



Středoškolská technika 2016

Setkání a prezentace prací středoškolských studentů na ČVUT

Interakce publika s živým přenosem pomocí protokolu IRC

Jan Voříšek

Gymnázium Zikmunda Wintra
náměstí Jana Žižky 186, Rakovník

Poděkování:

Děkuji svému konzultantovi Mgr. Martině Voříškové za pomoc při tvorbě mé práce.

Jan Voříšek

Abstrakt

Cílem práce je představit způsob tvorby interaktivního projektu netradičním způsobem, zaměřil jsem se především na práci s IRC protokolem a jeho použitím v kombinaci s živým online přenosem. Pro technickou realizaci projektu jsem využil program GameMaker: Studio a Open Broadcaster Software. Data o průběhu jsem sbíral vlastní metodou – programově do textových souborů, které jsem následně zpracoval do grafů a tabulek.

Anotace

Tato práce je rozdělena na teoretickou a praktickou část. Teoretická část se zabývá prací s IRC protokolem a jeho praktickým využitím ve spojení s živým online vysíláním. Praktická část práce se týká tvorby a průběhu mého vlastního projektu, předložení výsledků a jejich analýzou.

Klíčová slova

Internet Relay Chat, IRC, komunikace, živé vysílání, sociální experiment, vývoj softwaru, internet

Abstract

This aim of this paper is to present a new, unusual way of creating interactive online applications. It focuses on the practical usage of the IRC protocol in combination with video streaming and the creation of my own project built on this concept using GameMaker: Studio and Open Broadcaster Software. The paper concludes with a presentation of statistical data gathered by the application during its one-week run.

Annotation

This paper is divided into two parts – theoretical and practical. The theoretical part discusses the IRC protocol and its usage in combination with video streaming, whereas the practical part documents planning, creation and realisation of my own project based on this principle.

Keywords

Internet Relay Chat, IRC, online communication, live broadcast, social experiment, software development, internet

Obsah

Úvod.....	6
1. IRC protokol.....	7
1.1 Základní informace.....	7
1.2 Klienti.....	9
1.3 Kanály.....	9
1.4 Způsoby komunikace přes IRC	10
1.4.1 One-to-one.....	10
1.4.2 One-to-many.....	10
1.4.2.1 Komunikace pomocí seznamu	10
1.4.2.2 Komunikace se skupinou (channel)	10
1.4.2.3 Komunikace pomocí masky.....	10
1.4.3 One-to-all	10
1.4.4 CTCP a DCC.....	11
1.5 Zprávy a příkazy.....	11
1.5.1 Odpovědi.....	13
1.6 Připojení k IRC síti.....	13
1.7 IRC Služby	16
1.8 Módy.....	17
1.9 Bezpečnostní rizika.....	19
2. Twitch.....	20
2.1 Základní informace.....	20
2.2 Kanály.....	21
2.3 Chat.....	21
2.4 Twitch Partner Program.....	22
2.5 Streamování	22
2.6 Přehrávání.....	24

3. Twitch Plays Pokemon	25
4. Praktická část – Twitch Plays Hangman	28
4.1 Prvotní podnět.....	28
4.2 Použitý software	28
4.2.1 GameMaker: Studio	28
4.2.2 Open Broadcaster Software.....	30
4.3 Vývoj	30
4.3.1 Postup hry.....	31
4.3.2 Grafika.....	34
4.4 Sběr dat.....	34
4.5 Průběh a statistiky.....	35
4.5.1 Výsledky komunity	36
4.5.2 Výsledky jednotlivých hráčů.....	38
5. Závěr	39
6. Zdroje.....	40
6.1 Bibliografie.....	40
6.2 Internetové zdroje	40
6.3 Zdroje obrázků.....	41

Úvod

Internet a s ním spojené technologie se vyvíjejí rapidní rychlostí, o čemž svědčí fakt, že se neustále objevují nové a nové způsoby vzdálené komunikace. Co se ovšem stane, když někdo zkombinuje novou a starou technologii za účelem vytvoření něčeho zcela unikátního?

Jakožto člověka se zájmem v informačních technologiích mě velice zaujal fenomén, který se v roce 2014 na síti objevil díky neznámému australskému programátorovi. Ten si uvědomil, že by populární webovou platformu pro vysílání videa online mohl použít naprosto netradičně. Využil jejího technologického nedostatku k realizaci sociálního experimentu, který svou originalitou poté inspiroval vývojáře po celé Zemi.

O čem že je řeč?

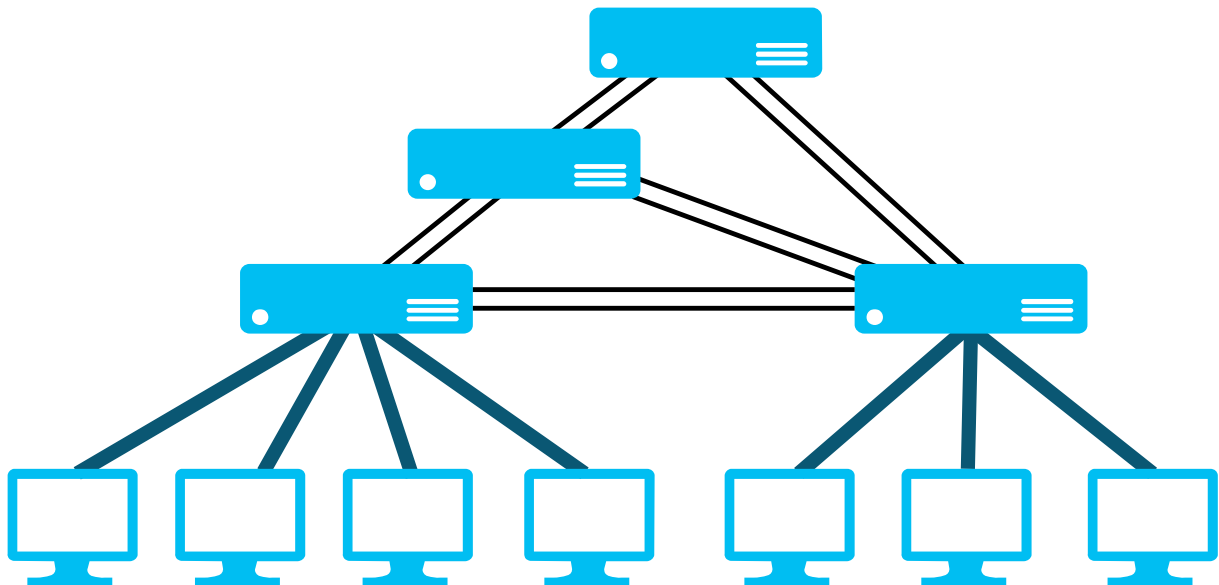
Vytvořil „interaktivní živý přenos“ za pomoci běžně dostupných bezplatných technologických prostředků. Umožnil internetovým uživatelům hrát klasickou videohru pouze pomocí jednoduchých textových příkazů zasílaných do chatu. Výsledek byl vsutku neočekávaný. I přesto, že byl obraz nejméně o deset vteřin opožděný pro většinu diváků (kterých bylo někdy i přes dvě stě tisíc online), komunita, která se kolem projektu utvořila, jako zázrakem hru dohrála.

V teoretické části mé práce si vysvětlíme, jak něco takového bylo technicky možné, v té druhé, praktické, vás provedu tvorbou a průběhem mého vlastního projektu inspirovaného tímto originálním experimentem.

1. IRC protokol

1.1 Základní informace

IRC (*Internet Relay Chat*) je otevřený komunikační protokol v aplikační vrstvě navržený pro přenos textu v reálném čase. Využívá síťovou architekturu klient/server (*obr. 1*) a transportní protokol TCP. Podporuje jak komunikaci ve skupinách, tzv. kanálech (*channels*), tak soukromou mezi klienty. Vytvořil jej v roce 1988 finský programátor Jarkko Oikarinen jako náhradu programu MultiUser Talk na finském BBS¹ OuluBox. V květnu roku 1993 byla vydána oficiální dokumentace IRC protokolu - RFC 1459. Stal se velice populárním, byl například také využíván jako způsob šíření zpráv během Války v Zálivu, kdy v běžných médiích docházelo k cenzuře. Podle dostupných údajů ztratil od roku 2003 kolem 60% svých uživatelů, dnes jej využívá zhruba 400 tisíc lidí. Pro svou jednoduchost se však stále využívá v menších komunitách nebo na webových stránkách, pro které je svými vlastnostmi ideální.

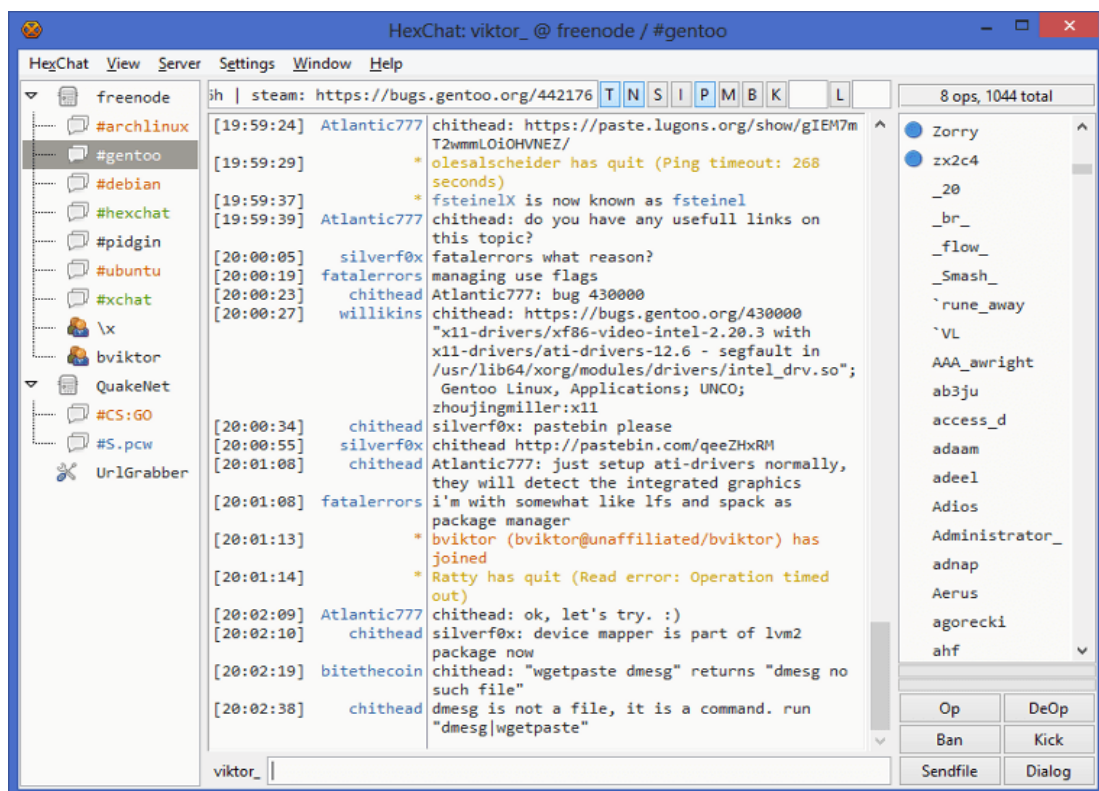


Obr. 1 - příklad zapojení IRC sítě

¹ *Bulletin Board System* – Systém „elektronických nástěnek“ běžící v počítačovém terminálu, mohl by se považovat za předchůdce dnešních internetových diskuzních fór.

