

Vytvořte systém pro interaktivní ovládání PC hry pomocí chytrého telefonu. PC i telefony budou připojeny ve stejné lokální síti. Využijte interaktivní možnosti chytrých telefonů (vibrace, pohybový senzor, displej ...). Navrhněte a implementujte síťové rozhraní pro komunikaci mezi PC a telefony. Vytvořené řešení demonstруйте na jednoduché PC hře. Síťové rozhraní i výslednou hru řádně otestujte a zdokumentujte.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAROVÉHO INŽENÝRSTVÍ



Bakalářská práce

Interaktivní ovládání PC hry pomocí chytrého telefonu

Marek Foltýn

Vedoucí práce: Ing. Filip Kříkava, Ph.D.

16. května 2016

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Filipu Křikavovi, Ph.D., za pomoc a příkladné vedení. Dále pak své manželce Veronice Foltýnové za trpělivost a ochotu vytvářet prostředí vhodné k tvorbě bakalářské práce, rodičům a celé mé rodině za velkou podporu ve všech směrech. Děkuji také všem, kdo se podíleli na testování hratelnosti hry.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 16. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Marek Foltýn. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Foltýn, Marek. *Interaktivní ovládání PC hry pomocí chytrého telefonu*. Bachelářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato bakalářská práce se zabývá tvorbou systému pro ovládání PC hry pomocí mobilního telefonu. Hlavním cílem je obohacení herního zážitku pomocí interaktivních prvků, které jsou na mobilních telefonech k dispozici. Součástí práce je analýza způsobů ovládání her, přehled interaktivních technologií v mobilních telefonech a samotná tvorba komunikačního systému, který je demonstrován na jednoduché hře.

Klíčová slova interaktivní ovládání, počítačové hry, smartphone, komunikační systém, Cocos2d-x, RakNet

Abstract

The main purpose of this thesis is to create an interactive PC game control system using smartphones in order to enhance the game experience. The thesis contains the analysis of different ways how a PC game can be controlled, the overview of interactive mobile technologies and also the communication system implementation, which is demonstrated in a simple game.

Keywords interactive control, computer games, smartphone, communication system, Cocos2d-x, RakNet

Obsah

Úvod	1
1 Ovládání her	3
1.1 Historie ovladačů	3
1.2 Klávesnice	4
1.3 Myš	4
1.4 Gamepad	5
1.5 Joystick a volant	5
1.6 Dotyková obrazovka	5
1.7 Ovládání pohybem	6
1.8 Srovnání	6
1.9 Shrnutí	7
2 Interaktivní prvky v mobilních telefonech	9
2.1 Dotykový displej	9
2.2 Akcelerometr	11
2.3 Gyroskop	11
2.4 Senzor přiblížení	11
2.5 Ostatní senzory	12
2.6 Shrnutí	12
3 Podobná řešení	13
3.1 Telefon jako univerzální gamepad	13
3.2 Apple AirPlay	14
3.3 AirConsole	15
3.4 Možnosti řešení	16
4 Tvorba systému	17
4.1 Požadavky	17
4.2 Návrh	20

4.3 Implementace	31
4.4 Testování	38
4.5 Dokumentace	41
4.6 Možnosti rozšíření	41
4.7 Využití v dalších hrách	42
Závěr	43
Literatura	45
A Obsah přiloženého CD	49

Seznam obrázků

1.1	Replika ovladače pro hru <i>Tennis for Two</i> [1]	3
2.1	Využití dotykové obrazovky ve hře Fruit ninja [2]	10
3.1	Mobilní telefon jako univerzální gamepad[3]	13
3.2	Technologie AirPlay [4]	14
3.3	Ukázka hry pomocí systému AirConsole [5]	15
4.1	Struktura síťového modulu	21
4.2	Architektura hry	27
4.3	Události přechodů mezi obrazovkami	28
4.4	Komunikační stavy hry a ovladače	29
4.5	Sdílení částí aplikace mezi hrou a ovladačem	32
4.6	Připojení nového hráče	33
4.7	Zobrazení fyzikálních objektů a detektoru gólu	34
4.8	Přesahy hřiště při různých poměrech stran	35
4.9	Trhané vykreslování hráče	36
4.10	Vykreslování hráče s predikcí pohybu	36
4.11	Vykreslování s predikcí pohybu a plynulou korekcí	37
4.12	Testování hrátelnosti	40
4.13	Oprava okrajů hřiště	41

Seznam tabulek

1.1	Srovnání herních ovladačů	7
4.1	Srovnání vybraných herních žánrů	23
4.2	Rozdělení herních handlerů	26

Úvod

Videohry existují od počátku prvních počítačů [6]. Jejich možnosti jsou úzce spjaty s vývojem výpočetního a grafického výkonu, hardwaru a dalších souvisejících technologií. Neustálé zmenšování součástí v současné době nabízí vysoký výkon ve velmi malých strojích [7]. Na moderní typy hardwaru se adaptovaly i hry. Jsou vyvíjeny pro přenosná zařízení, jako například mobilní telefony nebo tablety. Lidé tak mohou hrát doslova kdekoli. Nemusí být doma, jako je tomu u klasických počítačových a konzolových her, ale mohou hrát v čekárně, ve škole o přestávce, na cestě a podobně.

Mobilní zařízení jsou v dnešní době velmi úzce propojená s lidmi a jejich životem. Na základě toho se nabízí otázka, zda by tyto nové možnosti mohly nějakým způsobem obohatit herní interakci mezi lidmi. Existuje několik směrů, které se o tento cíl snaží:

Způsobem, jak dosáhnout lepší mezilidské interakce ve hře, je hra více hráčů, tzv. *multiplayer*. Jedná se o typ, kdy na sebe v herním světě reaguje více hráčů, ať už v reálném čase, nebo pomocí sdílení pokroku ve hře (např. žebříčky s výsledným skóre). Obě tyto varianty sice poskytují informace o hře ostatních lidí, avšak ve spojení s internetem nemají zásadní vliv na mezilidskou komunikaci: spíše vylepšují vlastnosti herního světa, aby zlepšily hráčův zážitek. Jako příklad uvedu běžného hráče počítačové hry, který hraje doma v pokoji multiplayerovou online hru přes internet: Sice se pohybuje ve stejném virtuálním světě jako další hráči, ale vliv na mezilidskou komunikaci je minimální. Dalším příkladem může být herní interakce pomocí sociálních sítí. Kromě svého pokroku ve hře vidí hráč i pokrok ostatních lidí a může ovlivnit jejich hru. Opět se ale nejedná o sociální interakci ve smyslu fyzické přítomnosti. Virtuální realita zde tvoří pouze prostor, skrz který ostatní hráči sledují jeden druhého.

I přes tyto nevýhody existují způsoby, jak spojit virtuální realitu ve hře s širší mezilidskou komunikací. Jedním z nich je hra v lokální síti, kdy jsou hráči fyzicky přítomni na jednom místě (pokoj, garáž). Hráči prožívají stejný herní zážitek jako v případě hry přes internet, avšak umocněný přítomností

ostatních hráčů. Reakce během hry pak mohou obsahovat prvky komunikace typické pro mezilidský kontakt ve fyzické přítomnosti: zvuk, vizuální a fyzický kontakt a podobně.

