

Tomáš Holan

Unity

**První seznámení s tvorbou
počítačových her**

Edice CZ.NIC



UNITY

První seznámení s tvorbou počítačových her

Tomáš Holan

Vydavatel:
CZ.NIC, z. s. p. o.
Milešovská 5, 130 00 Praha 3
Edice CZ.NIC
www.nic.cz

1. vydání, Praha 2020
Kniha vyšla jako 26. publikace v Edici CZ.NIC.
ISBN 978-80-88168-57-7

© 2020 Tomáš Holan

Toto autorské dílo podléhá licenci Creative Commons BY-ND 3.0 CZ (<https://creativecommons.org/licenses/by-nd/3.0/cz/>), a to za předpokladu, že zůstane zachováno označení autora díla a prvního vydavatele díla, sdružení CZ.NIC, z. s. p. o. Dílo může být překládáno a následně šířeno v písemné či elektronické formě, na území kteréhokoliv státu.

Následující názvy/označení, případně další názvy produktů a podobně, které jsou použity v této knize, jsou či mohou být ochrannými známkami či registrovanými ochrannými známkami příslušných vlastníků a jejich použití se řídí právními předpisy.

Unity: Unity Technologies

Windows, .NET a Visual Studio: MICROSOFT CORPORATION,

King's Quest: ACTIVISION PUBLISHING, INC.

Sinclair a ZX Spectrum: Sky In-Home Service Limited

Pac-man: Kabushiki Kaisha BANDAI NAMCO Entertainment

Blender: Blender Foundation

Unity

**První seznámení s tvorbou
počítačových her**

Předmluva vydavatele

Předmluva vydavatele

Vážení čtenáři,

dostává se vám do ruky tak trochu jiná kniha o programování. Odborné publikace se většinou pyšní hloubkou, do které zasahují, a objemem informací, které obsahují. Tato kniha naopak sází na čtivost, praktičnost a obecně zábavnost. A hned v úvodu otevřeně přiznává, že spousty témat se ani nedotkne.

S programováním jsem začínal na základní škole před skoro třiceti lety. A jedna z mých prvních myšlenek samozřejmě byla napsat si hru. Tehdy jsem zvládnul nanejvýš jednoduchý kvíz, ale to mi nebránilo snít. Postupem času jsem se zlepšoval a učil se toho víc a víc, ale začal jsem řešit reálné problémy, a na hry už nebyl čas. Když jsem s určitou technologií začínal, čerpal jsem obvykle z nějaké chytré knihy, která hlásala, že obsahuje vše, co potřebuji vědět. Já se ponořil do jejího čtení a nikdy ji nedočtl, protože po pár kapitolách suché teorie mě to přestalo bavit.

Tato kniha je jiná. Netvrdí, že vás naučí vše o Unity. Právě naopak, hrdě se hlásí k tomu, že neobsahuje ani solidní základy, natož kompletní popis celého enginu. Její předností je něco jiného, podle mě daleko zásadnějšího. Je velmi čtivá a názorná. Ukazuje, jak začít. Začne jednoduchým zadáním a snaží se ho naplnit. Vysvětluje, co je potřeba k dosažení cíle. Nestrávíte kapitoly čtením teorie, abyste se na konci knihy dozvěděli, že vám to možná někdy k něčemu bude. Naopak. Kniha začíná od toho, co je naším cílem, a pak teprve řeší, jak ho dosáhnout. Případně jak cíl změnit. Protože v reálném světě neexistuje perfektní zadání. Existují celé vývojářské týmy, které pracují roky na funkcích, které se nakonec ukáží jako okrajové a zbytečně složité, a jsou zahozeny. A přitom by mnohdy stačilo se nad zadáním včas zamyslet a upravit ho. V knize naleznete momenty, kdy se fungování hry změní, protože naplnění původního plánu by bylo zbytečně složité a nakonec by ani nemuselo fungovat dobře. Stejně jako v reálném životě do sebe v knize vše nezapadá na první pokus. Objevují se problémy, které je třeba řešit, a kniha ukazuje jak. I díky tomu vás „vtáhne do děje“.

Pro koho je tedy kniha určena? Není to publikace, kterou bude mít programátor her na polici, a po které sáhne, když nebude vědět, jak se nějaká obskurní konstrukce používá. Je to kniha, kterou zhltnete, pokud si chcete naprogramovat nějakou hru a nevíte, jak začít. A nemusí být jen pro ty, kteří už umí programovat. Ukázky jsou velmi názorné a jednoduché. Osobně mě překvapilo, jak málo a jak jednoduchý kód vlastně stačí k tomu, udělat si vlastní hru. Může to tak být i vhodná kniha pro všechny, kteří přemýšlí, jak začít s programováním vůbec. Za málo práce hodně zábavy. A jak s tím jednou začnete, už vás to jen tak nepustí.

Přeji vám zábavné čtení a těším se na spousty nových Indie her, u jejichž zrodu bude právě tato kniha stát.