Každá IRC síť se skládá z dvou a více vzájemně propojených serverů. Na každém serveru běží tzv. Internet Relay Chat daemon (IRCd) – program vyřizující požadavky od IRC klientů, kteří se na něj připojují. Téměř každá větší IRC síť si IRCd upravila podle svých představ a nedrží se striktně IRC dokumentace. Přes IRC lze dnes tedy například posílat i soubory, což ovšem přináší bezpečnostní rizika. Mezi přidané funkce můžeme řadit například „*color codes*“ – obarvování textu, tagy, šifrování komunikace pomocí SSL², podporu IPv6, nové příkazy, módy či služby a také detekci klientů připojujících se prostřednictvím proxy serveru³. Značná část těchto vylepšení je spojena s velmi známou skupinou vývojářů - IRCv3, která se aktivně snaží prosadit jejich standardizaci.

Pro běžnou komunikaci pomocí IRC je potřeba program, tzv. IRC Klient, jako třeba mIRC, Pidgin nebo HexChat (obr. 2). Většina IRC klientů je dostupná zdarma téměř na všech platformách.



Obr. 2 - prostředí programu HexChat

² *Secure Sockets Layer* – kryptografický protokol mezi transportní a aplikační vrstvou zprostředkovávající šifrovaný přenos dat v sítích. Dnešní verze se nazývá TLS – *Transport Layer Security*.

³ *Proxy server* – Server fungující jako prostředník mezi klientem a cílovým serverem, využíván za účelem anonymizace.

Dne 9.12.2015 mezi deset světově nejaktivnějších IRC sítí patřily tyto: IRCNet, QuakeNet, EFnet, Rizon, Undernet, ChLame, IRC-Hispano, OFTC, LinkBR a DALnet. Nejznámější českou IRC sítí je CZFree.

1.2 Klienti

Jakýkoliv přístroj připojený do sítě, který není serverem, se nazývá *klient*. V ní se identifikuje pomocí unikátní přezdívky (*nickname*), která může být maximálně 9 znaků dlouhá. Každý server potřebuje o každém klientovi znát tyto informace: adresu hostitele, na kterém klient běží, přezdívku (popřípadě jméno a příjmení) a adresu sítě, ke které se připojuje.

1.3 Kanály

Kanál (*channel*) by se dal označit jako „komunikační místnost“. Zprávy v něm odeslané obdrží pouze klienti na něj připojení (jeden klient může být připojen na několik kanálů najednou). Pokud se klient zkusí připojit na neexistující kanál, server ho vytvoří. Jakmile se na kanále nenachází žádní klienti, automaticky zaniká. Některé IRC sítě umožňují klientům zaregistrovat nové sítě, většinou pomocí jednoduchého příkazu. Pokud se tak stane, zakládající klient obdrží status *founder/owner* (*zakladatel/vlastník*) a zůstává přítomen na kanále, i když k němu není fyzicky připojen.

V síti se kanál označuje textovým řetězcem začínajícím znakem „#“ nebo „&“ a jménem samotným, které může být až 200 ASCII⁴ znaků dlouhé. Ve jméně se nesmí vyskytovat mezery, čárky, a kontrolní znaky G (7 znak v ASCII).

IRC síť umožňuje existenci dvou typů kanálů – první, se jménem začínajícím znakem „#“, je známý všem serverům v síti. Druhý, se jménem začínajícím „&“, je známý pouze serveru, na kterém byl vytvořen.

Každý kanál může mít také nastavené tzv. téma (*topic*) – text zobrazující se u názvu kanálu ve většině komunikačních programů.

⁴ ASCII – American Standard Code for Information Interchange, je tabulka anglických, interpunkčních a kontrolních znaků určená ke kódování textu. První verze vznikla roku 1963. Kvůli své zastaralosti a problémům (1 znak byl kódován jako 8 bitů, což velikost tabulky značně limitovalo) byla však v roce 2007 nahrazena mnohem vyspělejší metodou kódování textu - UTF-8.

1.4 Způsoby komunikace přes IRC

1.4.1 One-to-one

Komunikace mezi klienty v síti je z důvodu bezpečnosti navrhnutá tak, že zpráva projde v IRC síti co nejmenším počtem serverů, přičemž o ní ostatní servery vůbec neví.

1.4.2 One-to-many

1.4.2.1 Komunikace pomocí seznamu

Nejméně efektivní způsob komunikace způsobem one-to-many (komunikace klienta s několika jinými klienty najednou) funguje na stejném principu jako one-to-one komunikace, rozdíl je v tom, že místo zaslání zprávy jednomu klientu, se zpráva odesílá několika klientům najednou.

1.4.2.2 Komunikace se skupinou (channel)

Lepší způsob využívající již zmíněné kanály – skupin, jejichž existence je dynamická. Zprávy jsou zasílány pouze serverům, na kterých jsou klienti připojeni. Pokud se na jednom serveru a kanálu nachází několik klientů, je serveru zpráva zaslána pouze jednou, a on sám ji poté odešle potřebným klientům.

1.4.2.3 Komunikace pomocí masky

Princip je podobný jako u kanálů, v tomto případě je však zpráva zaslána uživatelům, jejichž hostitelské adresy odpovídají masce připojené u zprávy. Je tedy založen na lokaci.

1.4.3 One-to-all

Nejméně využívaným způsobem komunikace je odesílání zprávy všem serverům a klientům. Zejména ve velkých sítích dokáže odeslání i jediné zprávy tímto způsobem vygenerovat velké

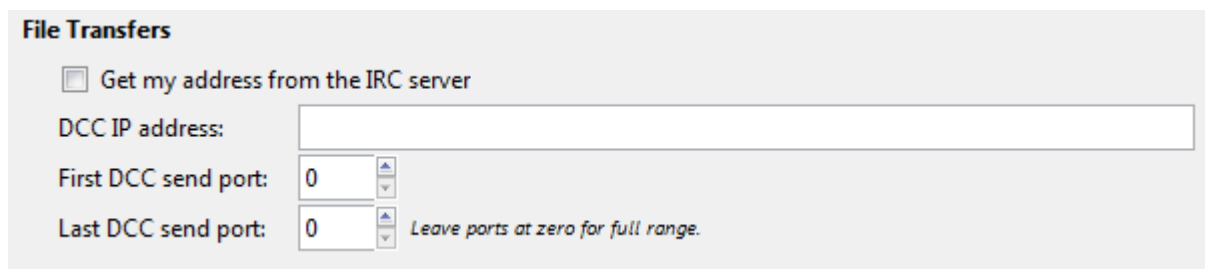
zatížení. Využívá se tedy zřídka, například při „vysílání“ důležitých zpráv o změně ve stavu serverů či kanálů (*NOTICE*).

Tento typ zprávy nemůže odeslat klient ostatním klientům, ale pouze serveru (při změně vlastnictví kanálu či změně módu, atd.). Ten ji poté předá potřebným serverům v síti.

1.4.4 CTCP a DCC

CTCP (*Client-to-Client Protocol*) protokol rozšiřuje možnosti komunikace mezi IRC klienty. Implementuje jej většina komunikačních programů. Umožňuje odesílání dotazů (*query*) na specifické informace (jako verze komunikačního softwaru - *VERSION*), či odesílání zpráv s kódováním, které obyčejné IRC nepovoluje (např. obsahující znaky pro odřádkování).

Mezi hlavní funkce CTCP patří především možnost vytvoření soukromého připojení mezi klienty prostřednictvím podprotokolu DCC (*Direct Client-to-Client*). Takové připojení vzniká přímo mezi nimi, bez využití jakéhokoli IRC serveru. Umožňuje například zasílání souborů, což je jedním z několika bezpečnostních rizik, které DCC přináší. Často se tedy doporučuje jeho vypnutí v nastavení našeho programu (*obr. 3*). Jelikož se nejedná o součást IRC protokolu, všechny CTCP požadavky se posílají jako obyčejné zprávy (*PRIVMSG*).



Obr. 3 - nastavení přenosu souborů přes DCC v programu HexChat

1.5 Zprávy a příkazy

Zprávy, se kterými IRC servery pracují, se většinou skládají ze tří částí: předpony (*prefix*), příkazu (*command*) a parametrů příkazu (*command parameters*), kterých může být až patnáct. Všechny části jsou odděleny jednou či více mezerami (32. znakem v ASCII).

K tomu, aby klient (uživatel) mohl síť využívat, potřebuje zasílat serveru příkazy. Například i běžnou zprávu chápe server jako příkaz (*PRIVMSG*, obr. 4). Pokud se klient chce například připojit ke kanálu, použije spolu s jeho jménem příkaz *JOIN*. Zprávy od serveru mají stejnou formu, jako zprávy od klienta.

```
<< PRIVMSG #kanal :Test.
```

Obr. 4 – Příkaz k odeslání obyčejné zprávy v podobě srozumitelné serveru.

Zpráva nemusí pokaždé začínat předponou, její přítomnost určuje první znak zprávy – je-li jím dvojtečka („:“), potom ji můžeme očekávat. Pokud neexistuje, je serverem předpokládáno, že zpráva pochází od klienta z připojení, ze kterého přišla. Většina klientů by proto předponu k odesílaným zprávám připojovat neměla, pokud ano, může obsahovat pouze přezdívku a *host mask*, pod kterou se klient na server přihlásil. Při jakýchkoliv neshodách s předponou a interní databází serveru je jím zpráva ignorována.

IRC využívá dnes již zastaralý protokol *Ident* pro identifikaci jednotlivých uživatelů (užitečný například v případě, že několik uživatelů využívá stejnou IP adresu). Při připojování na IRC síť většina serverů odešle klientovi *Ident* požadavek. V případě, že na něj klient neodpoví, je použita přezdívka, pod kterou se přihlašoval a je označen předponou „~“. První IRC zpráva by tedy mohla začínat například takto: „:nick!ident@host“.

Příkaz se musí značit buď jako validní IRC příkaz (např. „MODE“), nebo jako trojciferné číslo. Zprávy musí být zakončeny netisknutelnými znaky CR+LF (*Carriage Return + Line Feed*, 13. A 10. znak v *ASCII*) a nesmí přesahovat délku 512 znaků.

K ulehčení se ve většině komunikačních softwarů příkazy zasílají jako běžné zprávy, musíme k nim však přidat na začátek lomítko („/“). Sám program je poté přeloží do zprávy srozumitelné serverem a odešle. Pokud bychom se například nacházeli v síti a chtěli bychom se připojit ke kanálu „#kanal“, do pole programu bychom zadali příkaz „/join #kanal“ (obr. 5).

```
JanVorisek /join #kanal|
```

Obr. 5 - manuální připojení ke kanálu v programu *HexChat*

Série příkazů WHO, WHOIS a WHOWAS je využívána k zjištění informací o uživateli. Komunikační programy například využívají WHO k tomu, aby mohly sestavit seznam uživatelů aktivních na kanále. Ze seznamu se může klient skrýt nastavením módu *+i*.