Dalším způsobem je hra na jednom PC nebo konzoli. Pro tento způsob je typické sdílení jedné obrazovky (monitor, televize) a případně ovládacích periférií (klávesnice). Herní svět pak netvoří virtuální *bariéru*, přes kterou hráč vnímá ostatní lidi, ale stává se součástí prostředí společně s lidmi, podobně jako například stolní hry.

Hraní na sdílené obrazovce a v lokální síti se stalo hlavní motivací pro tvorbu této práce. Zajímá mě využití her jako obohacení zážitku ze společenské akce¹.

Jedním ze způsobů, jak více provázat hru s realitou, může být využití mobilních telefonů. Dnešní mobilní telefony v sobě obsahují velké množství senzorů, které lze použít pro interaktivní ovládání (dotykový displej, akcelerometr, gyroskop a další senzory okolního prostředí), proto se jeví jako vhodné zařízení k ovládání hry. Pokud spojíme multiplayerovou hru s jednou hlavní obrazovkou (televize nebo monitor) a ovládáním pomocí chytrého telefonu, vzniknou nové možnosti, jak propojit reálný svět s hrou. Hráči mohou kromě klasického ovládání využít vlastní displej společně s ostatními interaktivními prvky. Tím se herní pole rozšíří od hlavní obrazovky na další části prostoru.

Cíl projektu

V této práci se budu zabývat zkoumáním možností, jak využít interaktivní prvky mobilních telefonů pro ovládání počítačové hry. Mobilní zařízení tedy nebudou sloužit jen jako samostatné herní konzole, ale ani jako simulace periférií typu myš nebo gamepad. Budou tvořit jednotný celek společně se samostatným počítačem. Tuto myšlenku se budu snažit demonstrovat vytvořením systému komunikace mezi telefony a počítačem a jeho použitím v jednoduché hře. Výsledný produkt bude tvořit komunikační systém použitelný při vývoji libovolné hry a dvě aplikace (jedna pro mobilní platformu a druhá pro osobní počítač), které využívají tento systém.

¹Společenskou akcí je myšlena jakákoli vhodná příležitost trávení času s více lidmi. Může to být například návštěva přátel, večírek nebo volná hodina mezi vyučováním ve škole.

Ovládání her

V první kapitole se budu nejprve zabývat problematikou ovládání her v současné době, a to jak na počítačích, tak na ostatních zařízeních. Stručně se zmíním o historii ovladačů a poté popíšu nejpoužívanější z nich. Z tohoto přehledu budu vycházet během srovnávání alternativních ovládacích prvků v mobilních zařízeních a jejich přínosů. V této kapitole se nebudu podrobně zabývat softwarovým návrhem uživatelského ovládání, ale spíše využitím hardwaru pro herní účely. Hlavní náplní bude srovnání několika rozdílných způsobů ovládání, jejich výhod a nedostatků.

1.1 Historie ovladačů

Počítačové hry se začaly objevovat v 50. letech 20. století. Hardware dostupný v této době nebyl určený pro herní zaměření, např. známá hra *Tennis for Two* vytvořená Williamem Higinbothamem v roce 1958 v národní laboratoři v Brookhavenu využívala osciloskop jako grafický displej pro zobrazení primitivní dvourozměrné hry. Jako ovládání sloužilo jedno tlačítko a otočný regulátor [1].



Obrázek 1.1: Replika ovladače pro hru *Tennis for Two*[1]

Větší rozvoj herního hardwaru začal v roce 1971. Byl vytvořen první herní automat na mince s hrou *Galaxy Game*. Jednalo se o hru pro dva hráče ovládanou jednoduchým joystickem. Později se začaly objevovat další automaty a využívat jak tlačítka, tak i joystick nebo volant. Hry se v této době vyskytovaly především na herních konzolích. Vzhledem k narůstajícím prodejům osobních PC se však začaly dostávat i do této oblasti, kde byla nejrozšířenější periferií klávesnice. Vývoj konzolí se však nezastavil.

Převrat v ovládání PC, a to i v herním průmyslu, způsobil příchod počítačové myši, jak ji známe dnes. Mnoho herních konceptů bylo vylepšeno a pro hráče to představovalo větší pohodlí během hraní. Kromě toho se vyvíjely alternativní typy ovládání, některé úzce spojeny s herním žánrem (joystick, volant), nebo naopak vhodné pro mnoho typů her (gamepad) [1].

Kromě klasických ovladačů se objevují také snahy o přirozenější ovládání pomocí pohybu, tzv. *motion sensing*. Vzniklo proto několik ovladačů a herních konzolí, z nichž nejvýznamnější je Microsoft Kinect vzniklý v roce 2010 [8]. Více se problematice ovládání pohybem věnuje sekce 1.7.2.

V současné době se využívá mnoho různých typů ovládání. Následující kapitoly budou věnovány těm nejvíce rozšířeným především v osobních počítačích. Vysvětlím, k čemu se dají vhodně využít a jaké mají nedostatky. Následující přehled bude zároveň sloužit jako podklad pro analýzu v praktické části práce.

1.2 Klávesnice

Klávesnice představuje nepoužívanější počítačovou periférii vůbec. Její princip je velmi jednoduchý: každé stisknutí či uvolnění tlačítka způsobí odeslání informace do PC. Je možné detekovat události více tlačítek najednou, čehož využívají klávesové zkratky.

V herním průmyslu se klávesnice využívá v drtivé většině herních žánrů: od jednoduchých arkád, přes závody až po strategie. Jsou vhodné, pokud je potřeba rozlišit větší množství vstupů, které reprezentují jednotlivé klávesy.

Klávesnice ale nemusí být vždy ideální volbou. Diskrétní zpracování vstupu (stisknuto, nestisknuto) představuje nepohodlí při ovládání závodní hry: v zatáčení je zhoršená citlivost, auto buď zatáčí naplno, nebo vůbec. Tento nedostatek se vývojáři snaží řešit postupným natáčením kol, ale ani to není ideální. Při zatočení v menší zatáčce je nutné přerušovaně uvolňovat klávesu, aby se vytvořila iluze mírně natočeného volantu. V kombinaci s myší může být nevýhodou horší dostupnost kláves vzdálenějších od levé ruky.

1.3 Myš

Počítačová myš je druh polohovacího zařízení. Optický či laserový snímač detekuje pohyb myši po podložce a převede jej na pohyb kurzoru v obrazovce.

Dále bývá myš vybavena několika ovládacími tlačítky.

Při hraní má široké uplatnění tam, kde je využíván klasický kurzor nebo při nutnosti souvislého, ale přesného pohybu jako například otáčení hráče v FPS hře².

Nevýhodou při používání myši může být jednostranná zátěž. Kvůli pohybu po stole hráči zatěžují ruku s myší více, než ruku na klávesnici. Při pravidelném hraní může hráč utrpět poškození z opakovaného namáhání [9].

1.4 Gamepad

Gamepad je čistě herní periferie. Má tvar určený pro použití oběma rukama. Nachází se na něm množství tlačítek a může být doplněn jedním, nebo dvěma analogovými joysticky. Některé verze nabízejí i vibrační odezvu. Nejdříve se využíval u herních konzol, s rozvojem osobních PC se však stal i zde velmi populární. Primární zaměření na hry dělá z gamepadu vynikající ovladač pro mnoho herních žánrů. Eliminuje problém ergonomie myši a klávesnice a všechna tlačítka jsou snadno dostupná. Proto je velmi často využíván.

