo, jedna ruka, druhá ruka, jedna noha, druhá noha, čára přeš krk). V případě dokreslení celého panáčka a čára přeš krk hra končí.

Pak hráč si může dát novou hru nebo ukončit hru.

# Práce s programem

## Start programu

Při spuštění programu se objeví startovací menu.

Graphical user interface, application

Description automatically generated

Obrázek 1- Startovací menu

Program se může hned ukončit tlačítkem KONEC.

Při stisknutí tlačítka NASTAVENÍ se nám zobrazí nové okno se nastavením obtížnosti. Uživatel si může vybrat jedno ze tří možností (Snadné, střední, obtížné).

Graphical user interface, text, application

Description automatically generated

Obrázek 2- Okno nastavení obtížnosti

Když uživatel klikne na tlačítko NOVÁ HRA tak se mu zobrazí herní okno a bude hádat obtížnosti slova, který si nastavil (defaultně je snadná obtížnost).

Graphical user interface, application

Description automatically generated

Obrázek 3 - Hrací okno

## Průběh hry

Cílem hry je odhadnout skryté slovíčko pomocí jednotlivých písmen. Na každé písmeno lze kliknout pouze jednou. Pokud uživatel uhodne správně písmenko tak to písmenko se zobrazí na odpovídajícím místě ve slovu a nahradí symbol „**\_**“. Pokud hráč uhodne špatně tak se vykreslí část těla na levým obrázku.

Graphical user interface, application

Description automatically generated

Obrázek 4- Ukázka průběh hry

## Konec hry

Hra končí, pokud hráč odhadne správně slovíčko anebo náš panáček bude vykreslený celý. Po končení hry, zobrazí se nám slovíčko, které jsme museli hledat a nabídku hrát hru znovu neboli ukončit hru.

Graphical user interface, application

Description automatically generated

Obrázek 5 – Stav, když slovo nebylo odhadnuté

Graphical user interface, text, application, Word

Description automatically generated

Obrázek 6 - Stav, když hráč slovíčko odhadl

# Programovací část

## Technologie

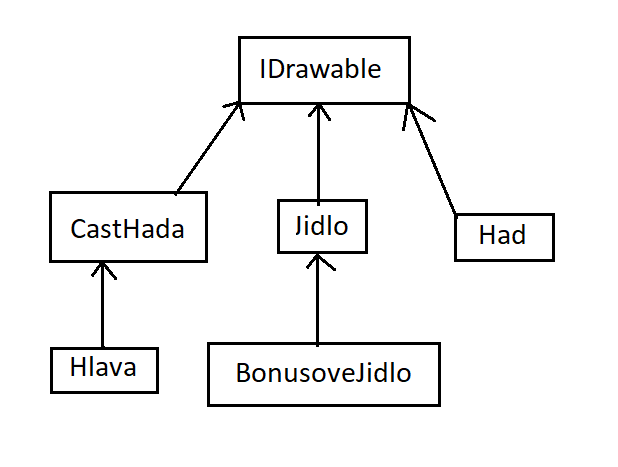
* C#: programovací jazyk.
* Window form: použený na design.

## Komponety programu

Kromě systémových věcí, program se skládá s následujících komponentů:

* IDrawable.cs
* CastHada.cs
* Had.cs
* Jidlo.cs
* Hlava.cs
* BonusoveJidlo.cs

Souvislost komponentů vysvětluje následující diagram dědičnosti.



Obrázek 7 - Diagram dědičnost tříd v programu

## Funkce jednotlivých komponentů

* **IDrawable.cs:** Tato třída je nadefinovaná jako interface a obsahuje pouze jednu funkci VykresliSe (Graphics g).  
  Interface se používá, aby třídy, které budou od ní dědit tak aby nezapomněli na tuto funkci.

Funkce VykresliSe() obsahuje parametr Graphics g, který nám dovolí kreslit na canvasu(plátno).

* **Had.cs:** Třída, která dědí od třídy **IDrawable.cs**. Vykresluje hada a kontroluje a dává nám vědět jeho všechny aktivity.
  + V konstruktoru se nám vytvoří had
  + JsiNaTetoPozici (int x, int y) zajistí aby jídlo se vygenerovalo na pozici kde se pohybuje had.
  + PosunSe (Jidlo jidlo) volá funkci, aby had se posouval. Zároveň hlídá kolizi mezi hlavou a jídlem, hlavou a části těla a okraje hracího plátna. Podle toho volá funkce jako jsou Jez (), Umri ().
  + Umri() dá programu vědět, že had umřel, pomocí události Action HadUmrel.
  + Jez () dá programu vědět, že had snědl jídlo, pomocí události Action HadSnedlJidlo.
  + ZvetsiSe () vytvoří novou část těla a dává mu stejný směr jako měla ta poslední část před ní.
  + ZmenSmer (Smer novySmer) mění směr hada
* **CastHada.cs:** řesí směr hada neboli směr všech částí toho hada a pohyb hada.
* **Hlava.cs:** Podstatě jefunguje jako část toho hada, a to je i důvodem proč dědí z třídy CastHada. Jediná věc, která se odlišuje je přepsání funkce VykreslitSe ().
* **Jidlo.cs:** třída pro vytváření a vykreslování jídla.
* **BonusoveJidlo.cs:** dědí z třídy Jidlo, jenom má pouze přepsanou funkci VykresliSe().
* **Form1.cs:** 
  + canvas\_Paint (object sender, PaintEventArgs e): Podstatě, všechny věci se doopravdy vykreslují v této funkci.
  + herniTimer\_Tick (object sender, EventArgs e): volá funkci, aby had se pohyboval a zárověn obnovuje canvas za každých 100ms.
  + bonusovyTimer\_Tick (object sender, EventArgs e): tento timer se zapne, když se objevý bonusové jídlo a má na starost odečítat hodnotu (value) progressBaru.
  + Ostatní funkce dělají přesně to jak se ta funkce jmenuje.

# Myšlenka, jak hru naprogramovat

Pomocí timeru hrací canvas se bude obnovovat za každých 100ms. Všechny části hada jsou uložený v Listu Jednotlivé části se budou pohybovat a části hada se budou pohybovat jako ty předchozí části. Pokud část toho hada či hlava toho hada bude na stejné pozice jako jídlo, tak to jídlo zmizí a přemístí se. Pokus jídlo se přemístí tam kde jsou části toho hada tak se vygeneruje znovu to jídlo, dokud nebude na validní pozici. Po snězení toho jídla, had se prodlouží o 1 čtvereček, který bude přidán na konec listu, kde jsou uložené části těla a dostane směr jako ta polední část, která tam byla. A had se bude hrát tak dlouho dokud neumře. Úmrtí nastane, když hlava hada bude na stejné pozici jako nějaká část z těla. Nebo když pozice hlavy bude mimo hracího canvasu.

# Průběh práce

Viz na githabu: <https://github.com/xuananh0/had>

# Závěr

Had byla moje velice oblíbená hra z dětství. Jsem velice rád, že jsem schopný si naprogramovat svého vlastního hada. Díky téhle hře jsem se naučil, jak funguje OOP a jak pracovat s WindowFormem.