

# Onono! The Puzzle Game

Šimon Růžička

ČVUT FIT

ruzicsi1@fit.cvut.cz

2. ledna 2023

Tento dokument je průvodní zprávou k semestrální práci do předmětu Programování v Pythonu[1] (BI-PYT) na FIT ČVUT, vytvořené v semestru B221. Při tvorbě semestrální práce jsem čerpal pouze z níže uvedených zdrojů.

## 1 Úvod

Semestrální práce implementuje známou *puzzle* hru Nonogram [2], spočívající v obarvování bílé mřížky na základě zadaných instrukcí. Hra je implementována za pomoci knihovny `pygame` [3], která se stará o veškeré vykreslování GUI.

Jedním z hlavních rozšíření této hry je možnost nahrát libovolný obrázek formátu PNG a nechat jej převést na samotnou hádanku. To umožní ve hře rychlé a jednoduché vytváření hádanek a velkou míru znovuhratelnosti. Ke zpracování vstupního obrázku se využívá knihovna PIL (tj. *Python Imaging Library*).

## 2 Moduly

Práce je členěna na několik Python modulů, které se starají o jednotlivé funkcionality hry. Pro spuštění hry lze využít modul `__main__` spouštěný přímo z příkazové řádky, je ale obsažen i modul `__init__`, pomocí kterého lze balíček importovat v interaktivním shellu/jiném skriptu a používat jednotlivé komponenty (vhodné především pro debug hry).

O veškeré vykreslování prostřednictvím `pygame` se starají dva moduly, `menu` a `app` pro samotnou herní obrazovku. Ty využívají společné konstanty (barevnou paletu, velikost obrazovky, ...) ze souboru `gui_definitions`.

Základní funkce hry se nachází v modulu `savegame`, který obsahuje třídu `Savegame`. Ta se používá jak pro import/export hádanky, tak i pro její reprezentaci za běhu hry, tvoří tak klíčovou součást kódu. Další funkce z běžného chodu hry jsou v modulu `gamelogic`. Speciálně oddělená je pak funkcionality pro načtení hádanky z obrázku v modulu `image`.

Zcela odděleně jsou pak *testovací moduly* v adresáři `tests`.

## 3 Reprezentace dat

Hra ukládá jednotlivé hádanky v podobě textového souboru obsahujícího údaje o vyplnění jednotlivých políček. K interakci s textovým souborem se používají funkce `numpy.loadtxt()` a `numpy.savetxt()`.

Po načtení do paměti bude hádanka reprezentována pomocí několika dvourozměrných `numpy.ndarray`, fakticky *matic*. Jedna matice bude ukládat načtená data, představující korektní řešení, do druhé se bude ukládat průběžný stav hry. Nápoředy pro jednotlivé řádky/sloupce (spolu s informací o tom, zda je daný řádek kompletně vyplněný) se však budou ukládat jako `list`, vzhledem k jejich rozdílné délce.

## 4 Řešitelnost hádanky

Problém řešení Nonogramu již byl matematicky popsán jako *NP-úplný* [4], není znám žádný algoritmus řešící problém v polynomiálním čase. V semestrální práci jsem se tak snažil problému algoritmického řešení Nonogramu vyhnout (tj. přenechat na uživateli), z čehož plynou i určitá herní omezení.

Hra se snaží kontrolovat uživatelské tahy a posoudit, zda neprovádí neplatné tahy (ideálně co nejdříve). Zde však dochází pouze k jednoduché kontrole po řádcích/sloupcích, která není stoprocentní. Jakmile je řádek považován za kompletní (obsahuje skupiny vybarvených bodů správných velikostí), herní GUI tuto skutečnost hráči oznámí - to však není záruka toho, že tento tah vede ke správnému řešení.

Jako jedno z možných řešení tohoto problému se jeví ukládání hádanek v podobě vyplněné tabulky namísto zadaných čísel, a následné porovnávání uživatelského řešení s referencí. Tím by zdánlivě odpadly starosti s výpočetně složitějším ověřováním hráčských tahů, je však špatný nápad se spoléhat pouze na tento způsob. Některé hlavolamy totiž mohou být zadané nejednoznačně (tedy mít více než jedno korektní řešení). Tato cesta ověřování tahů by znamenala, že by hra neumožnila provést zcela korektní řešení, což je pochopitelně nežádoucí. Problém ověřování, zda je hádanka jednoznačná či nikoliv, je obdobně výpočetně náročný jako samotné

řešení hádanky, proto nebude v semestrální práci obsažen.

## 5 Metody/postupy/algorithmy

### 5.1 Ověřování řešení hádanky

Metody/postupy/algorithmy

1. Vytvoří se kopie tipů uživatele, která má místo prázdných polí tipy, že pole nemá být vybarveno,
2. dojde k porovnání totožnosti s referenčním řešením.
  - Pokud se matice shodují, dále se nic nekontroluje a hra je ukončena.
3. Pokud se neshodují obě matice, dojde k porovnání shody s jednotlivými vektory nápověd,
  - v případě shody všech nápověd je hra také považována za ukončenou.