Z hlediska intuitivního ovládání gamepad zaostává podobně jako myš s klávesnicí. Hry jsou s ním sice dobře ovladatelné, avšak často je nutný trénink pro zvládnutí složitějších principů ovládání a tlačítkových kombinací.

1.5 Joystick a volant

Joystick a volant představují ovladače pro specifické druhy her. Základní částí joysticku je páka umístěná kolmo v pohyblivém kloubu. Má nejlepší uplatnění v leteckých simulátorech, kde náklon páky mění polohu leteckých klappek. Ovládání je intuitivní a snaží se přiblížit realitě.

Účelem volantu je simulace ovládání závodního vozu. Obvykle je dodáván s dvěma nebo třemi pedály a případně řadicí pákou. Vyšší modely mají vibrační odezvu, při vyjetí z vozovky tak hráč cítí haptickou odezvu.

Joystick a volant pomáhají věrně simulovat letecké nebo závodní prostředí, pro které jsou určeny. Na rozdíl od předešlých periférií jsou ale použitelné pouze v úzké oblasti her, proto bývají nenáročnými hráči nahrazovány gamepadem nebo klávesnicí.

1.6 Dotyková obrazovka

Dotyková obrazovka na osobních počítačích není v současné době masově využívána. Větší rozšíření má v oblasti mobilních zařízení, jako jsou mobilní telefony a tablety, proto se problematice dotykové obrazovky budu více věnovat v sekci 2.1. Některé hry však využívají dotykovou obrazovku jako náhradu

²First-person shooter - akční hra zobrazená z pohledu herní postavy

za jiné periferie (např. myš). Záleží pak na návrhu samotné hry, zda toto ovládání přinese nějaké benefity, či bude spíše překážkou.

1.7 Ovládání pohybem

Kromě tradičních hardwarových periférií existují další technologie, jak ovládat hry, a to pomocí přirozeného pohybu. Tyto systémy snímají pozici hráče nebo zařízení a podle jejich změn vypočítají reakci ve hře. Ovládání bývá snadné na naučení, protože navozuje pocit přirozené reakce na podněty herního prostředí.

V této kapitole zmíním dvě technologie: ovládání pomocí akcelerometru a snímání pohybu těla (*motion capture*). Obě tyto technologie rozšiřují možnosti interakce s elektronickými zařízeními, fungují však na odlišném principu.

1.7.1 Akcelerometr

Akcelerometr je elektromechanická součástka, která měří zrychlení sil ve třech osách. Tyto síly mohou být statické, jako například tíhová síla, nebo dynamické: způsobeny pohybem nebo vibrováním akcelerometru [10]. Akcelerometr umožňuje detekovat natočení zařízení v prostoru a jeho přibližný pohyb.

V herním průmyslu se využívá především v mobilních zařízeních a v herních konzolách. Může nahradit joystick, emulovat volant nebo polohovací zařízení. Při využívání akcelerometru může v nevhodném prostředí docházet k rušení: například při jízdě v autobuse je téměř nemožné hrát závodní hru, jelikož při zatáčení autobusu je síla působící na akcelerometr vychýlená a dochází k nechtěnému zatáčení.

Další informace o pohybovém senzoru jsou uvedeny v sekci 2.2.

1.7.2 Motion capture

Velmi zajímavou technologií je ovládání pohybem, tzv. *motion capture*. Jedná se o snímání části nebo celého těla a jeho převod v reálném čase na digitální model. Ten se využívá mimo jiné k ovládání hry. Nejvýraznějším ovladačem v herním průmyslu se stal *Kinect* vyvinutý v roce 2009 firmou Microsoft [11].

Motion capture se hodí například pro sportovní nebo akční hry a to i při více hráčích. Nevýhodou může být nutnost poměrně velkého prostoru pro potřeby hraní.

1.8 Srovnání

V minulých kapitolách jsem stručně popsal několik způsobů, jakými lze ovládat hry. Každý z nich má řadu výhod i nevýhod. Následující tabulka obsahuje shrnutí těchto ovladačů a porovnání jejich možností. Vzhledem k tématu této práce se zde budu zabývat především tím, jak je daný způsob interaktivní a zda tak může obohatit zážitek ze hry.

Tabulka 1.1: Srovnání herních ovladačů

Ovladač	Výhody	Nevýhody
Klávesnice	- počet kláves - široké využití	- pouze dva stavy tlačítka (stisknuto, nestisknuto) - některé klávesy hůře dostupné
Myš	- přesnost	- jednostranná zátěž
Gamepad	- ergonomický - navržen pouze pro hry - interaktivní vibrační odezva	- ovládání není intuitivní, je třeba se jej naučit
Joystick, volant	- věrně simuluje realitu - interaktivní vibrační odezva	- vhodné jen pro některé herní žánry
Dotyková obrazovka	- široké využití - provázanost grafiky s ovládáním	- v některých hrách neúspěšně simuluje jiné periferie
Akcelerometr	- intuitivní	- rušení např. v dopravním prostředku či při jiném pohybu
Motion capture	- velmi přirozené ovládání	- potřeba většího prostoru

1.9 Shrnutí

Srovnání ukazuje rozdíly ovladačů v ergonomii, možnostech zadávání vstupu a dalších parametrech. Pro návrh hry je tak nutné citlivě navrhnout způsob ovládání a minimalizovat negativní dopady periferií.

Interaktivní prvky v mobilních telefonech

V následující Kapitole 2 se budu věnovat analýze senzorů v mobilních telefonech, které interagují s okolním prostředím. Popíšu jejich princip fungování a využití jak v mobilních hrách, tak jako interaktivní ovladač počítačové hry. Některé z těchto senzorů využiji při tvorbě interaktivního ovladače v Kapitole 4.

2.1 Dotykový displej

Dotykový displej je označení pro kombinaci zobrazovacího displeje a průhledné vrstvy schopné zaznamenávat dotyky vnějších těles, obvykle prstu nebo stylusu. Existuje několik technologií snímání dotyku, stručně popíšu *kapacitní dotykový panel*.

Kapacitní dotykový panel využívá elektrické vodivosti vnějších těles. Dotyk takového tělesa s povrchem displeje naruší elektrostatické pole obrazovky. Ze změny pole je zjištěno umístění dotyku, které pak dále zpracuje řadič. Kapacitní displeje proto fungují pouze s vodivými předměty, jako je lidský prst nebo speciální stylus [12].

Dotykové displeje nabízejí velmi široké možnosti uplatnění ve hrách. Na rozdíl od klasických ovladačů totiž propojují zobrazení grafiky se samotným ovládáním. Například přesun objektů v herním světě je velmi přirozený: hráč se jednoduše dotkne objektu a tahem prstu jej přesune po obrazovce. Ve srovnání s počítačovou myší odpadá *mezivrstva* ve formě periferie připojené k zařízení a hráčův prst se tak stává samotným kurzorem.

Schopnosti dotykových displejů, jako jsou gesta a snímání více dotyků najednou, zásadním způsobem zjednodušují ovládání mobilních telefonů a tedy i mobilních her. Některé jsou přímo postaveny na využití dotykových displejů, např. úspěšná hra *Fruit Ninja* [2]. Hlavní ovládací prvek představuje tah prs-



Obrázek 2.1: Využití dotykové obrazovky ve hře Fruit ninja [2]

tem, který simuluje seknutí čepelí. Hráč se snaží prstem přeseknout co největší množství padajícího ovoce najednou. Hra je příkladnou ukázkou využití dotykové obrazovky: letící ovoce představuje podnět, aby se hráč trefil do daného místa na obrazovce. Ovládání tak přímo koresponduje se zobrazovanou grafikou (pokaždé je pro optimální přeseknutí potřeba jiné gesto prstem) a splývá s hrou v jednu ucelenou část.