Michal Hrušický, CZ.NIC

Obsah

Předmluva vydavatele	7
1 Úvod	15
1.1 Procházka	15
1.2 Unity engine	16
1.3 Zdroje informací	18
2 Začínáme	21
2.1 Začínáme	21
2.2 Prostředí Unity	22
2.3 Fyzika	26
2.4 Vlastnosti, parametry atributy	27
2.5 Kamera	31
2.6 Konec začátku	32
3 Pokračujeme	35
3.1 Návrh hry	35
3.2 Nový projekt	35
3.3 Pohyb, skripty, ovládání	36
3.4 Skript	38
3.5 Pohyb	43
4 Hra	53
4.1 Prefab	53
4.2 Jak změnit prefab	58
4.3 Pohyb kamery	60
4.4 Znovu pohyb hrdiny	65
4.5 Tvar hrdiny	70
5 Bludiště a další	75
5.1 Bludiště	75
5.2 Pozice hrdiny	79
5.3 Bludiště jako asset	81
5.4 Jak generovat bludiště skriptem	83
5.5 Ještě jednou pohyb	85
5.6 Vyslat paprsek	85
5.7 Sběr	86
5.8 Počet sebraných mincí a UI	92
5.9 Zvuky	97
5.10 Sestavení	98
5.11 Kde je tyranosaurus?	99

6	Projekt Tyranosaurus	103
6.1	Návrh	103
6.2	Nový projekt	103
6.3	Scény	103
6.4	Objekty ve scéně	104
6.5	Assets - export a import	106
6.6	Skripty	108
6.7	Sbírání mincí a detekce kolizí	112
6.8	Příprava Roviny	114
6.9	Poloha Mincí a Hrdiny	118
6.10	Problém: Otáčení při nárazu	122
7	Objekt Tyranosaurus	125
7.1	Příprava	125
7.2	Zapojení do hry	126
7.3	Ladění v Unity	127
7.4	Zpátky k Tyranosaurům	129
7.5	Pohyb Tyranosaury	130
7.6	Světlo	133
8	Scény	137
8.1	Zamyšlení	137
8.2	Přetrvávající objekty	138
8.3	Zpátky k první scéně	139
8.4	Zapojení objektu Data	144
8.5	Scéna Konec	146
8.6	Leccos tam schází	148
	Příloha 1: Mapa	151
	Příloha 2: Skripty	155
1	Hra	155
2	Tyranosaurus	163

1 Úvod

1 Úvod

1.1 Procházka

Tohle je kniha o programování her v Unity enginu a je určená těm, kdo by se chtěli s Unity seznámit, ale nevědí, odkud začít.

Nebude to učebnice, kde bych vyjmenovával všechna okna, tlačítka a položky v menu. Budu psát program a co budu potřebovat, to použiju a okomentuju. Taková společná procházka, možná trochu upovídaná.

Expert

Nejsem žádný expert na Unity, na druhou stranu programuju více než čtyřicet let a leccos z toho, co jsem programoval, byly hry, tak věřím, že mám trochu nadhled a že to půjde. A že pokud mi nějaká znalost bude chybět, tak si ji dokážu najít. Vzhůru do toho!

Varování

Budu ukazovat vývoj programu nebo programů a to zahrnuje i změny plánů a nesplněná předsevzetí. Mohl bych se jim vyhnout a ukázat vám jen výsledek, kde všechno pěkně vychází, ale připadá mi, že to bloudění a přehodnocování k vývoji programu patří. Navíc mi to dovolí ukázat víc možností, jak něco udělat, než jen jedinou správnou. A nakonec, chceme se něco naučit, takže cesta je cíl!

Hry

Když říkám hry, tak tím nemyslím šachy ani Hledání min, i když to by šlo v Unity naprogramovat taky. Půjde mi o hry, ve kterých se něco hýbe, chodí, létá, padá, plave.

Herní smyčka

Takové hry mají v sobě vlastně model celého herního světa – údaje o tom, kde je jaká postavička překážka, věc... a celá hra potom probíhá v něčem, čemu se historicky říká herní smyčka, tedy neustálé opakování dvou kroků:

- pohni všemi objekty světa
- vykresli svět

Herní objekt

Ano, ty postavičky, překážky, věci... prostě všechno, co ve hře má nějaký význam, se shrnuje pod označení **herní objekt**.

1.2 Unity engine

Cizí kód

Fred Brooks v roce 1987 napsal článek „*No Silver Bullet – Essence and Accident in Software Engineering*“, kde se zamýšlí nad tím, jestli něco dokáže v průběhu deseti let řádově urychlit vývoj software (nebo ho urychlovat tak, jak se zrychluje výkon hardware) – a dochází k závěru, že ne.

Softwarové projekty už tehdy měly zpoždění, nedodržovaly rozpočet a trpěly podobnými neduhy. Článek postupně prochází možnostmi, které by mohly pomoci, včetně automatického programování (pamatuju, byl to hit) nebo objektového programování (to už dnes používá snad každý) a postupně dochází k tomu, že žádná z těch nových technik nedokáže řádově urychlit vývoj programů, prostě proto, že vývoj programů je v principu složitý (to je ta „essence“ z názvu článku).