1.5.1 Odpovědi

V některých případech (například při vzniku chyb) může server či klient na určité zprávy odeslat odpověď. Ta se skládá z trojčíferného čísla, jména a samotné odpovědi.

Příklady:

- **401** – *ERR_NOSUCHNICK*, „<nickname> :No such nick/channel“, je odeslána, pokud uživatel použije neplatnou přezdívku jako jeden z parametrů nějakého příkazu.
- **402** – *ERR_NOSUCHSERVER*, „<server name> :No such server“, může být naopak odeslána, snaží-li se klient připojit na neexistující server.
- **412** – *ERR_NOTEXTTOSEND*, „:No text to send“ – žádný text k odeslání.
- **451** – *ERR_NOTREGISTERED*, „:You have not registered“ – odpověď odeslaná serverem, pokud uživatel není registrován.
- **421** – *ERR_UNKNOWNCOMMAND*, „<command> :Unknown command“ – klient se pokusil použít neexistující příkaz (obr. 6).

```
[06.12.2015 11:33:25] *|test :Unknown command
```

Obr. 6 - odpověď 421

1.6 Připojení k IRC síti

IRC protokol je asynchronní – klient může odeslat serveru několik příkazů po sobě nedbaje na to, zda server dokončil zpracování prvního. Při používání IRCv3 funkcí během připojování se může tedy stát, že server obdrží jiný příkaz, než ten, který očekával. Tento problém řeší systém CAP (*Client Capability Negotiation*) pracující na principu „otázka-odpověď“ - sérií příkazů klient od serveru zjistí, jaké IRCv3 funkce server podporuje. Pokud se chceme připojit na IRC síť moderním programem, je vhodné nejprve odeslat IRCv3 příkaz „CAP LS“. Jestliže server IRCv3 podporuje, odešle nám seznam jím podporovaných funkcí. V

případě, že odpověď neobdržíme, IRCv3 server nepodporuje vůbec. Proto je důležité odeslat tento příkaz před zahájením registračního procesu.

Následuje odeslání těchto standardních IRC příkazů v tomto pořadí – *PASS* (+heslo, vyžaduje-li ho server), *USER* a *NICK*. Příkaz *USER* specifikuje v parametrech vyžadované informace o klientovi. (jméno, hostname, adresa sítě...) *NICK* využívá jediný parametr – přezdívku klienta.

Poté běžně server odešle zprávu *PING* – na tu klient po obdržení musí odpovědět zprávou *PONG*. Server tak testuje, zda je klient stále připojen. Pokud vše proběhlo bez problémů, server odešle klientovi *MOTD* – *Message of the Day* – zprávu obsahující informace o síti, její pravidla, služby, atd. (obr. 7).

```
[09.12.2015 09:50:29] * - irc.rizon.so Message of the Day -
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - oo
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - 88d888b. dP d888888b .d8888b. 88d888b.
[09.12.2015 09:50:29] * - 88' `88 88 .d8P' 88' `88 88' `88
[09.12.2015 09:50:29] * - 88 88 .Y8P 88. .88 88 88
[09.12.2015 09:50:29] * - dP dP d888888P `88888P' dP dP
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - Rizon Chat Network -- http://rizon.net
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - Listening on ports: 6660 - 6669, 7000. SSL: 6697, 9999
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - This server is partly sponsored by Darkness Reigns (Holding) B.U.
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - Rules:
[09.12.2015 09:50:29] * - o No spamming or flooding
[09.12.2015 09:50:29] * - o No clones or malicious bots
[09.12.2015 09:50:29] * - o No takeovers
[09.12.2015 09:50:29] * - o No distribution of child pornography
[09.12.2015 09:50:29] * - o Clients must respond to VERSION requests
[09.12.2015 09:50:29] * - o Rizon staff may disconnect clients for any or no reason
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - First steps:
[09.12.2015 09:50:29] * - o To register your nick: /msg NickServ HELP
[09.12.2015 09:50:29] * - o To register your channel: /msg ChanServ HELP
[09.12.2015 09:50:29] * - o To get a vHost: /msg HostServ HELP REQUEST
[09.12.2015 09:50:29] * - o For other help with Rizon: /join #help
[09.12.2015 09:50:29] * -
[09.12.2015 09:50:29] * - Usage of this network is a privilege, not a right. Rizon is a
[09.12.2015 09:50:29] * - transit provider, therefore no person or entity involved with
[09.12.2015 09:50:29] * - *.rizon.net or irc.rizon.so takes any responsibility for
[09.12.2015 09:50:29] * - users' actions. Absolutely no warranty is expressed or implied.
```

Obr. 7 - MOTD síť Rizon

Je nutno podotknout, že při připojování tyto zprávy odesílá většina komunikačních programů automaticky, pokud jsme v něm potřebné informace předem nastavili. Pro uživatele je tedy vstup na síť otázkou stisknutí tlačítka.

Po dobu přítomnosti na síti (přibližně každých 20 sekund) by měl klient odesílat serveru příkaz *PING* (a obdržet *PONG*), obě strany se tak ujistí, že je připojení mezi nimi stále aktivní (podobně, jako při připojování). I tuto akci dělá program automaticky.

Předtím, než klient opustí síť, měl by odeslat příkaz *QUIT* s případnou zprávou, která se ukáže ostatním uživatelům. Server tak s ním může ukončit připojení a již od něj neočekávat žádné další zprávy.

1.7 IRC Služby

Za účelem rozšíření funkcionality moderních IRC sítí byla nezávislými vývojáři vytvořena řada služeb (*services*) – robotizovaných klientů („*botů*“) umožňujících například autentizaci uživatelů a kanálů či lepší správu a monitorování IRC sítě.

Nejpoužívanější dvojice autentizačních služeb - *NickServ* a *ChanServ*, vyvinuta týmem Anope⁵ jako součást stejnojmenného balíku IRC služeb, umožňuje snadnou registraci a přihlašování uživatelů i kanálů. Jelikož se nejedná o součást IRC protokolu, ovládají se zasíláním soukromých zpráv (*PRIVMSG*) obsahujících příkaz s příslušnými parametry. Pokud například přijde uživatel poprvé na síť s přezdívkou, kterou si nikdo neregistroval, bude tak moci učinit pomocí příkazu *REGISTER* (obr. 8). Mezi parametry této funkce patří heslo a případně e-mailová adresa, pokud si chce uživatel nechat zaslat verifikační e-mail (vyžadováno většinou IRC sítí). Po zaslání této zprávy bude přezdívka společně s heslem uložena do databáze NickServu.

```
[07.12.2015 19:27:16]-NickServ- | Your nick isn't registered.  
JanVorisek | /msg NickServ REGISTER heslo foo@bar.net|
```

Obr. 8 - registrace přezdívky

V případě, že se na síť připojí uživatel s přezdívkou, která již v databázi NickServu existuje, bude vyzván k identifikaci pomocí příkazu *IDENTIFY* (obr. 9), pokud tak neučiní, musí si změnit přezdívku. Přihlášený uživatel může také upravovat vlastnosti svého účtu pomocí příkazů *SET* a *ACCESS*.

```
[05.12.2015 17:26:42]-NickServ- | This nickname is registered and protected. If it is your  
[05.12.2015 17:26:42]-NickServ- | nick, type /msg NickServ IDENTIFY password. Otherwise,  
[05.12.2015 17:26:42]-NickServ- | please choose a different nick.
```

Obr. 9 - vyzvání ke změně přezdívky, pokud je již registrovaná jiným uživatelem.

⁵ Oficiální stránky týmu Anope - <https://www.anope.org>

Registrované přezdívky, které nebyly po určitou dobu využívány, jsou ze systému odstraněny a uvolněny pro případné jiné zájemce. Některé sítě tak mohou učinit i u registrovaných přezdívek, u kterých nedošlo k validaci pomocí e-mailu.

Mezi další často používané příkazy NickServu patří:

- **LOGOUT** – Odhlásí uživatele (opak *IDENTIFY*)
- **INFO <nickname>** - vypíše informace o přezdívce – vlastník, poslední adresa a čas, nastavení...
- **ACCESS <ADD/DEL> <mask>** - umožňuje přidat naši adresu (host mask), na seznam (*Access list*). Pokud NickServ nalezne naši adresu na tomto seznamu, rozezná nás jako vlastníka přezdívky bez nutnosti identifikace pomocí *IDENTIFY*.
- **LIST <pattern>** - Vyhledá a vypíše všechny registrované přezdívky splňující pravidlo specifikované v parametru.
- **SET <option> <parameters>** - změna různých nastavení, například *SET PASSWORD* (změna hesla), *SET PRIVATE* (viditelnost na seznamu přezdívek), *SET SECURE* (přístup k přezdívce pomocí access listu – nutnost identifikace pomocí *IDENTIFY*)...

Pomocí NickServ se uživatel také může přidávat do skupin přezdívek (*GROUP*), ve kterých si mohou uživatelé sdílet konfigurace, privilegia či poznámky (mema).

ChanServ – služba pro registraci kanálů, funguje obdobně – zakladatel kanálu si může kanál zaregistrovat, což přináší řadu bezpečnostních výhod, například možnost zamykání témat i módů kanálu, či rozšířená moderátorská privilegia. Mezi další služby v balíku Anope patří *MemoServ*, *OperServ*, *BotServ* a *HostServ*.

Díky svojí jednoduchosti vytvořila řada vývojářů IRC boty pro různé účely, od textových her či napodobování reálného uživatele po krádež hesel či spamování.

1.8 Módy

Uživatelům IRC sítě je možné nastavit tzv. módy. Ačkoliv mohou ovlivňovat i jiné vlastnosti, jejich hlavní účel je rozdělení uživatelů na třídy podle práv. Označují se písmeny a některé z nich se dají navzájem kombinovat. Obecně platí, že každý server má trochu jiné módy, jelikož verzí IRCd vzniklo postupem času nespočet. Přihlášení registrovaní uživatelé mají mód (+r). Používá-li síť službu ChanServ, mohou být v kanálech uživatelé pověřeni jejich správou (nastavování téma, banování problémových uživatelů, atd.). Tato třída uživatelů se nazývá *operátoři* s módem +o. Ten jim je automaticky přidělen po vstupu do kanálu, jsou-li autentizováni NickServem (*příkaz IDENTIFY*) a mají-li v databázi toto právo nastaveno jiným uživatelem s odpovídajícími právy (jiný operátor či správce sítě). Operátorem kanálu se automaticky také stává, jak již bylo dříve zmíněno, jeho tvůrce. Protože nejde o základní funkci IRC, všichni operátoři se na kanálech označují jako jejich tvůrci - *CHANNEL FOUNDER*. Dále je tu také mód *HALF-OPERATOR*⁶ (+h, „poloviční operátor“). Uživatelé s tímto módem mají pouze některá práva operátorů, jako například vyhazování uživatelů či nastavování některých módů kanálů.