Této provázanosti se dá využít i v počítačové hře ovládané mobilním telefonem: Hlavní herní scéna je zobrazována na displeji počítače a hráč drží v ruce ovladač (mobilní telefon), který je spojen s hrou v PC. Dotyková obrazovka zde může plnit funkci jak ovládací, tak i zobrazovací. To nemusí být nutně výhodou, protože při ovládání hry se hráč přirozeně soustředí především na obraz v PC a může být nepohodlné často měnit pohled z monitoru na mobil. Pokud je ale hra určena pro více hráčů, otevírají se nové možnosti, jak využít individuální obrazovku každého hráče: informace na mobilním displeji mohou například tvořit herní výhodu pro jednotlivé hráče.

Této techniky lze však dosáhnout pouze úzkým provázáním hry s ovladačem. Vzhledem k těmto schopnostem mobilních telefonů v kombinaci s dostatečným výkonem, dalšími senzory a rychlou bezdrátovou komunikací má využití dotykového displeje velký potenciál.

Kromě interaktivní provázanosti může být dotyková obrazovka využita jako vstupní zařízení ve formě klasických dotykových tlačítek či pro detekci gest jedním nebo více prsty.

2.2 Akcelerometr

Dalším interaktivním senzorem je akcelerometr. Objevuje se v ovladačích k herním konzolím i v mobilních telefonech, kde plní různé funkce: zajišťuje automatickou orientaci displeje v závislosti na natočení mobilu, pohybová gesta jako například zatřesení nebo položení displejem dolů mohou sloužit k ovládní systému a široké využití má ve hrách.

Existuje velké množství mobilních her využívající náklon telefonu k ovládní. Následující výčet obsahuje přehled častých použití. Každý příklad doplňují hry, kde se tyto prvky vyskytují [13]:

- Simulace volantu v závodní hře (Lane Splitter, Jet Car Stunts)
- Náklon telefonu odpovídá náklonu herní podložky (Crazy Labyrinth 3D)
- Využití akcelerometru jako joysticku v leteckém simulátoru (Skies of Glory)
- Náklon telefonu mění pozici herního objektu (Donkey Jump, Box Busters)

Pro interaktivní ovladač k PC hře je použití akcelerometru ideální příležitostí. Pohyb herního objektu náklonem telefonu je pro člověka přirozenější a vtahuje hráče intenzivněji než ovládání tlačítka.

2.3 Gyroskop

Gyroskop je součástka pro detekci polohy zařízení. Měří změnu úhlu natočení telefonu kolem své osy. To umožňuje detekovat pohyb zařízení lépe než akcelerometr, ale vzhledem k detekci relativní změny se postupem času objevuje odchylka, a proto se gyroskop využívá v kombinaci s akcelerometrem a někdy též s kompasem. Díky tomu je možné určit natočení telefonu vůči gravitační síle a detekovat změnu polohy zařízení ve všech směrech [14].

2.4 Senzor přiblížení

Senzor přiblížení, nebo také *proximity sensor*, funguje na principu detekce elektromagnetického záření, které generuje vysílač. V případě přiblížení objektu do blízkosti snímače je zaznamenán odraz záření zpět k senzoru a je vyhodnocena vzdálenost objektu [15].

Nejběžnější využití proximity senzoru v mobilním telefonu je při hovoru. Před přiložením k uchu senzor detekuje přiblížení hlavy a zhasne displej. Zabrání tím tak nechtěnému ukončení hovoru dotykem ucha s obrazovkou.

Herní využití proximity senzoru není obvyklé, neexistují žádné hry využívající proximity sensor ve hře, pouze simulace tlačítka v PC hře podomácku vyrobeným senzorem přiblížení [16].

2.5 Ostatní senzory

Současné mobilní telefony mohou disponovat ještě několika dalšími senzory okolního prostředí, ale vzhledem k minimálnímu využití v herní oblasti zde nebudou více zkoumány. Jedná se například o sensor okolního světla, teploměr a magnetický sensor (kompas).

2.6 Shrnutí

V této kapitole byly popsány nejčastější interaktivní senzory vyskytující se v dnešních mobilních telefonech, jejich princip a možné využití. V kapitole 4 se budu snažit některé z nich využít při tvorbě interaktivního ovladače pro PC hru.

Podobná řešení

Před samotnou tvorbou softwarového projektu zmíním podobné systémy na trhu. V současné době již existují řešení využívající mobilní telefon k ovládní jiné hry. Každé z nich však obsahuje nedostatky a narušuje tak výsledný zážitek.

3.1 Telefon jako univerzální gamepad

Na internetu existuje velké množství aplikací, které nabízejí funkcionalitu podobnou bezdrátovému gamepadu. Ukázkou takové aplikace je např. Mobile Gamepad [3]. Po nainstalování příslušného ovladače do PC je možné využít telefon jako vstupní zařízení: umožňuje nastavit si tlačítka na displeji, pohybový senzor jako joystick atd.



Obrázek 3.1: Mobilní telefon jako univerzální gamepad[3]

3. PODOBNÁ ŘEŠENÍ

Výhodou tohoto řešení je možnost ovládat PC podobně jako jiné ovládací periferie a díky tomu není ovladač vázán na konkrétní hru. Co však brání použití v interaktivní hře, je absence jiných funkcí než těch, které již obsahují gamepady a podobné ovládací prvky. To je způsobeno univerzálností ovladače. Pro zachování standardizovaného ovládacího rozhraní a chování jako klasické vstupní zařízení počítače není umožněna širší komunikace s hrou. Tyto aplikace tak nabízejí stejnou nebo často i méně kvalitnější alternativu ke klasickým herním gamepadům a jiným ovladačům.

3.2 Apple AirPlay

Společnost Apple nabízí pro své telefony funkci sdílení obrazu na externí obrazovku. Toho je docíleno pomocí zařízení AppleTV a bezdrátové technologie AirPlay [4].

Toto řešení již lépe využívá interaktivní možnosti mobilních přístrojů, integrovaný displej tvoří druhou obrazovku, může zobrazovat ovládání dobře optimalizované pro konkrétní hru nebo dodatečné informace o hře (skóre, počet ujetých kol atd.). Zásadní nevýhodou však je nemožnost hrát hru s více hráči kvůli zvolené architektuře: herní logika včetně vykreslování běží na straně mobilního zařízení a do externí obrazovky se přenáší pouze obraz.



Obrázek 3.2: Technologie AirPlay [4]

3.3 AirConsole

Řešením, které se nejvíce podobá tématu této bakalářské práce, je herní platforma švýcarského startupu AirConsole [5]. Jedná se o webovou aplikaci společně s multiplatformní mobilní aplikací. Pomocí webového prohlížeče se vytvoří hra a k ní se připojí telefony jako ovladače. V nabídce je několik různých her a každá má vlastní ovladač v rámci mobilní aplikace. Samotná hra tedy probíhá na obrazovce počítače. AirConsole také poskytuje API, takže lze vytvořit vlastní hru v HTML5 nebo herním engineu Unity a propojit ji s tímto systémem.