Ale při pohledu zpět vidíme, že vývoj programů dnes rychlejší je; třeba program napodobující kapsní kalkulačku, s tlačítky a ovládním myši dokáže student prvního ročníku vysoké školy napsat za hodinu, možná méně. A je to díky možnosti, kterou autor článku nezapočítal – možnosti použít cizí kód! (Slovem kód tady označuji i dále budu označovat program, celý nebo část, ve zdrojovém tvaru nebo nějak přeložený.)

Když zmíněný student píše program ovládaný myší, tak nemusí řešit komunikaci s myší, ať už pomocí sériového rozhraní RS232 (dříve) nebo pomocí USB (dnes), protože o to se stará operační systém. A pokud jiný student chce ve své webové stránce vykreslovat trojrozměrné animace, použije na to nějakou knihovnu (nebudu jmenovat, ale je jich víc).

Používání cizího kódu s sebou nese spoustu problémů, ale o tom teď nechci mluvit.

Knihovna, framework, engine

Knihovna je jedna z možností, jak sdílet a znovu-používat dříve napsaný kód.

Většina programovacích jazyků umožňuje do zdrojového kódu začlenit další kód, buďto ve formě zdrojového kódu nebo výsledek překladu a tak stačí příslušné funkce (případně data, definice tříd a podobně) vložit do nějakého zdrojového kódu nebo projektu nebo balíčku (různé názvy v různých programovacích jazycích), který budeme dále používat; možná my a možná i někdo

jiný. Takovému (zdrojovému) kódu nebo množině (zdrojových) kódů potom říkáme knihovna. Z knihovny si vytáhneme, co se nám hodí, a také můžeme klidně použít více knihoven najednou, třeba jednu pro kreslení a druhou pro komunikaci po internetu.

Framework funguje v jistém smyslu naruby – zatímco u knihovny „hlavní program“ píšeme my a vybíráme si, co použijeme z cizích knihoven, u frameworku je to tak, že funguje sám a my pouze můžeme doplňovat nebo měnit některé funkce. Třeba když v .NET Frameworku vytvoříme WinForms-aplikaci, nemusíme napsat ani řádku kódu a přesto máme program, který jde spustit, zobrazí okno, které se dá zvětšovat a přesouvat myši i klávesami, zobrazuje svou ikonku v seznamu běžících programů, zobrazuje se ve správci úloh a má řadu dalších funkcí. A pokud chceme, můžeme pozměnit jeho funkci, pomocí které na začátku vytváří obsah svého okna a přidat tam nějaká tlačítka nebo jiné komponenty a můžeme také přidat funkce, které budou reagovat na události, a tím vlastně zařídit, že ten program bude dělat to, co chceme. Ale hlavní program jsme měli od samého začátku hotový a jenom doplňujeme a vylepšujeme. A hlavní program jsme nenapsali my, ale autoři frameworku.

Herní engine (a Unity není jediný (pamatuje někdo AGI? Třeba King's Quest od Sierra Online?), viz třeba Unreal Engine, Cry Engine nebo Godot) je povahou spíš framework.

Obvykle zahrnuje **vykreslování** (protože to je spousta práce a herní návrháři mají jiné zájmy), **detekci kolizí** (abychom poznali, kdy na sebe objekty narazily), **fyziku** (aby na objekty mohly působit síly, setrvačnost a další objekty), **herní smyčku** plus nástroje na vytváření a editaci herního světa přidáváním a nastavováním herních objektů. Většina herních engineů dovoluje také programování, ať už ve svém vlastním jazyku nebo v některém z univerzálních jazyků. V Unity lze programovat v jazyku C#.

Dostupnost, licence

Unity engine je ke stažení ze stránek společnosti, dostupná je jak bezplatná, tak placená verze. Bezplatná verze je omezená pro osobní použití a pro společnosti se ziskem do určité výše.

Verze

Jako každý software, který se používá, i Unity engine se vyvíjí. To znamená, že pokud si stáhnete nejnovější verzi, za čas už nebude nejnovější. Novější verze může být lepší (proto to dělají), ale Unity nemusí umět bez problémů otevřít projekt, který jste vytvořili v jiné verzi (viz ta moje poznámka o problémech s používáním cizího kódu). Takže (obzvlášť pokud chcete vytvářet nějakou hru společně s dalšími lidmi) je důležité domluvit se, jakou verzi Unity budete používat.

1.3 Zdroje informací

Návodů, jak v Unity udělat to či ono, je plný internet. Nejspolehlivější, nejpečlivější, nejudržovanější a další nej- je **Unity Manual** od autorů Unity na stránce <https://docs.unity3d.com/Manual/>, nezapomeňte si vlevo nahoře vybrat verzi Unity, o které chcete informace. Můžete si ho i stáhnout jako zip.