



Bakalářská práce

Modování Minecraftu

Studijní program:

Studijní obor:

Autor práce:

Vedoucí práce:

B0613A140005 – Informační technologie

Aplikovaná informatika

Jiří Růta

Ing. Jana Vitvarová

Liberec 2024

Tento list nahradte
originálem zadání.

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

30. 3. 2024

Jiří Růta

1 Minecraft

2 Minecarft z programátorské stránky

2.1 Blok

Každý druh bloku má svůj objekt v registru. Tento objekt obsahuje proměnné a metody specifické pro daný druh bloku. Tento objekt je třídy *Block* anebo z ní dědí. Jelikož je společný pro všechny bloky daného druhu, pokud chceme ukládat data specifická pro jeden block musíme použít *blockstate* anebo *tile entity*.

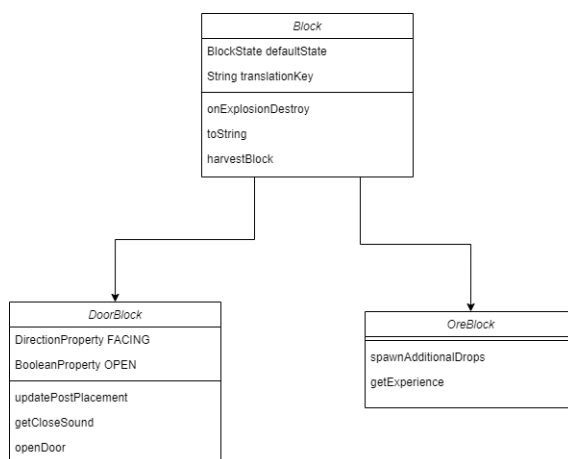


Figure 2.1: dědičnost třídy Block

2.2 blockstate

Blockstate je neměnná třída obsahující základní datové typy. Každý druh bloku v něm skladuje jiné proměnné. Pro každou kombinaci proměnných je vygenerován singleton. Každý blok má v jednu dobu pouze jeden blockstate. Jelikož je blockstate generován pro každou možnou kombinaci proměnných, nedoporučuje se to používat pro velké množství proměnných.

2.3 tile entities

Pokud v bloku potřebujeme skladovat velké množství proměnných, můžeme využít *tile entity*. Objekt třídy *tile entity* je samostatný pro každý blok.

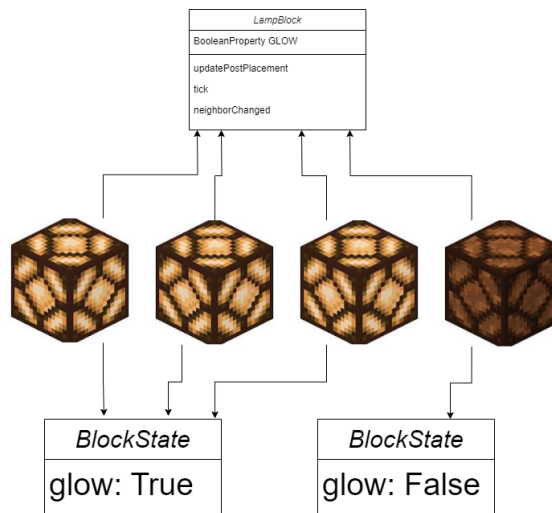


Figure 2.2: :blockstate

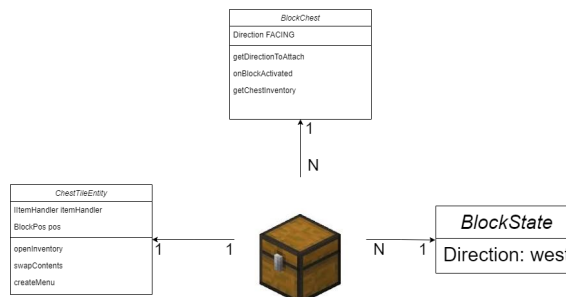


Figure 2.3: příklad vztahů

2.4 Inventář

Inventář v minecraftu je rozdělen na políčka. Každé políčko může obsahovat pouze jeden druh předmětu a pouze omezené množství. V programu je každé políčko reprezentováno objektem *itemstack*.

