

Hra v C#

Vsuvka než začneme: Jak zapojit GitLab:

1. **Rozbalte** (POZOR: opravdu rozbalit, ne Otevřít!) **cshra.zip** do složky na lokálním disku (NE na síťovém disku!)
2. Otevřete si svůj projekt pod <https://gitlab.mff.cuni.cz/teaching/nprg031/2022-summer>
3. Zkopírujte si **URL** vašeho projektu (**Ctrl-L Ctrl-C**)
4. Otevřete soubor **CSHra.csproj** ve VisualStudiu
5. Menu **File / Add to Source Control**
NEBO (pokud není ^^^^)
Git / Vytvořit úložiště Git...
6. V okně **Create a Git repository** zvolte **Existing remote**
NEBO (pokud „NEBO“ výše)
Jiné / Ukončuje se vzdálené úložiště (hodně zvláštní překlad!)
...a do **Remote URL** („Vzdálená adresa URL“) vložte URL vašeho projektu
7. Klikněte na **Create and Push**
8. ...vyskočí okno **Git Credential Manager**
9. V GitLabu Zvolte **Settings / Access Tokens**, vyplňte **Token name** (třeba **cshra**), nastavte mu **Expiration date** na zítřejší datum
NEBO ho na konci nezapomeňte zrušit,
klikněte na **Create project access token** a zkopírujte si výsledný token
(vypadá třeba takhle: **glpat-Ys62TXLPMtampEc6PC4i**).
10. Vytvořený token vložte i jako login i jako heslo do **Git Credential Manager**.
11. Pokud vám Push selhal (podívete se do okna **Output**, přepněte ho na **Source Control – Git**), tak si v okně **View / Git Changes** místo **master** zvolte **New branch**, pojmenujte si ji třeba **hra** – a zkuste znovu **Push**.
12. V GitLabu **Repository / Branches** byste měli vidět vytvořenou větev a uložený projekt.
13. Nové verze commitujte a odesílejte v okně **Git Changes** pomocí **Commit** a **Push**.

Jak vymazat uložené přihlašovací údaje:

Ovládací panely / Uživatelské účty /
/ Spravovat přihlašovací údaje / Přihlašovací údaje systému Windows
/ Obecné přihlašovací údaje

Zdrojové soubory

- **Hra.cs** logika hry
- **Form.cs** okno
- **plan.txt** mapa jedné úrovně hry
- **ikonky.png** ikonky jednotlivých prvků hry

Třída Mapa

Obsahuje údaje o stavu hry:

- **char [x,y] plan** obsahuje údaje o tom, co na kterém políčku je:
 - mezer – volné místo
 - X.....zed'
 - h.....hlína
 - B.....balvan
 - D.....diamant
 - H.....hrdina
 - < ^ > v.....příšera v některém ze čtyř směrů
 - E.....východ
- **Hrdina** je objekt hrdiny, ovládaný hráčem
- **PohyblivéPrvkyKromeHrdiny** seznam (List) Pohyblivých objektů, **neobsahuje** hrdinu!
- **stisknutaSipka** – údaj, jestli je stisknutá nějaká šipka

Hlavní herní „smyčka“

Časovač pravidelně volá

- **mapa.PohniVsemiPrvky(stisknutaSipka);**
pohyb všech pohyblivých prvků, pohyb hrdiny
- **mapa.VykresliSe(g, sirka_okna, vyska_okna);**
vykreslení

Chování jednotlivých pohyblivých prvků

...obstarává jejich funkce **UdelejKrok()**, pro každý prvek jinak.

Plán hry

Je uložen v textovém souboru **plan.txt**, při načtení se přepíše do pole **plan** a pokud načítaný prvek odpovídá některému z pohyblivých objektů, vytvoří se odpovídající objekt a přidá se do seznamu **PohyblivéPrvkyKromeHrdiny**, nebo do proměnné **Hrdina**. Všechny tyto proměnné patří objektu **mapa** třídy **Mapa**.

Vykreslování

Funguje tak, že se

1. podle pozice **Hrdiny** určí souřadnice výřezu, který se bude kreslit (mapa může být větší než zobrazovaná plocha).
2. pro každý prvek se podle znaku v **plánu** zjistí odpovídající ikonka a nakreslí se.

Ikonky

jsou uloženy v jediném souboru **ikonky.png**, vedle sebe.

Výška tohoto obrázku určuje šířku jednoho čtverečku (jsou to čtverečky) a z ní a z velikosti obrázku se spočítá počet ikonek. Funkce **NactiIkonky** je načte a uloží do pole **ikonky**.

Úkoly

Pohyb Hrdiny 1

Naprogramujte tělo funkce `Hrdina.UdelejKrok()`.

Až se vám to podaří, hlasitě vykřikněte „CHODÍ!“ a commitněte do GitLabu verzi „CHODÍ!“.

Nápověda:

Pro nové souřadnice Hrdiny jsou připraveny proměnné `nove_x` a `nove_y`, na začátku funkce je do nich uložena aktuální poloha – takže když je nezměníte, tak se Hrdina nebude hýbat (a naopak k pohybu hrdiny stačí změnit hodnoty těchto proměnných).

Bude se vám hodit proměnná `stisknutaSipka`.

Funkce `Hrdina.UdelejKrok()` už teď dělá to, že se pomocí funkce `mapa.JeVolnoNeboHlina(int x, int y)` podívá, jestli je na nové pozici volno nebo hlína (kterou hrdina umí prokopat) a když ano, tak zavolá funkci

`mapa.Presun`, která přesune obsah plánu (pole) a pokud na původním políčku byl nějaký pohyblivý objekt (včetně Hrdiny), tak mu navíc změní souřadnice `x` a `y` (to je důležité!).