Obrázek 3.3: Ukázka hry pomocí systému AirConsole [5]

Spojení probíhá přes externí servery, konkrétně přes infrastrukturu Google [5]. Kvůli tomu, že telefony nekomunikují s počítačem přímo, ale přes servery společnosti Google, má spojení větší odezvu, což silně ovlivňuje zážitek ze hry. Ovládací rozhraní je vytvořené pomocí HTML5, na displeji se tedy vykresluje webová vrstva. Zpracování dotyku v tomto případě není ideální, objevují se chyby (například špatná detekce kliknutí) a projevuje se vyšší odezva. Zároveň ovládání nevyužívá interaktivní možnosti mobilních telefonů, jediný možný způsob interakce je pomocí dotykového displeje. AirConsole tak vytváří podobnou funkcionalitu, jako je emulace gamepadu.

Přes všechny tyto nedostatky se AirConsole nejvíce blíží zaměření této práce, především hlavním cílem, a to vytvořit herní platformu podporující mezilidskou komunikaci lidí fyzicky přítomných na jednom místě.

3.4 Možnosti řešení

V rámci této práce jsem našel tři možnosti řešení, jak odstranit nedostatky předchozích systémů:

3.4.1 Rozšíření existující PC hry

Prvním způsobem vytvoření ovládání pomocí mobilního telefonu je vytvoření mobilního ovladače k již existující PC hře s otevřeným zdrojovým kódem. Stěžejní částí by bylo vytvoření „adaptéru“ napojeného na mobilní telefon. Nevýhodou je rozsáhlost takového projektu a složitost jejího rozšíření o interaktivní prvky.

3.4.2 Univerzální herní ovladač

Druhým způsobem je vytvoření mobilní aplikace poskytující rozhraní pro komunikaci s libovolnou hrou. Podobně je vytvořen systém AirConsole. Jeho velkou výhodou je univerzálnost použití, kdy jedna mobilní aplikace může být využita pro mnoho her, které implementují požadované rozhraní. Na rozdíl od AirConsole by pro univerzální použití včetně interaktivních senzorů muselo rozhraní podporovat velké množství funkcí, např. definice ovládání, načtení grafiky z libovolné hry, poskytnutí ovládací logiky mobilnímu ovladači a funkce pro využití mobilních senzorů. Složitost tohoto řešení však překračuje rozsah bakalářské práce.

3.4.3 Komunikační modul

Třetí možností je vytvoření systému pro síťovou komunikaci PC a ovladače. Takovéto řešení by bylo možné použít při vývoji libovolné hry, ale s nutností vytvořit samostatný ovladač zvlášť pro každou hru. Na druhou stranu není třeba vytvářet rozhraní pro sdílení herní logiky a grafických prvků, to může být implementováno v ovladači nebo vyřešeno sdílením zdrojového kódu s hrou. Zařízení si budou mezi sebou posílat zprávy s herními informacemi, na které budou reagovat definovaným způsobem. Tento návrh umožňuje vytvořit snadno integrovatelný systém komunikace. Mobilní telefony budou mít zároveň širší možnosti využití hardwaru.

Toto řešení jsem zvolil pro implementaci v bakalářské práci, protože eliminuje nevýhody stávajících řešení na trhu a zároveň nabízí přijatelný kompromis mezi složitostí systému a univerzálností pro případ použití v dalších hrách.

Tvorba systému

Informace o možnostech ovládání hry pomocí mobilního telefonu využiji v praxi a vytvořím systém, ve kterém bude mobilní telefon v roli interaktivního ovladače PC hry. To vyžaduje tvorbu komunikačního protokolu a hry, která bude využívat tento protokol. Softwarový projekt komunikačního protokolu i hry samotné je hlavní náplní praktické části této bakalářské práce.

Struktura kapitoly odpovídá vývojovým fázím softwarového projektu [17]. První část je zaměřena na analýzu systému, požadavky a softwarový návrh, v implementační fázi zmíním některé zásadní části realizace, testování programu a hratelnosti a na závěr se budu zabývat možnostmi, jak by bylo možné vytvořený systém rozšířit v budoucnosti.

4.1 Požadavky

Cílem první fáze softwarového projektu je specifikace vlastností projektu, určení prostředí, ve kterém bude systém nasazen, a vymezení funkcionalit nutných pro naplnění účelu systému.

Pro ucelený přehled o systému nejprve popíšu běžný průběh hraní hry postavené na systému vytvořeném v této práci: Skupina hráčů s chytrými telefony se připojí do jedné WiFi sítě společně s PC, kde poběží hra (server). Poté spustí ovladač v mobilním telefonu, který automaticky vyhledá hry dostupné v lokální síti. Po kliknutí na název nalezené hry naváže ovladač spojení s hrou. Hráč, který může měnit nastavení hry (*admin*, typicky první připojený hráč), nastaví parametry hry a po připojení všech hráčů zahájí hru. V případě odpojení nebo ztráty spojení bude možné se do hry opět připojit. Po skončení hry se systém dostane do stejné fáze jako před jejím zahájením, tj. bude opět možné měnit herní parametry apod.

Z výše uvedeného scénáře budu vycházet při tvorbě požadavků, které jsou rozděleny do dvou částí: komunikační protokol a hra samotná. Každá část má vlastní seznam funkčních a nefunkčních požadavků. Tyto požadavky představují základní kritéria pro návrh architektury systému a jeho funkcionalit.

4.1.1 Komunikační protokol

Účelem komunikačního protokolu je zajistit obousměrnou výměnu zpráv mezi mobilním telefonem (*ovladačem*) a počítačem (*hrou*). K tomu je vhodné použít stejný systém v obou zařízeních, aby je bylo možné implementovat pouze jednou a ne pro každou platformu zvlášť. Stejně tak je třeba zachovat identickou strukturu síťových zpráv mezi všemi zařízeními.

Ovladače budou s hrou komunikovat v lokální Wi-Fi síti. Při bezdrátové komunikaci je nutné počítat s výpadky signálu. Může je způsobovat nedostatečný výkon Wi-Fi routeru, příliš velká vzdálenost od vysílače nebo objekty v prostoru, které brání šíření signálu. Proto je třeba u zpráv rozlišit, zda je nutné, aby byla doručena spolehlivě (např. kliknutí hráče na tlačítko), nebo zda systému nevádí případná ztráta zprávy (např. často se opakující informace z pohybového senzoru).

Vzhledem k univerzálnímu zaměření systému je vhodné oddělit přenos zpráv a jejich zpracování (parsování) pro dosažení větší modularity. Dále je výhodné umožnit konfiguraci síťových parametrů, aby odpovídala požadavkům konkrétní hry.

Specifikace funkčních požadavků softwarového systému popisuje základní chování systému a jeho dílčích částí. Tato specifikace slouží jako nutný základ při návrhu projektu a také jako podklad pro zhodnocení úspěšnosti aplikace [18].

Pro komunikační protokol platí tedy následující funkční požadavky:

1. Systém umožňuje nalézt všechny běžící hry v lokální síti.
2. Každá síťová zpráva má tyto vlastnosti: typ zprávy, data zprávy, spolehlivost doručení, zachování pořadí.
3. Hra i ovladač dokážou detekovat ztrátu spojení.
4. Síťový port (nebo jejich rozsah) a další lze definovat zvlášť jak pro ovladač, tak pro hru.
5. Ovladač umožňuje odeslání informací o interakci uživatele (kliknutí na tlačítko, data z pohybového senzoru).
6. Ovladač je schopen přijímat informace o stavu hry (herní události, kompletní stav hry pro vykreslení na mobilním displeji).