Běžný uživatel může sám sobě nastavit jen některé módy. K těm patří například mód *DEAF* (+D, „hluchý“, pouze u některých IRCd), který zamezí uživateli příchod příchozích zpráv (*příkaz PRIVMSG*). Nastavují se pomocí příkazu *MODE* (obr. 10 a obr. 11).

Módy mohou mít nejen uživatelé, ale také kanály. *INVITE_ONLY* (+i) například umožňuje přístup na kanál pouze uživatelům přizvaných pomocí příkazu *INVITE*. Pokud se kanál například nachází v módu *MODERATED* (+m), pouze uživatelé s určitými módy (+o, +h, +v⁷...) v něm mohou komunikovat.

⁶ Také známý jako „Halfop“.

⁷ Mód *VOICE* („hlas“) – jeho jediná funkce je ta, že uživatel s ním může komunikovat v kanálu s módem +m

JanVorisek	<code>/mode JanVorisek +D +i</code>
JanVorisek	<code>/mode JanVorisek +Di</code>

Obr. 10 - nastavení módů Deaf a Invisible v síti Rizon v programu HexChat

JanVorisek	<code>/mode JanVorisek -D -i</code>
JanVorisek	<code>/mode JanVorisek -Di</code>

Obr. 11 – odebrání dříve nastavených módů

1.9 Bezpečnostní rizika

Někdy může být veřejná IP adresa klienta viditelná skrz hostname. Některé IRC servery IP adresu maskují, což ovšem není možná při komunikaci s klienty pomocí DCC. Jedním z řešení tohoto problému je například využití tzv. *IRC Bounceru (BNC)*, tedy proxy serveru.

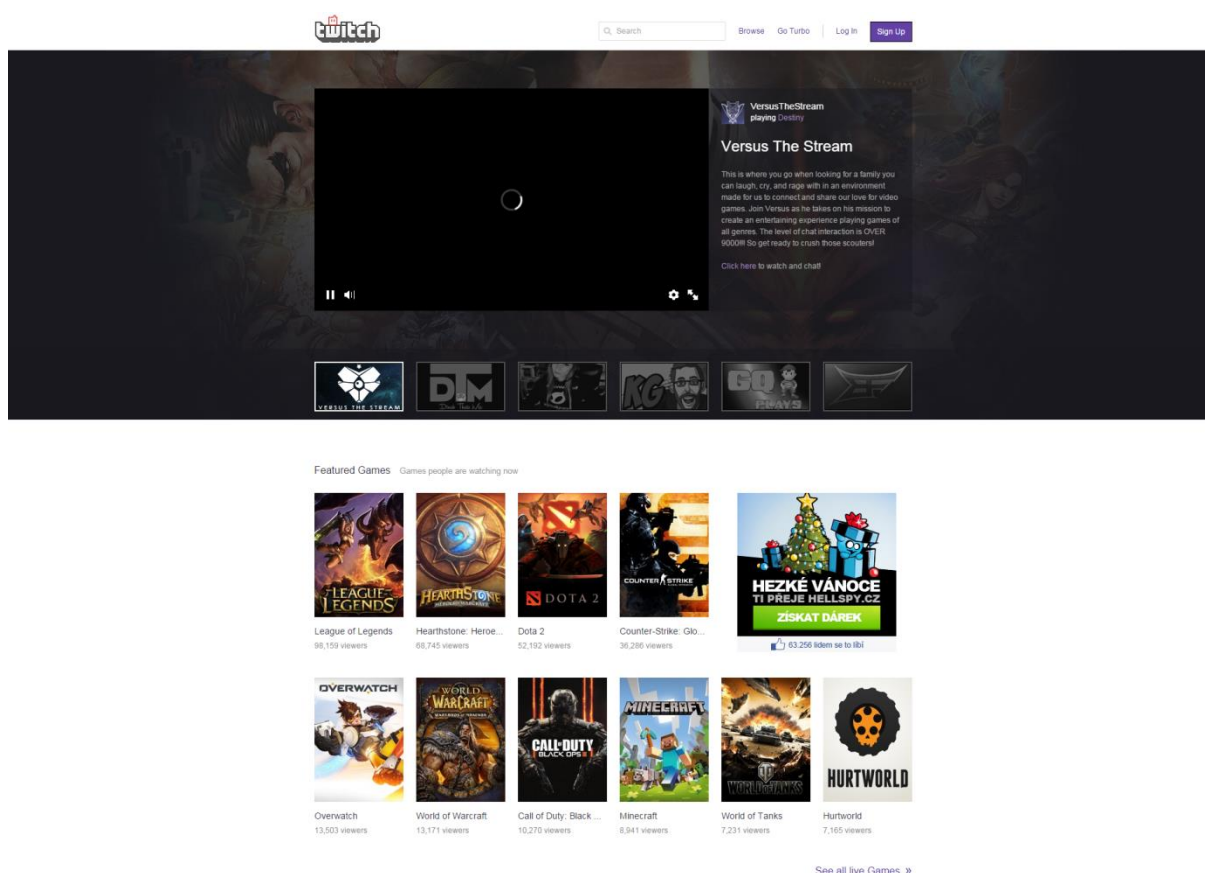
Jelikož je IRC protokolem, který za normálních podmínek posílá data v plaintext podobě přes nešifrované spojení, je získání citlivých údajů velmi jednoduché, například „nasloucháním“ odchozích datových paketů programem, jako je například WireShark.

Dalším často zmiňovaným problémem je napodobování jiných uživatelů, kdy neautorizovaný klient s přezdívkou jiného člověka může pomocí jednoduchého sociálního inženýrství získat citlivá data (třeba hesla). Proto je nutné být při komunikaci na IRC sítích velmi obezřetný.

2. Twitch

2.1 Základní informace

Twitch (na adrese www.twitch.tv, obr. 12) je americká livestreamovací⁸ video platforma vlastněná dceřinou společností Amazonu – Twitch Interactive, Inc. Zaměřuje se především na obsah týkající se videoher, v roce 2015 však přidala kategorii *Creative* určenou pro umění. Umožňuje jak sledování a vysílání živého videa, tak ukládání a export záznamů předešlých přenosů na službu YouTube.



Obr. 12 - domácí strana stránky Twitch.tv

Původně byla odnoží obecné streamovací platformy Justin.tv. Tu založili v roce 2007 mladí američtí podnikatelé a investoři Justin Kan a Emmett Shear. Dětila obsah do několika kategorií, z nichž se ta určená videohram stala nejpopulárnější. V červnu 2011 se společnost rozhodla založit novou platformu určenou výhradně pro herní obsah – Twitch. Justin.tv byla později v srpnu 2014 pro malou popularitu zrušena.

⁸ Livestreaming – živý přenos videa

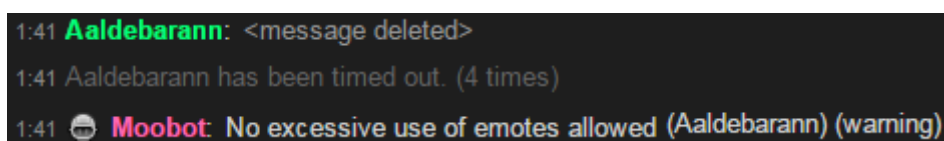
Dnes má Twitch přes 100 mil. návštěvníků za měsíc, více než 12 tisíc účtů využívajících partnerský program a průměrně 106 minut sledování videa na osobu denně. Celkově ji pro živé vysílání využívá přes 1.7 mil. *streamerů*. Twitch je dostupný jak v prohlížečích, tak i jako aplikace na platformách Android, iOS, X360, Xbox One, PS4 a Ouya. Vydělává především z reklam a uživatelského programu „Turbo“, který za určitý poplatek odstraní reklamy a přidá uživateli několik dalších výhod.

2.2 Kanály

Každý streamer má přidělený kanál – „hlavní“ stránku, na které mohou uživatelé živý přenos sledovat a chatovat s ostatními. Dá se do určité míry graficky upravit pomocí systému panelů. Ty se zobrazují pod oknem s přenosem a je možno do nich zadat například text, či vložit obrázek. Jsou využívány pro popis kanálu, pravidla chatu, odkazy na sociální média, seznam emotikon a podobně.

2.3 Chat

Twitch umožňuje uživatelům na kanálech komunikovat pomocí chatu umístěného vedle okna s videem. Dají se mu nastavit různá pravidla, jako jsou filtry slov, zakázání URL odkazů, limit odesílání zpráv apod. Streamer může jmenovat některé uživatele moderátory, kteří na chat dohlížejí. Zajímavé však je, že Twitch pro chat využívá právě protokol IRC, což přináší značné výhody, jako třeba možnost chatování pomocí běžného IRC klienta (softwaru). Je tedy také možné využití botů, například za účelem automatizovaného moderování chatu. Jednu z těchto služeb nabízí velmi populární *Moobot* (*obr. 13*), moderátorský bot, kterého si kdokoliv může zdarma propojit se svým kanálem a nastavit si ho dle vlastních představ.



Obr. 13 - Moobot dočasně banující uživatele za přehnané odesílání emotikon

Jelikož Twitch používá vlastní upravené IRCd, musí mít každý klient v IRC síti na platformě účet a při připojení je od něj vyžadována autorizace unikátním heslem (*OAuth*), které si může

nechat na stránkách služby vygenerovat. Autentizace se provádí již dříve zmíněným příkazem *PASS*. Při chatování v prohlížeči přímo na stránce kanálu je autorizace automatická. Využití IRC však přináší i svoje nevýhody – často se objevují tzv. *spambots*, kteří do chatů hromadně odesílají nebezpečné odkazy.

2.4 Twitch Partner Program

Uživatelé splňující určité požadavky (průměrně více než 500 diváků, streamování 3x týdně) se mohou stát partnery služby Twitch. Jeden z hlavních benefitů partnerství je tzv. *subscribe button* umožňující divákům stát se za určitý měsíční poplatek „*odběratel*“ kanálu (*subscribers*) a jejich oblíbeného streamera tak finančně podpořit. Odběratelé získají výhody jako například bonusové emotikony či možnost komunikovat v chatu i v módu *subscribers-only*, v němž běžní diváci komunikovat nemohou. Partneři také získávají část výtěžku z reklam přehrávaných na jejich kanále.