Poslední krok (porovnání shody s nápovědami) se také dá využít pro určitou formu nápovědy hráči. Pokud se řešení v jednom řádku/sloupci s nápovědou shoduje, vykreslená čísla změni barvu. Tím uživatel dostane najevo, že tento řádek má "splněný", a může se soustředit na zbytek hrací plochy.

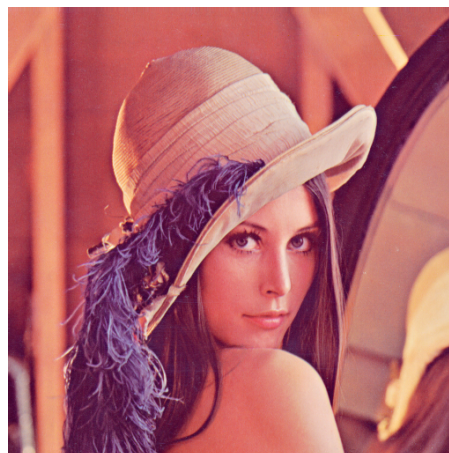
### 5.2 Konverze obrázku

Hra umožňuje vzít libovolný obrázek formátu PNG a převést jej na hratelnou hádanku. U toho je možné zvolit libovolnou velikost hrací plochy (menší než rozlišení obrázku) a *obtížnost hádanky*, tj. podíl vyplněných a nevyplněných polí. Algoritmus, který tento proces provádí, funguje následovně:

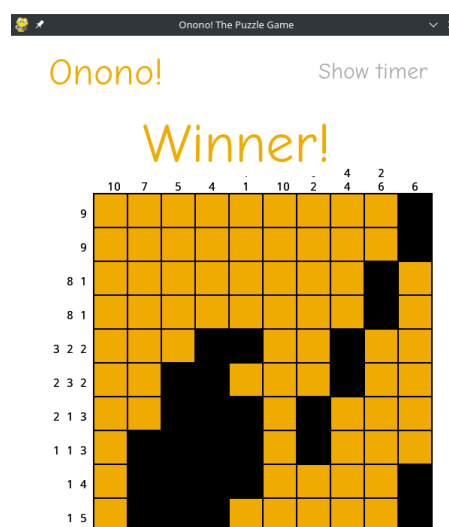
1. Zpracuje relativní cestu k souboru a otevře jej pomocí PIL,
2. převede obrázek na černobílý (tj. 256 různých barev),
3. přeškáluje obrázek na požadovanou velikost.
  - Pokud obrázek není čtvercový, vyřízne z něj pouze střed.
4. Následně se obrázek převede na `numpy.ndarray`,
5. aplikuje se hranice/*threshold* intenzity při převodu na dvojbarevný obrázek.

Právě aplikace thresholdu je nejzrádnější část – ten se musí pro jednotlivé obrázky lišit. Program tak iteruje přes různé thresholdy, aby se výsledný obrázek nejlépe přiblížil k požadovanému poměru

vybarvených polí. Zde se opíráme o informaci, že zvýšením thresholdu nikdy nemůže vybarvených polí přibýt (počet zůstane stejný, nebo klesne) a naopak.



Obrázek 1: Modelka Lenna [5]



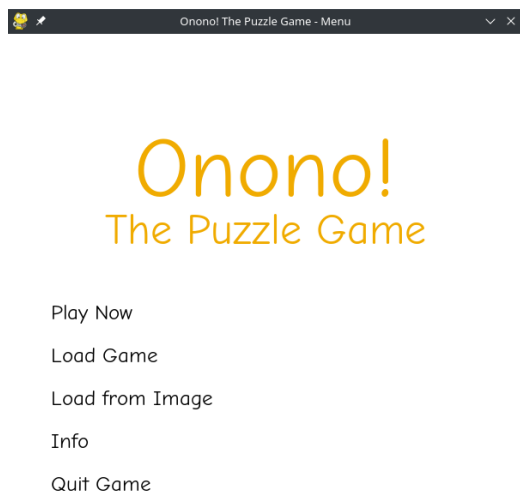
Obrázek 2: Lenna převedená na hádanku

## 6 Ukázka aplikace

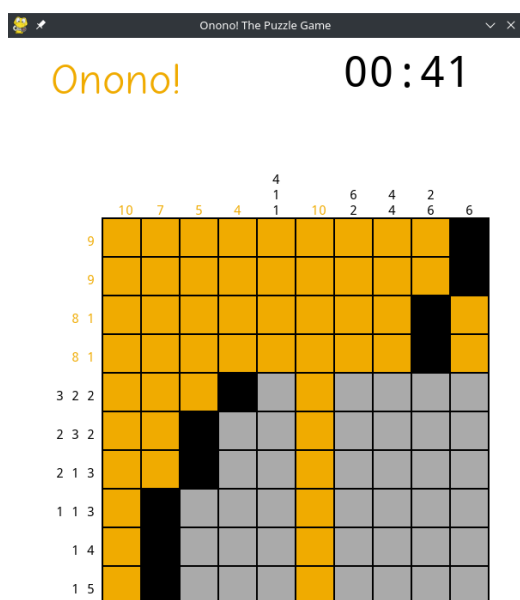
Po spuštění hry se uživateli objeví *menu 3* – umožní hráči vygenerovat novou (náhodnou) hádanku nebo ji načíst ze souboru (a to buď textového s korektním řešením, nebo obrázku).