Funkční požadavky jsou doplněny požadavky nefunkčními, které specifikují kvalitu aplikace podle kritérií. Mezi ně se řadí například výkon, spolehlivost, rozšiřitelnost a další. Zaměřil jsem se především na cílovou platformu, modulárnost a jednoduché nasazení v dalších hrách:

1. Síťová komunikace probíhá v rámci stejné Wi-Fi sítě. Podpora pro *blue-tooth* a další bezdrátové technologie není implementována.

2. Komunikační systém běží na alespoň jedné mobilní platformě (Android, iOS nebo Windows Phone) a alespoň jedné PC platformě (Windows, Linux nebo OS X).
3. Systém je modulární: je možné definovat vlastní typy zpráv, jejich strukturu a vlastnosti podle požadavků hry.
4. Parsování zpráv nezávisí na komunikačním frameworku. Je možné implementovat rozdílné parsování nebo i několik různých metodik najednou.
5. Zdrojový kód komunikačního modulu je sdílený mezi hrou a ovladačem.
6. Komunikační modul nezávisí na herním engine a herním žánru, a je tak použitelný v libovolné hře.

4.1.2 Hra

Požadavky na hru a herní zařízení se v první řadě netýkají žánru a obsahu, ale snaží se především využít komunikační systém a možnosti interaktivity. Volba žánru a konkrétního typu hry je proto přesunuta až do fáze návrhu, kde bude nutné vhodně navrhnout zapracování interaktivních prvků do herních mechanik.

Důležitou vlastností hry je demonstrace nových možností oproti klasickým ovladačům. Mezi ně se mimo jiné řadí schopnost vykreslování grafiky v mobilní obrazovce. Tuto a další nové možnosti je třeba ve hře výrazně prezentovat.

Pro celou hru (ovladač i PC) platí následující funkční požadavky:

1. Hra využívá k ovládní pohybový senzor mobilního telefonu a dotykovou obrazovku.
2. Ovladač je schopen vykreslit herní scénu nebo její část na obrazovce mobilního telefonu.
3. Hra je určena pro 2 až 8³ hráčů (ovladačů).
4. Mobilní telefon může využívat vibrace jako interaktivní odezvu na události ve hře.
5. Každá instance hry má nastaven název serveru a herní parametry.
6. Parametry hry může měnit právě jeden z připojených hráčů.
7. Zvolený herní žánr je vhodný pro hru více hráčů a obsahuje jak kooperativní, tak kompetitivní složku.
8. Veškerá konfigurace hry probíhá pomocí ovladače: po spuštění hry na PC již není potřeba další obsluha PC (mimo ukončení hry).

³Více hráčů již není praktické pro zobrazení na jedné obrazovce.

9. Zobrazováním hry nebo její části na displeji mobilního telefonu přináší danému hráči taktickou výhodu oproti ostatním.

Nefunkční požadavky na hru zahrnují především použití cílové platformy:

1. Ovladač poběží na alespoň jedné mobilní platformě (Android, iOS nebo Windows Phone).
2. Hra poběží alespoň na jedné PC platformě (Windows, Linux nebo OS X).
3. Použitý herní engine je multiplatformní a použitelný jak ve hře, tak v ovladači.

4.2 Návrh

Návrh systému je druhou fází softwarového projektu. Obsahuje podrobnější popis komponent systému, jejich objektů a popis jejich vzájemných vztahů.

Hra pro PC i ovladač používají dvě základní komponenty. První z nich je síťový modul zajišťující veškerou síťovou komunikaci a druhou komponentou je herní engine.

4.2.1 Komunikační modul

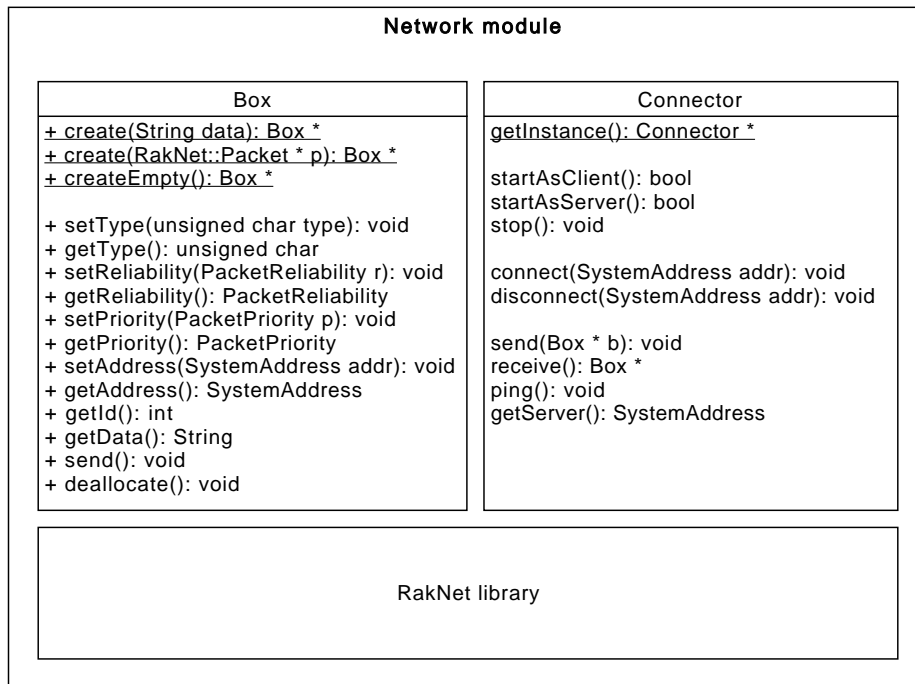
Úkolem komunikačního modulu je zajistit výměnu zpráv v lokální síti. Kromě toho definuje obecnou síťovou zprávu a její vlastnosti. Modul tedy tvoří dvě základní komponenty: **Box** a **Connector**.

Hlavním přínosem modulu je odstínění implementace síťové vrstvy od hry a herního engine a velmi snadné nasazení v libovolné jiné hře zaměřené nejen na interaktivní ovládání pomocí mobilního telefonu. Výhodou vytvoření abstraktní vrstvy v porovnání s přímým použitím RakNetu je především zjednodušení navázání spojení, manipulace s datovými pakety a nezávislost na vnitřní implementaci.

4.2.1.1 Box

Box je objekt reprezentující síťovou zprávu. Používá se jako vyšší úroveň abstrakce pro předávání herních zpráv mezi klienty a serverem. Jeho cílem je zapouzdřit veškeré informace potřebné pro práci s daty. Používá se v obou směrech komunikace, což znamená, že se vytváří před odesláním, ale také po přijetí zprávy.

Obsahuje data zprávy, typ zprávy, adresu příjemce (pokud bude zpráva odeslána), nebo adresu odesílatele (pokud byla zpráva přijata), dále vlastnosti doručení, mezi které patří spolehlivost (spolehlivé doručení znamená, že odesílající zařízení čeká na potvrzení přijetí zprávy), priorita, jak rychle má být zpráva doručena (zprávy s nízkou prioritou mohou být řazeny do fronty



Obrázek 4.1: Struktura síťového modulu

a odeslány najednou, což šetří síťový provoz) a nastavení, zda je nutné dbát na zachování pořadí zpráv, v jakém byly odeslány.