2.5 Streamování

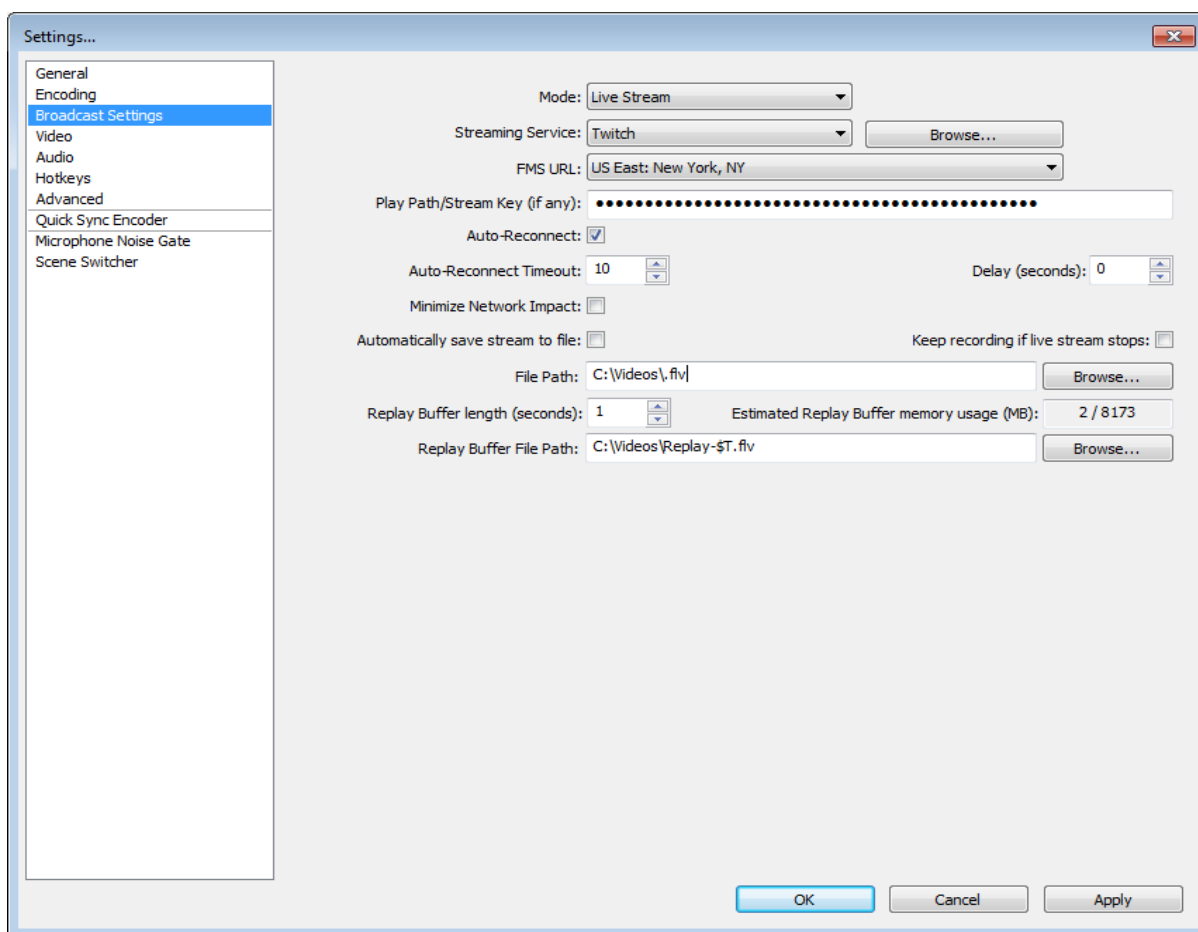
Ke streamování videa je potřeba tzv. stream key a streamovací software. Streamovací software (*broadcasting software*) zprostředkovává samotnou kompozici a kódování přenosu (video/audio) a jejich odeslání ingest serveru⁹ cílové platformy (v našem případě Twitch). Mezi nejpopulárnější patří komerční *XSplit*, open-source¹⁰ *OBS (Open Broadcaster Software)* a *FFSplit*. Streamovací klíč (*stream key*) je přidělen platformou každému účtu, jedná se o kód složený z písmen a číslic umožňující streamovat na odpovídající kanál. Zadává se do nastavení softwaru.

⁹ *Ingest server* -- Server zpracovávající příchozí audiovizuální data

¹⁰ *Open-source software* – Software se zdrojovým kódem volně dostupným veřejnosti

Pro nejlepší kompatibilitu s platformou Twitch se doporučuje pro streamování používat video kodek H.264 (či jeho volná alternativa X.264) v módu CBR (*constant bit rate*) s maximálním bitrate 3500 kb/s. Interval klíčových snímků by měl být nastaven na dvě sekundy.

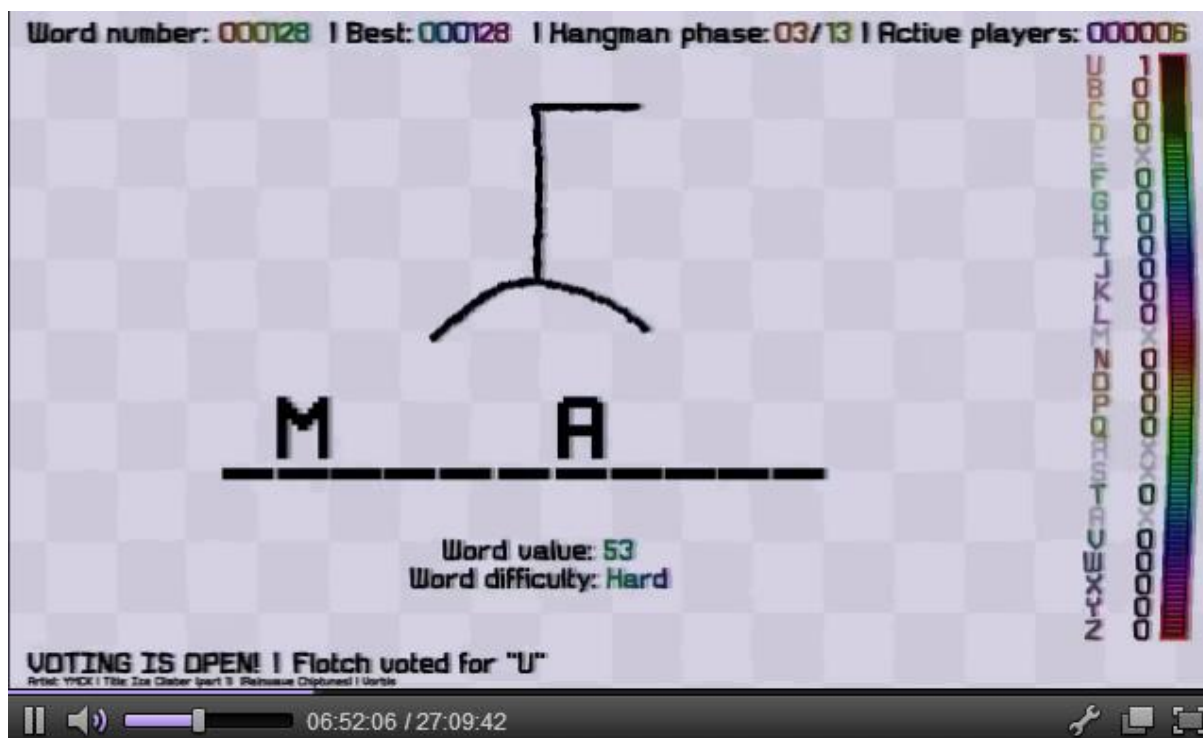
Pro audio je nejvhodnější použít formát AAC či MP3, přičemž maximální bitrate je 160 kb/s pro AAC a 128 kb/s pro MP3. Doporučená vzorkovací frekvence je 44.1 KHz pro MP3 a jakákoliv pro AAC. Latence přenosu je pro většinu uživatelů 5 až 20 sekund, záleží především na lokaci, volbě ingest serveru (*obr. 14*) a nastavení kódování přenosu.



Obr. 14 - menu nastavení OBS

2.6 Přehrávání

Služba je často kritizována za využívání platformy Flash pro přehrávač videa. 22. července 2015 byl přehrávač vylepšen, a ačkoliv se video stále přehrává pomocí Flashe (obr. 15), ovládací prvky využívají technologii HTML5. Přehrávač záznamů stále plně využívá původní verzi.

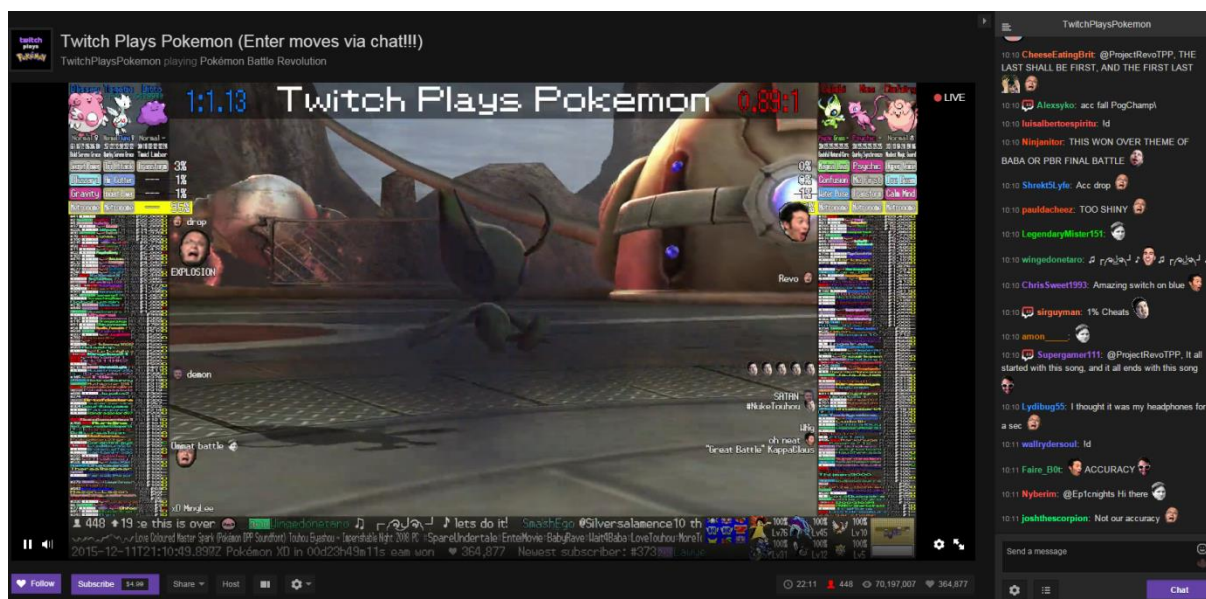


Obr. 15 - přehrávač záznamu

3. Twitch Plays Pokemon

Twitch Plays Pokemon je sociální experiment od neznámého australského programátora spuštěný 12. února 2014. Prvotní princip byl jednoduchý – chat se pokouší dohrát *Pokémon Red* na herní konzoli Game Boy od Nintendo pomocí emulátoru¹¹ ovládaného chatem. Právě fakt, že Twitch využívá pro svůj chat IRC protokol, toto umožnil. Autor pro tento účel vytvořil program v Pythonu, který interpretoval textové příkazy jako stisky kláves. Samotný přenos je kvůli limitacím platformy opožděn od chatu přibližně o 15 sekund.

Popularita kanálu TwitchPlaysPokemon (*obr. 16*) rychle vzrostla, především díky originalitě nápadu – průměrně jej sledovalo přes 80 tisíc lidí najednou, přičemž nejméně 10% z nich aktivně hrálo. Služba Twitch byla kvůli nečekané popularitě donucena přesunout TPP na svůj dedikovaný chat server.