Následně zvolíme jednu hádanku a na obrazovce se objeví herní plocha, na které pomocí myši vybarvujeme jednotlivá pole. Na obrázku 4 můžeme vidět, že po kompletním vyplnění řádku/sloupce dostáváme dříve zmíněnou nápovědu v podobě zbarvených čísel/nápověd. V pravém horním rohu se nachází časomíra, kterou lze pomocí kliknutí vypnout.

Po vyplnění celého hracího pole se hra vrátí zpět do menu, kde je možné hru ukončit nebo hrát dále.



Obrázek 3: Herní menu



Obrázek 4: Nedořešená hádanka

## 7 Závěr

Výsledkem práce je tedy určitý prototyp počítačové hry, který sice nemá jednotlivé požadované vlastnosti pro to, aby udržel zájem hráče na delší dobu, ale jedná se o fungující základní koncept, který je dále možné rozšiřovat. Osobně jsem s výsledkem spokojený, vzhledem k vhodně zvolenému tématu a používaným knihovnám šla práce rychle od ruky.

### 7.1 Možná rozšíření

Jak již vyplynulo z dosavadního textu, hru by šlo rozšířit o algoritmus pro řešení hádanek (viz Řešitelnost hádanky 4). Tím by bylo možné mimo jiné hráči poskytovat živou zpětnou vazbu při řešení

hádanek na jeho tahy, případně posuzovat jednoznačnost řešení.

Bylo by možné rozšířit hru na její iteraci s více barvami [6], obzvlášť v kombinaci s konverzí obrázků by toto rozšíření mohlo vypadat působivě.

Z rozšíření techničtějšího rázu by pak bylo možné implementovat *profil hráče*, který by ukládal nejlepší časy na jednotlivé standardizované hádanky. Tyto časy by pak bylo možné synchronizovat v online sdílené tabulce nejrychlejších časů, což by ještě podpořilo lidskou soutěživost.

Technické možnosti hry umožňují různé vlastnosti, u kterých bohužel nezbyl prostor na jejich implementaci do herního rozhraní s hráčem. Hráč by si před spuštěním hádanky při náhodném generování/načítání z obrázku mohl zvolit její velikost (v rámci rozumného rozmezí) nebo obtížnost (poměr vyplněných polí). K tomu by bylo potřeba implementovat další obrazovku menu (pravděpodobně celý jeden modul, i když by k tomuto účelu šlo přiohnout současné menu), ve které by si hráč mohl zvolit výše zmíněné nastavení.

### 7.2 Podpora zařízení

Knihovna `pygame` podporuje řadu různých uživatelských vstupů – v tuto chvíli hra přijímá víceméně pouze vstup prostřednictvím myši. Bylo by možné vytvořit podporu pro dotykové displeje, na kterých by se hra také intuitivně dobře ovládala.

Vzhledem k tomu, že není velké množství uživatelů s desktopovým operačním systémem (Linux, Windows) a dotykovým displejem, tato změna sama o sobě by příliš nového nepřinesla. Knihovna `pygame` však umožňuje také *zabalit* hotovou aplikaci pro operační systém Android, používaný na valné většině mobilních zařízení.

## Reference

- [1] Python Software Foundation. Python 3.11.1 documentation. <https://docs.python.org/3.11/>, 2022. [Online; accessed 27-December-2022].
- [2] Wikipedia. Nonogram — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Nonogram&oldid=1125785460>, 2022. [Online; accessed 27-December-2022].
- [3] Pete et al. Shinnars. Pygame 2.1.4 documentation. <https://www.pygame.org/docs/>, 2021. [Online; accessed 27-December-2022].
- [4] Jan N. van Rijn. Playing games: The complexity of klondike, mahjong, nonograms and

animal chess. <https://liacs.leidenuniv.nl/assets/2012-01JanvanRijn.pdf>, 2012. [PDF Online; accessed 27-December-2022].

- [5] Wikipedia. File:Lenna (test image).png — Wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=File%3ALenna%20\(test%20image\).png&oldid=1055437767](http://en.wikipedia.org/w/index.php?title=File%3ALenna%20(test%20image).png&oldid=1055437767), 2022. [Online; accessed 28-December-2022].
- [6] Chugunyy K.A. (KyberPrizrak). Colour japanese crosswords – nonograms.org. <https://www.nonograms.org/nonograms2>, 2016. [Online; accessed 27-December-2022].