4.2.1.2 Connector

Connector je objekt, který stanovuje programové rozhraní veškerého fungování síťové komunikace: obsahuje metody pro aktivování síťových rozhraní, připojování a odpojování klientů, posílání a příjem zpráv. Společně se třídou **Box** tvoří základní pilíř pro výměnu zpráv mezi hrou a ovladačem.

4.2.1.3 RakNet

Hlavičková definice tříd **Connector** i **Box** tvoří abstraktní rozhraní nad samotnou implementací a umožňuje tak snadnou změnu platformy, případně síťové knihovny. Pro implementaci rozhraní jsem zvolil knihovnu **RakNet**.

RakNet je multiplatformní síťová knihovna se zaměřením na hry. Je napsaná v jazyce C++. Vyznačuje se velmi nízkými nároky na hardware, kvalitní dokumentací a množstvím funkcionalit, které se často využívají v multiplayerových hrách [19]. Mezi ně patří například Lobby systém⁴, šifrování komuni-

⁴Lobby systém - virtuální místnost, kde se shromažďují hráči před začátkem online hry.

kace nebo peer-to-peer spojení mezi hráči. Veškerá komunikace je postavena na protokolu UDP, přesto knihovna obsahuje vlastní implementaci pro podporu spolehlivého doručení zpráv.

Knihovnu jsem zvolil především pro její snadné použití, kvalitní dokumentaci a podporu jak klasických počítačů, tak i mobilních platforem. Stejnou implementaci modulu je tak možné využít v ovladači i ve hře na PC.

4.2.2 Hra

Návrh hry se skládá z několika částí. Prvním krokem je volba herního žánru. Poté se budu věnovat hledání herního enginu, na kterém bude hra postavena. Poslední částí bude již samotný návrh struktury systému, objektů, vztahů mezi nimi a tvorba samotného komunikačního protokolu.

4.2.2.1 Volba herního žánru

Jak bylo zmíněno v analýze, důležitým požadavkem pro hru je možnost efektivního využití interaktivních možností chytrých telefonů. K tomu musí být uzpůsoben herní žánr tvořící kostru veškeré hrátelnosti. Proto jsem nejprve zvolil dva směry a porovnával jejich výhody a nevýhody:

Prvním z nich je akční 2D hra, kde kamera zobrazuje scénu shora. Hlavním pilířem hry je souboj dvou týmů. Cílem je dosáhnout většího počtu zásahů než protivník. Každý hráč ovládá jednu postavu vybavenou zbraní. Mezi interaktivní prvky lze zařadit ovládání postavy pohybovým senzorem a intuitivní míření pomocí dotykového displeje (místo dotyku prstu a střed obrazovky vytvoří přímkou definující směr výstřelu). Inovativním prvkem je zobrazení herní mapy: většina herního území se zobrazuje na obrazovce, kterou vidí všichni hráči. Vyskytují se ale místa, kde hráč může vyjít mimo herní území (např. dveře nebo vchod do domu). Po vkročení na tato místa postava zmizí z hlavní obrazovky a danému hráči se na mobilním displeji vykreslí interiér domu i s hráčem. Tímto se elegantně rozšíří herní prostor na další objekty a zároveň je zachována interaktivita mezi hráči (např. je možné zasáhnout hráče v domě skrz okno). Zmíněný prvek považuji za velmi efektivní příklad využití mobilního telefonu, který obohacuje samotnou hru a přináší nové taktické možnosti.

Tento návrh hry bohužel obsahuje i méně vhodné prvky. Zobrazení velké části mapy, kterou uvidí hráči obou týmů, odhaluje pozice všech hráčů a ubírá tak možnost taktizování v terénu, jelikož se hráč nemůže skrýt před protivníky jinak, než že vejde do výše zmiňované „interaktivní místnosti“. Tím by však mohla nastat situace, že hráči budou takto skrytí příliš často a hlavní obrazovka téměř ztratí význam.

Druhým návrhem je sportovní zaměření, konkrétně 2D fotbal. Podobně jako u akční hry má kamera pohled shora a soupeří proti sobě dva týmy s cí-

Obvykle nabízí chat mezi hráči, volbu herních místností nebo žebříček výsledků.

lem skórovat vícekrát než soupeř. Hlavní obrazovka zobrazuje fotbalové hřiště s hráči. Hráč ovládá svou postavu pomocí akcelerometru a dotyk s obrazovkou mu umožňuje vystřelit (sílu výstřelu určuje doba dotyku před uvolněním prstu z obrazovky). Obohacením hry jsou bonusy. Objevují se náhodně na hřišti a při jejich sebrání poskytuje danému hráči časově omezenou výhodu. Mezi bonusy patří zvýšení rychlosti, zvětšení maximální síly kopu a především neviditelnost. Neviditelný hráč zmizí z hlavní obrazovky a na mobilním displeji se mu začne vykreslovat hřiště totožné s hlavní obrazovkou. Jeho taktická výhoda spočívá v tom, že sám sebe vidí, ale není viděn ostatními a může procházet skrz hráče. Bonus neviditelnosti představuje výraznou inovaci podobně jako rozšířené místnosti v návrhu akční hry, eliminuje však případné nedostatečné využití hlavního displeje.

Všechny zmíněné a ještě některé další výhody a nevýhody těchto dvou návrhů shrnuje tabulka 4.1 (prvek s větším přínosem je vyznačen kurzivou).

Tabulka 4.1: Srovnání vybraných herních žánrů

Akční hra	Fotbal
Pohyb postavy pomocí akcelerometru	
Taktická výhoda při zobrazení hry v mobilním displeji	
Obsahuje kooperativní i kompetitivní složku hry	
<i>Rozšíření scény o interaktivní místnosti</i>	Bonus neviditelnosti
Riziko časté nepřítomnosti na hlavní obrazovce	<i>Možnost neviditelnosti je časově omezena</i>
Hlavní obrazovka ubírá možnosti taktizování	<i>Hřiště je optimální pro zobrazení na společné obrazovce</i>
Střelba na základě místa dotyku s obrazovkou	Síla kopu se odvíjí od délky dotyku
Míření může být těžké na naučení	<i>Kopnutí je velmi intuitivní</i>

Podle tabulky se jeví jako vhodnější kandidát implementace fotbalu. Mě osobně více zaujal i z důvodu atraktivity pro diváky například na večírku s přáteli, kteří mohou být jako pozorovatelé lépe začleněni do hry.

Po zvážení všech těchto aspektů jsem se rozhodl použít fotbal jako téma demonstrační hry.

4.2.3 Herní engine

Slovní spojení herní engine je možné definovat různými způsoby. Závisí například na provázanosti herní logiky s vykreslováním, obecnosti engine z pohledu herního žánru apod. V této práci je pod pojmem *herní engine* myšlen softwarový framework, který obsahuje základní komponenty potřebné pro vývoj her,

jako je vykreslování grafiky, detekce kolizí, zpracování uživatelských vstupů, práce se zvukem a další [20].