Obr. 16 – kanál TwitchPlaysPokemon

Ze začátku program interpretoval všechny příchozí zprávy, ale s nárůstem popularity projektu bylo nutné systém předělat. 18. února 2014 tvůrce proto přidal hlasovací systém, jím nazvaný *Democracy* – každých 30 sekund byl programem zvolen příkaz s nejvíce hlasy a vykonán ve hře. Kvůli nátlaku komunity byl autor donucen přidat zpátky i starý systém, *Anarchy*. Nechal

¹¹ Program schopný napodobení hardwaru určitého zařízení za účelem spuštění softwaru vytvořeného pro jiný systém. (nejčastěji herních konzol)

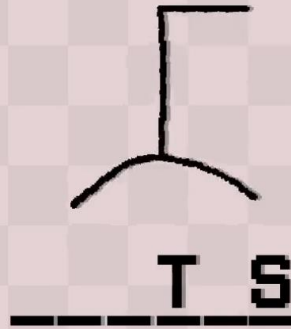
účastníky hlasovat, který systém chtějí využívat. Přepnutí na systém *Democracy* vyžadovalo více hlasů, než na mód *Anarchy*. V průběhu projektu se tedy hlasovací systém dynamicky měnil.

1. března 2014 komunita hru dohrála. Následovaly další díly ze série – *Pokémon Crystal*, *Emerald*, *FireRed*, *Platinum*, *HeartGold*, *Black*, *Black 2* (byla použita těžší, modifikovaná verze hry) , *X* a *Omega Ruby*. „Druhá série“ započala 12. února 2015, místo originálních her se však spíše zaměřovala na různě modifikované/pirátské verze. Dnes, v prosinci 2015, na kanále běží hra *Pokémon Battle Revolution*, kde hráči sázejí a vyhrávají body v automatizovaných bitvách.

Popularita TPP odstartovala novou vlnu projektů ovládaných tímto neortodoxním způsobem. Namátkou bychom mohli jmenovat například Twitch Plays Tetris, Twitch Plays Dark Souls či Twitch Plays Fallout. Objevily se ale i projekty, které se na videohry nezaměřovaly, jako například Twitch Installs Arch Linux, kde komunita pomocí zasílání příkazů do chatu instaluje distribuci GNU/Linux Arch Linux.



Word number: 000041 | Best: 000770 | Hangman phase: 03/13 | Active players: 000005



Word value: 34
Word difficulty: Normal

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Artist: Masatomo Miyamoto, Take Shant, Shin-kun, Pazu | Title: Ending (Rainyuse Chiotune) | Uorhis

Sdilet

1:33 8 446 54

Upravil panely

What is this?

Hi there! Welcome to Twitch Plays Hangman! A new experiment inspired by the well-known Twitch Plays Pokemon project.

What this basically is, is a hangman clone purposely developed to be controlled by the chat. Not only are there over 70K+ possible words, there is also leaderboard, so that you can score points on your own and become the very best!

This thing was developed by me, @Plokal, as a part of my seminar paper about this twitch-chat-game phenomenon.

Additional stuff

Word sources:
• [manythings.org](#) and [Ashley Bovan](#)

Thanks to:
• Guys from [itgame-maker](#) (Jiju, Ishaveve, Jack, Sidoraki) for helping me with testing. You guys are great.
• YoYoGames for making this possible!
• Everyone who plays, hosts or shares this!

How to play

The standard rules of Hangman apply here.

Everyone starts off with 500 points.

- Nothing is case-sensitive

The game uses a voting system to determine what letter the chat is going to guess.

To vote for a letter, simply send it in the chat. You know like this:

a

Only works during the voting period!

If you think you know the hidden word, you can use the !guess command to try your luck:

"!guess bet-amount word" or "!guess word"

You cannot guess the word without placing a bet. The lowest bet amount possible is the value of the current word displayed on screen. If you use the shorter version of this command, the game will automatically bet the above mentioned amount for you.

Example: !guess 123 cookie

Only works during the voting period!

If you use the !pts or !score command, the bot will tell you how many points you currently have.

You can use this command every 60 seconds.

FAQ

Q: For how long is this project gonna run?

A: For a couple of days now, then maybe from time to time.

Q: How did you make this?

A: I used **GameMaker: Studio**, it's a pretty neat tool for making all kinds of games and applications!

Q: Source?

A: I'll put it on GitHub as soon as I finish cleaning it up.

Obr. 17 – Kanál TwitchHangman

4. Praktická část – Twitch Plays Hangman

4.1 Prvotní podnět

Myšlenka tohoto fenoménu mne inspirovala ke zrealizování vlastního projektu (*obr. 17*). Mým záměrem bylo vytvořit projekt navrhnutý přímo pro využití s Twitch chatem, nikoliv „prostředníka“ mezi chatem a jiným programem.

4.2 Použitý software

4.2.1 GameMaker: Studio

Především kvůli svojí jednoduchosti jsem pro tvorbu projektu zvolil profesionální edici nástroje *GameMaker: Studio* od společnosti YoYo Games Ltd. Ačkoliv jde o nástroj určený primárně pro vývoj 2D videoher, je velmi flexibilní. Umožňuje export aplikací v podstatě na všechny populární platformy, včetně HTML5. IDE¹² GM:S je naprogramováno v Delphi. Podporuje vlastní externí rozšiřující moduly (*extensions*). Využívá objektový model, vývoj v něm probíhá v proprietárním C-like¹³ interpretovaným skriptovacím jazyku GML, či pomocí drag-n-drop systému ikoněk určeném především pro začínající uživatele.

Aplikace se dají exportovat dvěma způsoby – jako spustitelný soubor/aplikační balík cílové platformy obsahující bajtkód aplikace a interpret (*Runner*), nebo zkompileované obfuskované¹⁴ C++ (*YoYo Compiler*). Jedinou výjimkou je HTML5, pro které se GML převede na obfuskovaný JavaScript. Možnost exportu pro určitou platformu získáme zakoupením tzv. modulu. V základní verzi (která je zdarma) GM:S obsahuje pouze modul pro Windows.

Jeho předchůdcem byl program GameMaker od nizozemského matematického informatika a učitele Marka Overmarse. Ten v roce 1999 začal v Delphi vyvíjet nástroj pro tvorbu 2D animací zvaný Animo, který později změnil na nástroj pro vývoj 2D her. První veřejně

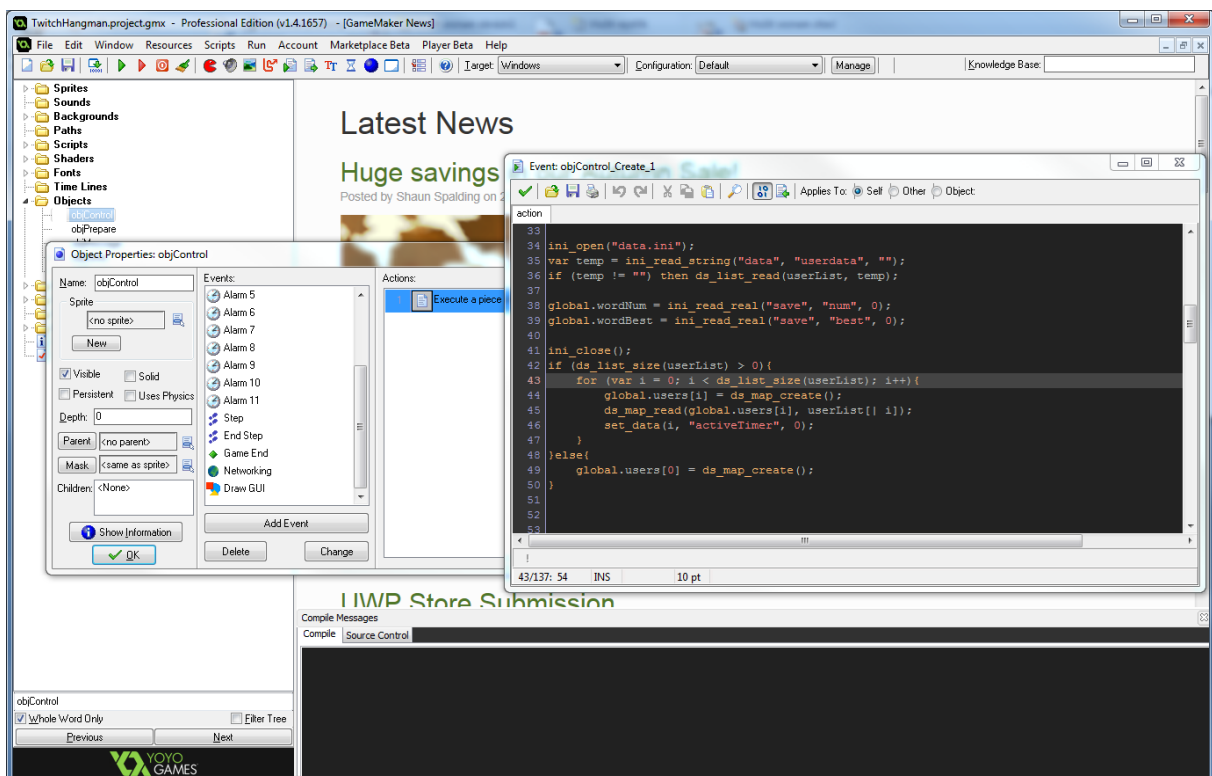
¹²*Integrated development environment* – grafické prostředí aplikace pro vývoj softwaru

¹³*C-Like* jazyk – programovací jazyk připomínající svou syntaxí jazyk C

¹⁴*Obfuscace* – proces, jehož účelem je učinit zdrojový kód co nejhůře čitelný pomocí technik jako odstranění formátování, komentářů, či přejmenování proměn a tříd

vydaná verze, 1.1, nyní již pod názvem GM, byla vydána v listopadu téhož roku. Od té doby se vyvíjel a 26. ledna 2007, o téměř osm let později, oznámil Overmars vstup do partnerství se skotskou, tehdy dvoučlennou, společností YoYoGames. V roce 2010 vyšla první verze GM pro Mac OSX s funkcemi dva roky staré verze GM – 7.0. Runner byl stejně jako IDE napsaný kompletně v Delphi, a protože nebyl kód ve spustitelných souborech nijak obfuskovaný, objevily se programy, které z něj dokázaly zdrojový kód aplikace sestavit.

22. května 2011 vyšel GameMaker: Studio 1.0 – zcela nová verze GM podporující export na širší škálu platform a funkce jako nativní verzování či implementaci 2D fyzikálního enginu Box2D. Funkce IDE (obr. 18) byly značně vylepšeny a systém exportu aplikací byl přepsán z Delphi na ten, který známe dnes.

