

## Ajtáci a hackeři

*Každý kód je jako stavba, která potřebuje pevné základy, aby se nezhroutila. Ale při tom musí zůstat štíhlá, aby byla rychlá. Úkol zněl jasně: Musíme vytvořit jednoduchý program, který na pokladně umožní načíst novou věrnostní kartičku. Jenže to by se o tom nesměli doslechnout hackeři, kteří se nám snaží pokaždé naše dílo narušit.*

V naší nové hře si mezi sebe účastníci rozdělí role programátorů, kteří se snaží vytvořit stabilní a kvalitní program, a hackera, který jim toto snažení kazí.

<b>Počet hráčů</b>	3-6
<b>Věk</b>	od 10 let
<b>Herní doba</b>	15 minut

### Obsah balení

- 50 dřevěných kostiček různých barev a tvarů
- 40 poznámkových kartiček
- když Vám kartičky dojdou, můžete si je vytisknout z odkazu <https://www.programator-shop.cz/produkt/hra-ajtaci-a-hackeri/>
- dřevěná základna
- pravidla

### Příprava hry

Účastníci partie vezmou tolik *poznámkových kartiček*, kolik je hráčů. Na jedné zakroužkují roli *hackera* a na ostatních roli *programátora*. Lístky vezmou a zamíchají. Každý si potají vezme jednu a podívá se na svoji roli. Přeloží část se svou rolí podél dvojité čáry tak, aby ostatní nemohli jeho roli přečíst a on si mohl při tom zapisovat pozice kostek. Každý hráč si vezme tužku.

Tím hráči určili své role, jeden bude hrát *hackera*. Všichni ostatní se stanou *programátory*. *Hacker* se bude snažit potají propašovat do programu chyby tak, aby jej co nejvíce destabilizoval a *programátoři* na to nepřišli.

Do středu stolu hráči položí dřevěnou základnu tak, aby na ni všichni pohodlně dosáhli. Poblíž umístí společnou zásobu padesáti dřevěných kostiček. Z nich vylosují dvanáct náhodných kostiček, které rozloží do společné nabídky. Začíná hráč, který naposledy napsal nějaký program. Nebo ho můžete vybrat náhodně.

### První fáze partie: Příprava opěrných bodů programu

Začínající hráč si v první fázi vybere libovolnou z nabídky 12 připravených kostek a položí ji kamkoliv na společnou základnu.

Od té chvíle se hráči střídají na tahu, ve kterém si vždy mohou vybrat sami (nebo po domluvě s ostatními) z připravené nabídky barevných kostek a přiložit ji do stavby tak, aby se dotýkala alespoň jedné již položené

kostičky. Při tom musí splňovat všechna **pravidla pro stavbu** (viz. Pravidla pro stavbu).

Při pokládání mohou hráči libovolně kostičky natáčet. Ke stavbě mohou používat obě ruce. Celou stavbu by měli konstruovat co nejstabilnější, přesně jako to je s každým počítačovým programem. Musí totiž myslet na to, že program bude možná potřeba časem upravit podle dalších požadavků. Proto je důležité vytvořit co nejlepší a pevnou základnu.

## **Druhá fáze partie: Psaní samotného programu**

Druhá fáze partie nastává po vyčerpání nabídky 12 kostek, kdy je postavena základní struktura programu. Nyní lze vybírat ze všech zbývajících 38 kostek, a to libovolný dílek, který pak lze přiložit do stavby.

*Programátoři* se snaží postavit z kostiček pyramidu, která bude stabilní. Současně dávají pozor na možná zranitelná místa v programu.

*Hacker* se snaží najít vhodný okamžik, kdy by položil nepozorovaně chybu na správné místo.

## **Identifikace zranitelných míst programu**

*Programátoři* se snaží ve stavbě najít místa, která jsou zranitelná, která by si mohl vybrat hacker. Takové místo by si měli pro jistotu uložit do systému správy verzí (tzv. git).

Kdykoliv ve hře si může hráč v roli *programátora* vybrat libovolný z již položených kamenů (kromě *opěrných bodů programu*) a poznačit si jeho identitu a pozici do své tabulky. Najde si v křížné tabulce políčko odpovídající barvě a tvaru kamene, který je podle něj zranitelným místem a hacker by si ho mohl vybrat pro svůj útok. Do políčka zapíše patro, na kterém kámen leží, tedy počet kamenů pod ním. Svoji volbu ostatním neukazuje a o svých úmyslech ani volbě kamene se NESMÍ RADIT s ostatními hráči. Jedná se pouze o jeho vlastní tip.

## **Zneužití zranitelných míst programu**

*Hacker* se snaží zakrýt své úmysly a zvolit si ve stavbě (= programu) takový kámen, který nebude snadné vyměnit (= opravit). I on si poznačí identitu kamene do tabulky.

Najde si v křížné tabulce políčko odpovídající barvě a tvaru kamene, na který útočí. Do políčka zapíše patro, na kterém kámen leží, tedy počet kamenů pod ním.

Při tom si nemusí počínat nijak tajemně, protože dělá totéž, co všichni ostatní. Ani on nesmí ukazovat svoji volbu ostatním ani se o ní nijak radit.