2.4.1 Itemstack

Třída obsahuje 2 důležité proměnné, jaký předmět obsahuje (odkaz na jeho třídu) a jeho množství. Třída obsahuje převážně metody na porovnávání objektů Itemstack mezi sebou, předávání informací o předmětu, ukládání a načítání dat.

2.4.2 Item

Každý druh předmětu má svůj singleton, který dědí třídu *Item*. Tato třída obsahuje proměnné a metody pro daný předmět. Například obsahuje metodu "onUse", která je volána pokud se hráč pokusí použít předmět. Jelikož je tato třída společná pro všechny itemy pokud chceme ukládat data specifická pro jeden předmět musíme

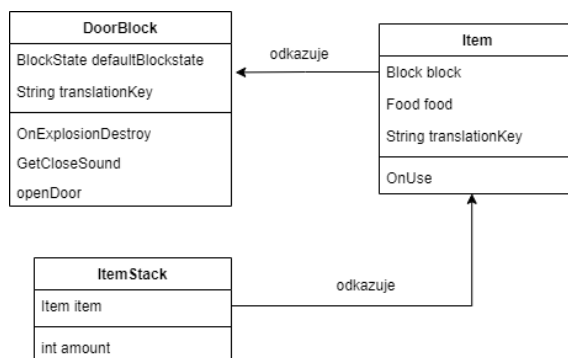


Figure 2.4: ItemStack vztahy

použít NBT.

2.5 UI

Některé bloky ,předměty otevřou hráči UI. Například když hráč klikne na pec, otevře se mu UI ve kterém může odložit předměty do pece, aby je pec předělala na jiné. Ui se v Minecarftu skládá ze dvou částí: Container a Screen.

2.5.1 Container

Container obstarává logicku a data. Komunikuje s ostatními částmi hry. Běží na serveru i na klientu.

2.5.2 Screen

Určuje jak bude UI vypadat. Zobrazuje text a obrázky. Komunikuje pouze s Container. Může obsahovat logicku, ale měla by být jen okolo grafiky v UI. Běží pouze na klientu.

2.6 Výroba

Velká část minecarftu je vytváření nových předmětů z těch které už hráč vlastní. Z čeho se předměty vyrábějí je zapsáno v JSON souborech.

2.6.1 Recept

Ve hře je několik druhů receptů, každý druh receptu má svojí třídu implementující interface IRecipe. Každý JSON soubor ve složce s recepty má hodnotu "type", která říká kterou třídou ho má hra přečíst. Při zapnutí se načtou veškeré recepty z JSON souborů a pro každý json soubor se vytvoří jeden objekt třídy implementující IRecipe. Tyto objekty obsahují veškerá data z JSON souborů a implementují

metody na porovnání zdali je možné recept vyrobit. Práce těchto tříd není vyrobit daný předmět, ale pouze říct zdali je možné daný předmět vyrobit a předat vyžádané informace.

2.7 Client a Server

Minecraft kód vždy běží dvakrát, jednou jako server a jednou jako client. I když hráč hraje sám na vlastním počítači, minecraft pořád běží rozdělen na stranu serveru a klienta. Na obou stranách běží stejný kód. Na rozlišení se používá metoda "world.isRemote", která vrací true pokud kód běží na klientu. Některé operace musí běžet pouze na jedné straně a druhou stranu upozornit pomocí paketů. Například vytváření nových předmětů by se musí provádět na serveru, jinak je možné že server nebude o předmětu vědět a když se ho hráč pokusí použít tak nebude fungovat, protože o něm server neví. Na druhou stranu, vykreslování grafiky se musí provádět pouze na klientu.

3 Můj mód z pohledu hráče

4 Můj mód z pohledu programátora