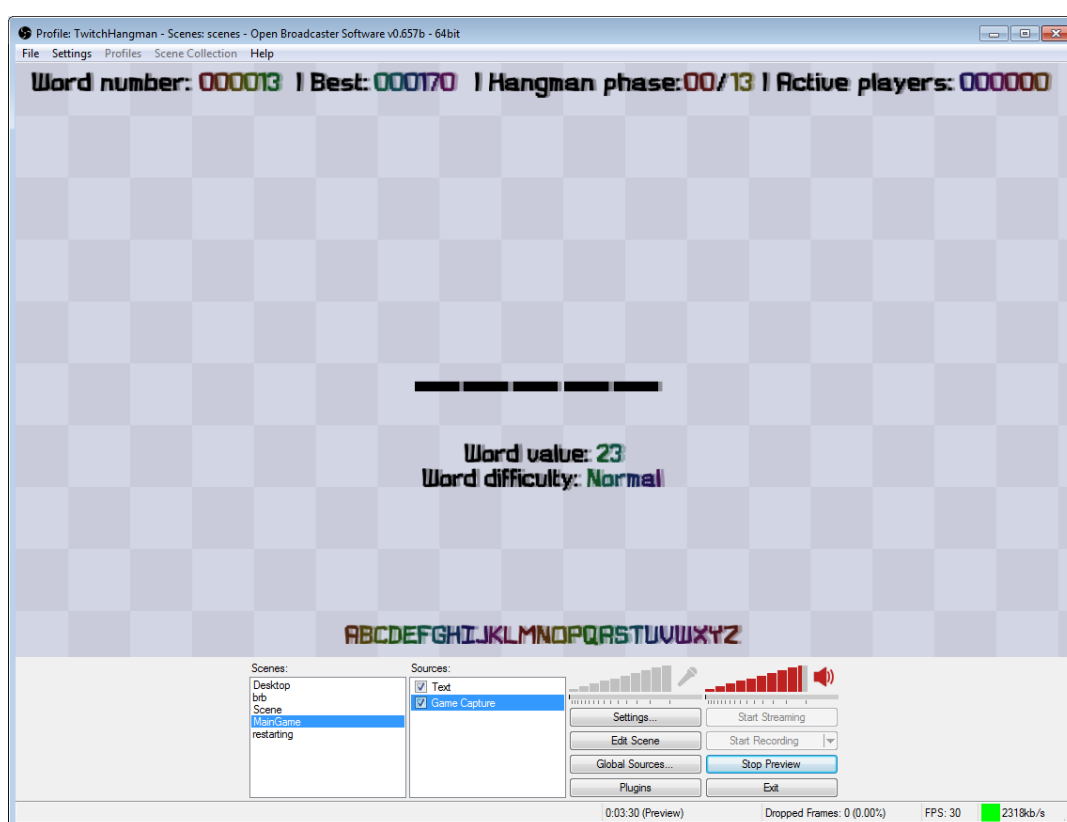


Obr. 18 - IDE programu GameMaker: Studio

2. července 2014 společnost oznámila spuštění webové služby *GameMaker: Marketplace* – obchodu, kde kdokoliv vlastníci profesionální edici GM:S může prodat svoje výtvořky pro použití v projektech (grafika, hudba, skripty, rozšiřující moduly...) či nabídnout služby jako programování či tvorba grafiky. 16. února 2015 společnost odkoupila korporace Playtech za 16.4 milionů USD. Dnes zaměstnává více než 25 lidí.

4.2.2 Open Broadcaster Software

O streamování videa se postaral již dříve zmíněný *Open Broadcaster Software*, který je velmi jednoduchý na použití, velmi dobrá open-source alternativa komerčních programů jako XSplit s vlastním API¹⁵ a podporou Plug-inů. Aktuální verze, 0.657b, je výsledkem práce více než 40 nezávislých vývojářů. Existují dvě verze programu – *OBS* (původní, obr. 19) a *OBS MultiPlatform* (nová verze s lepším API a podporou Windows, OSX 10 a GNU/Linux).



Obr. 19 - prostředí programu Open Broadcaster Software

4.3 Vývoj

Vývoj projektu začal v létě 2015, původně šlo o „kreslicí“ program, ve kterém uživatelé mohli pomocí příkazů vytvářet geometrické útvary, či psát text. Postupem času jsem však došel k závěru, že tento koncept, ačkoliv funkční, jednoduše není zábavný. Proto jsem se na podzim 2015 rozhodl projekt předělat na vylepšený klon klasické hry oběšenec. Jako hlavní

¹⁵ *Application Programming Interface* – rozhraní pro programování aplikací, většinou ve formě knihovny

motivační prvek pro hráče jsem do hry zakomponoval systém nejlepších skóre - kdokoliv může pro sebe získávat body buď hlasováním pro správná písmena nebo sázením bodů na slova. Data hráčů jsou ukládána do lokální databáze, přičemž systém jsem navrhl tak, aby se záznam o hráči vytvořil, jakmile odeslal první zprávu do chatu.

```
client = network_create_socket(network_socket_tcp);
var network = network_connect_raw(client, "irc.twitch.tv", 6667); //Establish network connection
if (network >= 0){
    show_debug_message("Connection successful.");
}else{
    show_debug_message("Connection failed.");
    game_restart();
}

sendMessage("PASS oauth:" + BROADCASTER_OAUTH, 1); //PASS
sendMessage("USER " + BROADCASTER_NAME + " 0 * :" + BROADCASTER_NAME, 1); //USER
sendMessage("NICK " + BROADCASTER_NAME, 1); //NICK
sendMessage("JOIN " + CHANNEL_NAME, 1); //Join our channel...
sendMessage("CAP REQ :twitch.tv/tags", 1); //IRCv3 tag support
```

Obr. 20 – připojení k síti a přihlášení

```
var message = argument[0];
if (argument_count > 1){
    if (argument[1] == 1){
        message += chr(13) + chr(10);
    }
    if (argument[1] == 2){
        message = "PRIVMSG " + CHANNEL_NAME + " :" + message + chr(13) + chr(10);
    }
    if (argument[1] == 3){
        message = "PRIVMSG " + CHANNEL_NAME + " :/me " + message + chr(13) + chr(10);
    }
}

var tempBuffer = buffer_create(string_length(message), buffer_grow, 1);
buffer_seek(tempBuffer, buffer_seek_start, 0);
buffer_write(tempBuffer, buffer_string, message);
network_send_raw(client, tempBuffer, buffer_get_size(tempBuffer));
buffer_delete(tempBuffer);
```

Obr. 21 – metoda pro zaslání zpráv IRC síti

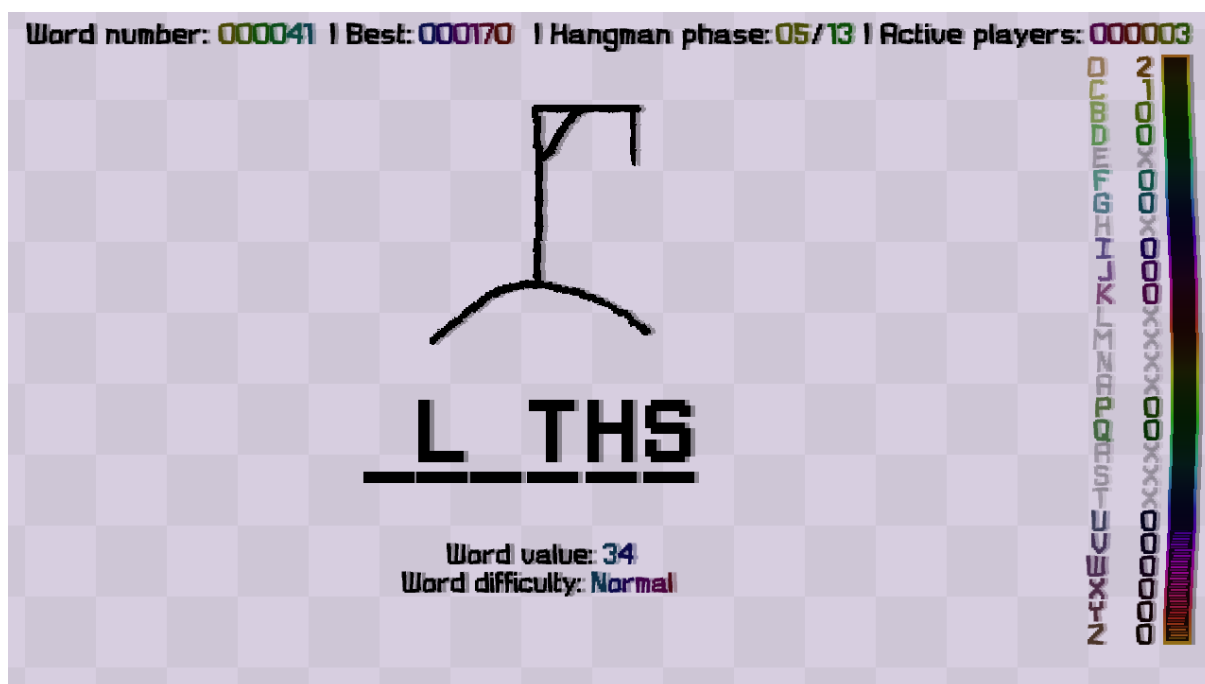
4.3.1 Postup hry

Všichni hráči začínají s 500 body, cílem komunity je uhodnout co nejvíce slov bez prohry. Kolo začíná náhodným výběrem slova ze dvou seznamů slov – lehčích a těžších. První seznam byl sestaven se slov na stránkách ManyThings¹⁶, druhý byl poupravenou verzí seznamu 97 tisíc podstatných jmen od Ashley Bovan¹⁷.

¹⁶ Dostupné na <http://www.manythings.org/>

¹⁷ Dostupné na <http://www.ashley-bovan.co.uk/words/partsofspeech.html>

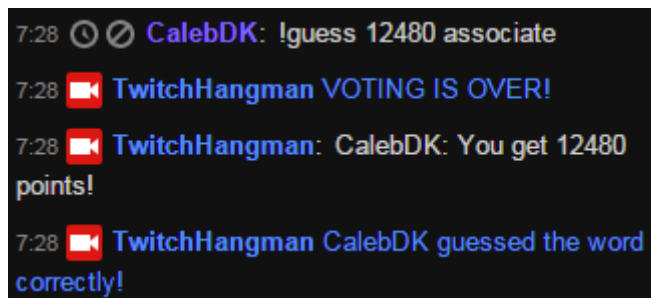
Počáteční fáze oběšence je 0. Poté začíná fáze hlasování, kdy hráči mohou pomocí chatu hlasovat pro písmeno, které bude program hádat, či přímo tipovat celá slova. První příchozí hlas spustí třicetisekundový časovač (obr. 20), po jehož vypršení hra hádá písmeno s nejvíce hlasy. Pokud slovo písmeno obsahuje, odkryje se a všichni hráči, co pro něj hlasovali, obdrží body. V opačném případě se hráčům odečte pouze hodnota písmena¹⁸ a oběšenci se přidá fáze. Hodnota slova je uvedena na obrazovce a určuje minimální možnou sázku při tipování slova.



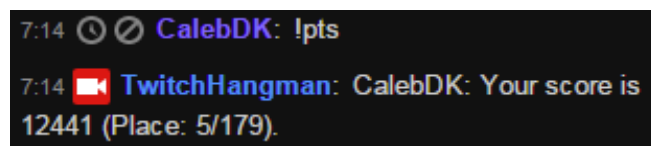
Obr. 22 - hlasování

Během hlasování mohou hráči také zkusit uhodnout slovo příkazem „!guess“ (obr. 21). Nelze hádat bez vsazení bodů. První parametr příkazu je sázka (pokud není uvedena, vsadí se hodnota slova), druhý parametr je slovo samotné. Odeslání příkazu „!pts“ vypíše aktuální skóre a pozici hráče, který jej odeslal (obr. 22).