## **Konec stavby**

Jakmile hráči do stavby přiloží čtyřicátou kostičku (*tip*: nemusíte je počítat, staví vědět, že Vám zbývá deset nepřiložených), je program hotov.

Pokud se stavba zhroutí před dokončením, okamžitě prohrávají všichni hráči, včetně *hackera*. *Programátoři* i *hacker* totiž přišli díky tomu o práci, protože ani jeden nesplnil svůj úkol.

### **Třetí fáze partie: Oprava chyb**

*Hacker* má hotovo. Dokončil svoji práci a v programu ukryl několik chyb.

Práce nyní čeká na *programátory*, kteří musí společnými silami a s využitím zbývajících deseti kostiček nahradit chyby tak, aby je program (stavba) neobsahoval. *Hacker* je pouze sleduje.

Nejdříve ale všichni odhalí své role a zvolené kameny zapsané na *poznámkových kartičkách*.

Pokud některý z *programátorů* uspěl a dokázal správně identifikovat zranitelné místo, které si vybral i *hacker*, nebude třeba takové místo opravovat. Vyřešení této změny v kódu lze udělat jednoduše díky tomu, že si je *programátoři* uložili do systému správy verzí.

Všechny zbývající chyby, které nikdo neidentifikoval před finální verzí programu, je třeba nyní urychleně opravit. *Programátoři* při tom mohou spolupracovat, libovolně stavbu společně nadzvedávat, držet či naklánět. Nikdy ale nesmí jakékoliv části stavby odkládat stranou na stůl ani jinou podložku (s výjimkou kamenů odstraněných chyb).

Každou chybu musí nahradit jedním nebo více kameny ze zásoby deseti nepoužitých kamenů. Tyto kameny mohou libovolně natáčet a kombinovat, aby dokázali chybu nahradit.

Pokud se stavba v tuto chvíli zhroutí, *programátoři* prohrávají a vyhrává *hacker*.

Pokud se *programátorům* podaří nahradit všechny zbývající včas neidentifikované chyby, vyhrávají a *hacker* ve svém záškodnictví neuspěl.

### **Konec hry**

Hra může skončit **prohrou všech hráčů**

- pokud dojde ke zboření stavby kdykoliv v průběhu první fáze, psaní programu

Hra může skončit **výhrou hackera**

- pokud dojde ke zboření stavby v rámci druhé fáze, opravy chyb

Hra může skončit **výhrou programátorů**

- pokud se jim podaří nahradit všechny neidentifikované chyby (kostky zvolené *hackerem*), aniž by se stavba zřítla

## **Pravidla pro stavbu**

- při přikládání hráči **mohou** používat obě ruce, mohou si přidržet stávající sestavu kostiček
- nově přiložené kameny se **musí** vždy dotýkat základny nebo jiného kamene
- kostičky dotýkající se základny nikdy **nesmí** přesáhnout její plochu
- hráči **nesmí** už jednou položené kostičky v rámci prvních dvou fází hry přesouvat ani nadzvedávat

## **Kooperativní varianta**

Pokud neradi soupeříte, budete potřebovat navíc jednu šestistěnnou kostku (není součástí balení).

### **Průběh kooperativní hry**

Před každým tahem v druhé fázi hodí jeden z hráčů šestistěnnou kostkou. Když padne jednička, musí vybrat kámen, označit jej jako chybu a místo svého tahu jej položit na libovolné místo stavby.

V ostatních situacích platí pravidla původní hry.

Při opravě programu spolupracují všichni hráči a vyhrají nebo prohrají společně v závislosti na tom, jestli stavba zůstane stát nebo se zhroutí.

Role	Programátor			Hacker		
Tvar /Barva	Modrá	Žlutá	Zelená	Červená	Šedá	Hnědá
Krychle						
Váleček						
Destička						
Nízký kvádr						
Vysoký kvádr						

Role	Programátor			Hacker		
Tvar /Barva	Modrá	Žlutá	Zelená	Červená	Šedá	Hnědá
Krychle						
Váleček						
Destička						
Nízký kvádr						
Vysoký kvádr						

## **Psaní počítačového programu**

Za každou obrazovku, grafikou nebo textem stojí v konečném důsledku vždy nějaký program, který zajišťuje, že se Vám právě tyto prvky zobrazí a v jaké podobě.

Ten může být psaný v nespočtu různých programovacích jazyků, právě proto jsme v naší hře vše udělali abstraktně a jednotlivé programové bloky zde tvoří barevné kostky. Můžete si je představit jako části kódu, které se starají o různé věci – jedna třeba o zápis dat, druhá o jejich čtení, další zase zajišťuje zobrazení jednoho konkrétního okna nebo upozornění.

Každý program by ale měl být napsaný kvalitně a bez chyb. Ale ne vždy to je pravda, protože v rozsáhlých algoritmech se často může vyskytnout chyba nebo nějaká mezera. Jako třeba když dovřete dveře, ale v nich zůstane malá škvíra. Tou se sice neprotáhne králík ani myš, ale malý brouček už ano. A právě tady vstupují do hry ti, kdo škodí a snaží se tyto drobné škvíry ve struktuře využít ku svému prospěchu nebo prostě jen škodolibě ničit práci druhých.

## **System správy verzí (git)**