Pohyb Balvanu 1

Naprogramujte tělo funkce `Balvan.UdelejKrok()` tak, aby balvan padal dolů.

Až se vám to podaří, hlasitě vykřikněte „PADÁ!“ a commitněte do GitLabu verzi „PADÁ!“.

Nápověda:

Bude se vám hodit funkce `mapa.Presun` a funkce `mapa.JeVolno(int x, int y)`, která vrací `true`, pokud je políčko pod balvanem prázdné. Pokud na políčku je pohyblivý objekt, funkce `mapa.Presun` změní jeho údaje `x` a `y` (tj. neměňte je, změní je funkce `mapa.Presun`).

Pohyb Hrdiny 2

Upravte tělo funkce `Hrdina.UdelejKrok()`, tak, aby hrdina při pohybu doleva nebo doprava tlačil balvan, pokud je za ním (balvanem) volné místo.

Až se vám to podaří, hlasitě vykřikněte „TLAČÍ!“ a commitněte do GitLabu verzi „TLAČÍ!“.

Nápověda:

Na přesunutí balvanu použijte také funkci `mapa.Presun`, která (už to píšu potřetí) dělá i to, že se podívá, jestli na přesouvaném políčku není nějaký PohyblivýObjekt, jehož souřadnice by bylo potřeba změnit.

Pohyb Balvanu 2

Upravte tělo funkce `Balvan.UdelejKrok()`, tak, aby poznal, že po přesunu (**pozor, jedině po tom, co spadne!**) dopadl na Hrdinu.

V takovém případě hra končí vítězstvím Balvanu. Mapa má proměnnou `stav`, dosadte do ní správnou hodnotu, aby hra skončila.

Až se vám to podaří, hlasitě vykřikněte „**PROHRA!**“ a commitněte do GitLabu verzi „**PROHRA!**“.

Pohyb Diamantu

Stejně jako Balvan, Diamant by měl také padat dolů, pokud je pod ním volno.

Stejně jako Balvan, i Diamant dopadem na Hrdinu ukončí hru.

Až se vám to podaří, hlasitě vykřikněte „**DIAMANT!**“ a commitněte do GitLabu verzi „**DIAMANT!**“.

Diamant a sebrání diamantu

Když Hrdina narazí do Diamantu, diamant se zničí, hrdina vstoupí na jeho místo, zmenší se počet zbývajících diamantů a pokud už je nula, přepíšu se v plánu zavřené dveře („E“) na otevřené dveře („e“) (na to `mapa` už má funkci `mapa.OtevriVychod`).

Takovou komplikovanou akci by ale neměl řešit Hrdina, proto přidejte do mapy metodu

`SeberDiamant(int x, int y)` a Hrdina ji jenom zavolá.

Až se vám to podaří, hlasitě vykřikněte „**SBÍRÁ!**“ a commitněte do GitLabu verzi „**SBÍRÁ!**“.

Průchod východem

Zavřenými dveřmi hrdina projít nemůže, otevřenými ano a v tom případě

(`mapa.JeOtevrenyVychod(x, y) == true`) buďto hra končí vítězstvím Hrdiny nebo pokračuje další level (jen načtení další mapy). Nastavte stav hry.

Až se vám to podaří, hlasitě vykřikněte „**VÝHRA!**“ a commitněte do GitLabu verzi „**VÝHRA!**“.

Pohyb příšery

Naprogramujte tělo funkce `Prisera.UdelejKrok()`.

Příšera se v každém kroku posune dopředu nebo otočí doprava nebo otočí doleva a to tak, aby obcházela zeď, kterou má po pravé ruce.

Až se vám to podaří, hlasitě vykřikněte „**PŘÍŠERA!**“ a commitněte do GitLabu verzi „**PŘÍŠERA!**“.

Nápověda (pokud už nemáte příšeru vyrobenou z domácího úkolu):

- Nejdříve si rozmyslete algoritmus (s tužkou a papírem).
- Mohlo by vám pomoci, když si ve třídě `Prisera` vyrobíte funkce `JeZedNapravo`, `JeZedPredeMnou`... apod.

Bonus: Balvan a příšera

Pokud padající balvan spadne na příšeru, příšera exploduje a všechna volná políčka nebo hlínu ve čtverci 3x3 se středem na pozici (už bývalé) příšery promění na diamanty.

Až se vám to podaří, hlasitě vykřikněte „**POKLAD!**“ a commitněte do GitLabu verzi „**POKLAD!**“.

Volitelně: Pohyb Balvanu 3

Upravte tělo funkce `Balvan.UdelejKrok()`, tak, aby v případě, že pod ním není volno, ale je pod ním balvan a napravo nebo nalevo je volno, padal šikmo. Pokud je volno na obou stranách, vybere si opak toho, co si vybíral naposledy.

Až se vám to podaří, hlasitě vykřikněte „**CIKCAK!**“ a commitněte do GitLabu verzi „**CIKCAK!**“.

Bonus: Zvuky

Při kroku Hrdiny, při dopadu kamene, sebrání diamantu nebo explozi příšery přehrajte zvuk.

Až se vám to podaří, hlasitě vykřikněte „**ZVUK!**“ a commitněte do GitLabu verzi „**ZVUK!**“.

Bonus: Další úrovně

Nakreslete si plán dalších úrovní a při výhře v jedné úrovni načtete a spustíte další (třeba tak, že při načítání K-té úrovně nejdříve K-1 úrovní přeskočíte).

Až se vám to podaří, hlasitě vykřikněte „**LEVEL!**“ a commitněte do GitLabu verzi „**LEVEL!**“.