¹⁸ Hodnoty písmen jsou následující: Q - 55; J - 53; X - 38; Z - 32; W - 14; V - 12; K - 11; F - 9; Y - 7; B - 6; G - 4; H - 4; D - 4; P - 4; M - 4; U - 3; C - 3; L - 2; T - 2; N - 2; O - 2; R - 1; A - 1; I - 1; S - 1; E - 1



Obr. 23 - hráč hádající slovo



Obr. 24 - vypsání skóre po odeslání příkazu „!pts“

Kolo končí, je-li splněna jedna z následujících podmínek:

- a) Oběšenec dosáhne třinácté fáze
- b) Jeden z hráčů uhodne slovo
- c) Celé slovo je postupně odkryto

Po skončení kola se zobrazí tabulka nejlepších výsledků ve hře (*obr. 23*) i v chatu, poté následuje restart.

Word number: 000026 | Best: 000170 | Hangman phase: 07/13 | Active players: 000005

1. mrfahrenheit94	01025010
2. TheRealGravvy	00364817
3. LegacyCrono	00017649
4. EmiliusTheAwesome	00004153
5. tehwave	00003989
6. Peping01	00003565
7. WongCR	00003512
8. Bakufreak	00002005
9. Tealgamemaster	00001271
10. Cytidine	00001248
11. UltimaN3rd	00001219
12. SuperBashSisters	00001152
13. CalebDK	00001110
14. Dusk208	00001004
15. Antfacekilla	00000874

Obr. 25 - tabulka výsledků

4.3.2 Grafika

Z důvodu komprese videa jsem zvolil jednoduchý grafický styl, který by vypadal dobře i při nižším bitrate. Veškeré herní elementy, kromě pozadí a postavičky, jsou zobrazeny jako text (použil jsem mé vlastní písmo). Z estetického hlediska jsem se nejvíce soustředil na plynulé přechody a animace. Jediným statickým prvkem na obrazovce jsou informace o hře – číslo slova, rekord, fáze oběšence a počet aktivních hráčů.

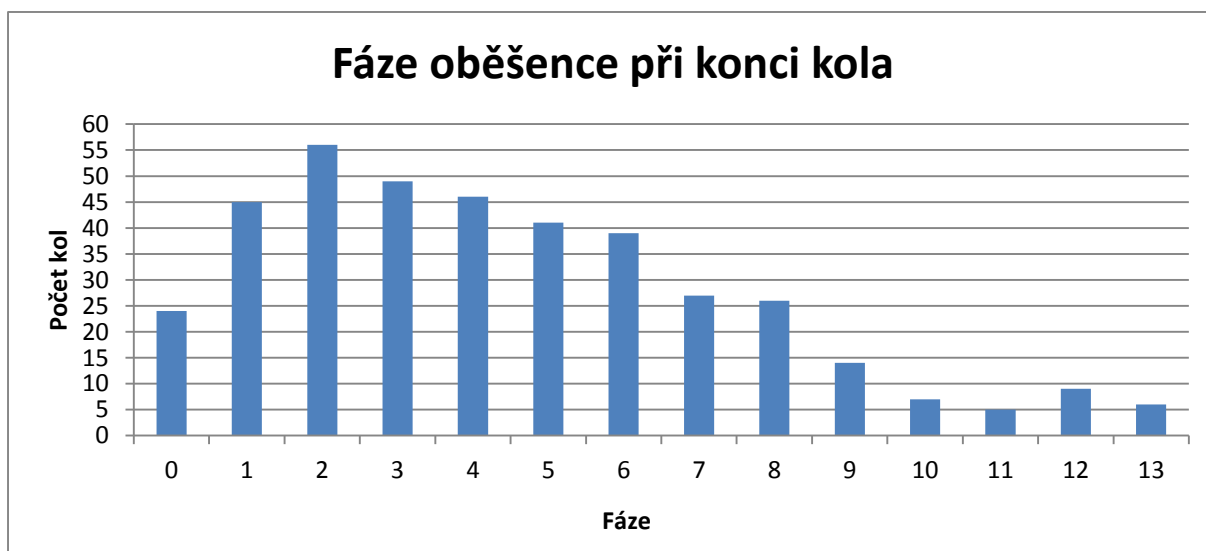
4.4 Sběr dat

Každou hodinu program zaznamenával počet aktivních hráčů, počty hlasů pro každé písmeno a informace o kolech v ní dokončených: obtížnost, první hádané písmeno, fáze oběšence, koncové stádium (prohra, uhodnutí slova, uhodnutí písmen), čas, hodnota a hádané slovo (obr. 24). Data byla zaznamenávána do konfiguračních souborů (*.ini) v oddělených složkách. Data o sledovanosti byla poskytnuta službou Twitch.



Obr. 29 - graf porovnání celkového počtu hlasů oproti prvním hlasům kola

Z grafu (obr. 27) můžeme vypořadovat, že písmena, která obdržela první hlas, byla často těmi, která byla zvolena programem. Příčinou tohoto úkazu je především fakt, že většinu času hru hráli méně než tři diváci.



Obr. 30 - graf fází oběšence při konci kola

Malá obtížnost hry měla za následek, že jen minimum kol se dostalo do posledních fází (obr. 28). Většina skončila při druhé.

4.5.2 Výsledky jednotlivých hráčů

1. bigfriendlygames	364222030
2. CalebDK	51250742
3. EmiliusTheAwesome	5567133
4. mrfahrenheit94	1025015
5. TheRealGravvy	364867
6. LegacyCrono	17649
7. boksyS	11357
8. Minwene	9844
9. Peping01	8811
10. Bakufreak	7678
11. tehwave	3989
12. UltimaN3rd	3112
13. Daethcore_Seth	2248
14. Tealgamemaster	1298
15. Cytidine	1248
16. SuperBashSisters	1243
17. Dusk208	1004
18. Antfacekilla	874
19. Pk_fms	769
20. friedchicken009	758

Obr. 31 - tabulka dvaceti nejlepších hráčů

Z tabulky (obr. 29) je jasně vidět, že rozdíl v bodech mezi prvním a dvacátým místem je velmi velký, zvláště vezmeme-li v potaz fakt, že každý hráč začíná s 500 body. Polovina hráčů skončila s 500 či méně body. Za velký rozdíl ve finálním skóre může především špatně navržený hlasovací systém, konkrétně v tom, že neexistoval žádný limit na maximální sázku. Tento problém se projevil především tehdy, kdy se účastnil velmi nízký počet lidí.

5. Závěr

V průběhu krátké testovací doby, během níž nedošlo k žádným větším technickým potížím, získal můj projekt jen omezený počet uživatelů, kteří se však opakovaně na kanál vraceli. Díky tomu se mi podařilo získat dostatek dat ke zpracování a splnit tak svůj původní záměr. Chyby spojené s nevyvážeností hry byly později opraveny.

Jelikož o zdrojový kód projektu projevilo několik lidí zájem, rozhodl jsem se jej vydat (dostupný na <https://goo.gl/Tn58ZW>). Jsem přesvědčen, že poslouží jako skvělý příklad pro vývojáře zajímající se o tvorbu programů ovládaných pomocí IRC protokolu.

Po dokončení experimentu jsem byl mimo jiné kontaktován komunitním manažerem YoYoGames s prosbou o poskytnutí videozáznamu z jeho průběhu. Touto firmou bylo 14. 3. 2016 vydáno oficiální propagační video „2016 Made with GameMaker Showreel“, jehož je můj projekt součástí.

Při tvorbě této práce jsem získal množství cenných zkušeností - především s prací s vlastními databázemi a se zpracováváním komplexních textových řetězců a dat. Mimo jiné jsem také rozšířil svoje znalosti IRC protokolu a internetových sítí.

6. Zdroje

6.1 Bibliografie

1. Alex Charalabidis, The Book of IRC: The Ultimate Guide to Internet Relay Chat, No Starch Press, 1999, ISBN: 978-1886411296

6.2 Internetové zdroje

1. RFC 1459 - Internet Relay Chat Protocol <<https://tools.ietf.org/html/rfc1459>>
2. RFC 1459 - Internet Relay Chat: Client Protocol <<https://tools.ietf.org/html/rfc1459>>
3. Anope IRC Services <<https://www.anope.org/about.php>>
4. The IRC Prelude <<http://www.irchelp.org/irchelp/new2irc.html>>
5. ChanServ – AnopeWiki <<https://wiki.anope.org/index.php/ChanServ>>
6. NickServ – AnopeWiki <<https://wiki.anope.org/index.php/ChanServ>>
7. IRCv3.1 Client Capability Negotiation
<<http://ircv3.net/specs/core/capabilitynegotiation-3.1.html>>
8. IRC Networks – Top 100 <<http://irc.netsplit.de/networks/top100.php>>
9. Internet Relay Chat – Wikipedia, the free encyclopedia
<https://en.wikipedia.org/wiki/Internet_Relay_Chat>
10. List of Internet Relay Chat Commands – Wikipedia, the free encyclopedia
<https://en.wikipedia.org/wiki/List_of_Internet_Relay_Chat_commands>
11. Client-to-client Protocol – Wikipedia, the free encyclopedia
<https://en.wikipedia.org/wiki/Client-to-client_protocol>
12. Direct Client-to-Client – Wikipedia, the free encyclopedia
<https://en.wikipedia.org/wiki/Direct_Client-to-Client>
13. About – Twitch <<http://www.twitch.tv/p/about>>
14. Twitch | Broadcast Requirements
<<http://help.twitch.tv/customer/portal/articles/1253460>>
15. Twitch – Partner Application <<http://www.twitch.tv/partner/signup>>
16. Moobot, your Twitch chat moderation bot <<http://twitch.moobot.tv/>>
17. Twitch In the Shell <<http://twitchintheshell.com/>>
18. The Official Twitch Blog – Recent Chat Issues and TwitchPlaysPokemon
<<http://blog.twitch.tv/2014/02/recent-chat-issues-and-twitchplayspokemon/>>

19. The Official Twitch Blog – Video Player Controls Now in HTML!
<<http://blog.twitch.tv/2015/07/video-player-controls-now-in-html/>>
20. Twitch.tv – Wikipedia, the free encyclopedia
<<https://en.wikipedia.org/wiki/Twitch.tv>>
21. Justin.tv – Wikipedia, the free encyclopedia <<https://en.wikipedia.org/wiki/Twitch.tv>>
22. GameMaker: Studio v1.4 Early Access is here! Introducing the GameMaker: Marketplace! | News | YoYoGames <<https://www.yoyogames.com/news/229>>
23. GameMaker: Studio – Wikipedia, the free encyclopedia
<https://en.wikipedia.org/wiki/GameMaker:_Studio>
24. ManyThings <<http://www.manythings.org/>>
25. List of Verbs, List of Nouns, List of Adjectives <<http://www.ashleybovan.co.uk/words/partsofspeech.html>>

6.3 Zdroje obrázků

1. <https://hexchat.github.io/> - screenshot