Důležitou částí návrhu je výběr herního engine. Pro účely této hry jsem stanovil několik požadavků, které musí engine splňovat:

- Podpora více platforem (jak PC, tak i mobilní telefony) podobně jako RakNet, aby bylo možné použít oba tyto frameworky současně. Pokud engine nebude napsán v jazyce C++, měl by podporovat práci se sítí, nebo musí existovat alternativní síťová knihovna použitelná s tímto engine.
- Vzhledem k nulovému finančnímu rozpočtu tohoto projektu musí být engine licencován zdarma.
- Herní engine musí umět pracovat s 2D grafikou a být nezávislý na herním žánru.
- Mezi funkce herního engine musí patřit mimo jiné podpora dotykové obrazovky a zpracování pohybového senzoru.
- Musí existovat volně dostupná kvalitní dokumentace a návody, jak herní engine používat.

Z velkého množství různých možností se do výběru použitelných engineů dostal *Torque 2D*, *Cocos2d-x* a *Marmalade*. Všechny tyto enginey nabízejí funkce splňující výše uvedené požadavky. Nevýhodou *Torque 2D* je méně rozsáhlá dokumentace a návody v porovnání s *Cocos2d-x* a *Marmalade*, proto jsem jej vyřadil. Při rozhodování mezi *Cocos2d-x* a *Marmalade* jsem bral v potaz také licenci na herní engine: *Cocos2d-x* je na rozdíl od *Marmalade* zveřejněn s otevřeným zdrojovým kódem, proto jsem jej zvolil pro tuto práci.

4.2.4 Návrh ovládání

Ovládání hry je realizováno pomocí mobilního telefonu spojeného se hrou. Pro připojení do hry a nastavení parametrů zápasu počítá návrh s klasickým využitím tlačítek zobrazených na dotykové ploše. Mimo to je třeba navrhnout ovládání akcí herní postavy v zápase. Tyto akce jsou dvě: pohyb a kopnutí do míče.

Hlavní ovládací prvek pohybu tvoří akcelerometr. Zajišťuje pohyb herní postavy po dvourozměrné ploše hřiště. Změřením velikosti náklonu telefonu se vypočítá směr a rychlost hráče. Toto ovládání je zajímavé především z pohledu křivky učení: je velmi jednoduché pochopit princip pohybu pomocí akcelerometru, ale získání dostatečné citlivosti pro přesnou a rychlou hru vyžaduje delší cvik. Tuto myšlenku vystihuje aforismus Nolana Bushnella⁵: Nejlepší

⁵Nolan Bushnell je americký podnikatel. Ve videoherním průmyslu se proslavil založením společnosti Atari, což byla průkopnická firma v oblasti počítačových her [21]

hry jsou lehké na naučení, avšak obtížné na ovládnutí [21]. Zmíněný princip platí v menším měřítku i pro ovládání akcelerometrem.

Ovládání kopnutí do míče zajišťuje dotyková obrazovka, konkrétně výrazné tlačítko umístěné optimálně pro dotyk palce. Při stisknutí tlačítka ještě nedojde ke kopu, pouze se začne měřit doba stisknutí tlačítka. Ke kopu dojde až po uvolnění tlačítka a jeho síla se vypočítá právě podle doby stisku. Aby se hráči nepohybovali po hřišti se stále „nabitým kopnutím“, přidal jsem do návrhu ještě snížení maximální rychlosti pohybu v závislosti na stupni intenzity střely. Tento prvek motivuje hráče, aby „nabíjeli“ pouze v situacích, kdy potřebují kopnout do míče. Herními parametry, které je nutné citlivě nastavit pro vyváženou hratelnost, jsou počáteční a maximální intenzita kopu, rychlost nárůstu intenzity kopu, rychlost klesání maximální rychlosti. Ladění těchto parametrů probíhalo během testování hratelnosti.

Využití interaktivních senzorů v mobilním telefonu je vhodné analyzovat, zda přináší nějaké benefity ve srovnání s tradičními způsoby ovládání z první kapitoly. Ovládání kopnutí je srovnatelné s klasickými hardwarovými požadavky, dostačující informaci tvoří stavy *stisknuto* a *nestisknuto*. Z tohoto hlediska nenabízí dotyková obrazovka žádný přínos.

Zajímavější výsledky přináší srovnání akcelerometru s ostatními typy ovládání. Použití tlačítek pro pohyb podobně jako na klávesnici nebo gamepadu znamená nemožnost citlivosti pohybu a představuje komplikaci při vykreslování hřiště na mobilní obrazovce: prsty brání ve viditelnosti. Z hlediska principu fungování ovladače se jako nejpodobnější jeví joystick. Umožňuje plynule měnit směr ve dvou dimenzích, má tak podobné možnosti jako akcelerometr. Výhodou akcelerometru je však přítomnost v mobilním telefonu a tím velká dostupnost.

4.2.5 Architektura hry

Hra na PC tvoří server pro interaktivní ovladače. Každý ovladač je spojen s jednou postavou a z těchto ovladačů je zvolen jeden (typicky označovaný jako *admin*), který může spravovat nastavení hry.

Jedinou herní scénou je samotné hřiště. Je zobrazeno po celou dobu spuštění hry. Před zahájením zápasu se však na něm nevyskytuje míč. Hráči se na hřišti objeví po připojení a volbě týmu. Poté se mohou volně pohybovat po hřišti jak během zápasu, tak i před ním.

Z hlediska objektového návrhu jsou objekty rozděleny do dvou kategorií: herní a systémové objekty. Herní objekty reprezentují části hry z pohledu reálného světa. Patří mezi ně fotbalové hřiště (třída *StadiumScene*), míč (*Ball*), hráč (*Player*) a obecná reprezentace herního bonusu *BonusInterface*, kterou implementují tři použité bonusy (neviditelnost, rychlost a síla kopu).

Systémovou kostru hry tvoří třída *Game* a zpracování herních událostí pomocí návrhového vzoru *command*[22]: každá interakce s hrou nebo změna stavu hry vyvolá událost, která je zpracována příslušným *handlerem*. Udá-

losti jsem rozdělil na 6 kategorií. Každá událost je rozlišena identifikátorem, ke kterému je přiřazen příslušný handler. Přehled typů událostí, jejich popis a příklady použití shrnuje tabulka 4.2.

Tabulka 4.2: Rozdělení herních handlerů

Kategorie	Popis	Příklady událostí
Void	Událost, která nevyžaduje žádná data	Prvotní spuštění hry, konec zápasu (časoměry)
Box	Přijetí Boxu ze sítě	Data z akcelerometru, připojení do hry, zahájení hry adminem, ...
Touch	Uživatelský dotyk a jedna z fází <i>pressed</i> , <i>moved</i> , <i>released</i>	Kopnutí, podržení tlačítka pro odpojení
Acceleration	Událost vyvolaná přečtením dat z pohybového senzoru	Naklonění telefonu, periodické čtení senzoru
Collision	Kolize dvou herních objektů	Srážka dvou hráčů, vstřelení gólu
String	Událost zpracovávající textový řetězec	Změna jména hráče nebo hry

Důležitou součástí obsluhy handlerů je jejich správa a přístup k nim. K tomu slouží třída `HandlerMap`, která přiřazuje každému identifikátoru události nejvýše jeden Handler. Obsluhu události popisuje následující příklad: Každý zápas má časový limit a po jeho skončení se vyvolá událost s identifikátorem `VOID_COUNTDOWN_FINISHED`. Pro její obsluhu je nutné najít handler kategorie `Void` (nepředáváme handleru žádná další data) s tímto identifikátorem. Zpracování události handlerem zajistí následující příkaz:

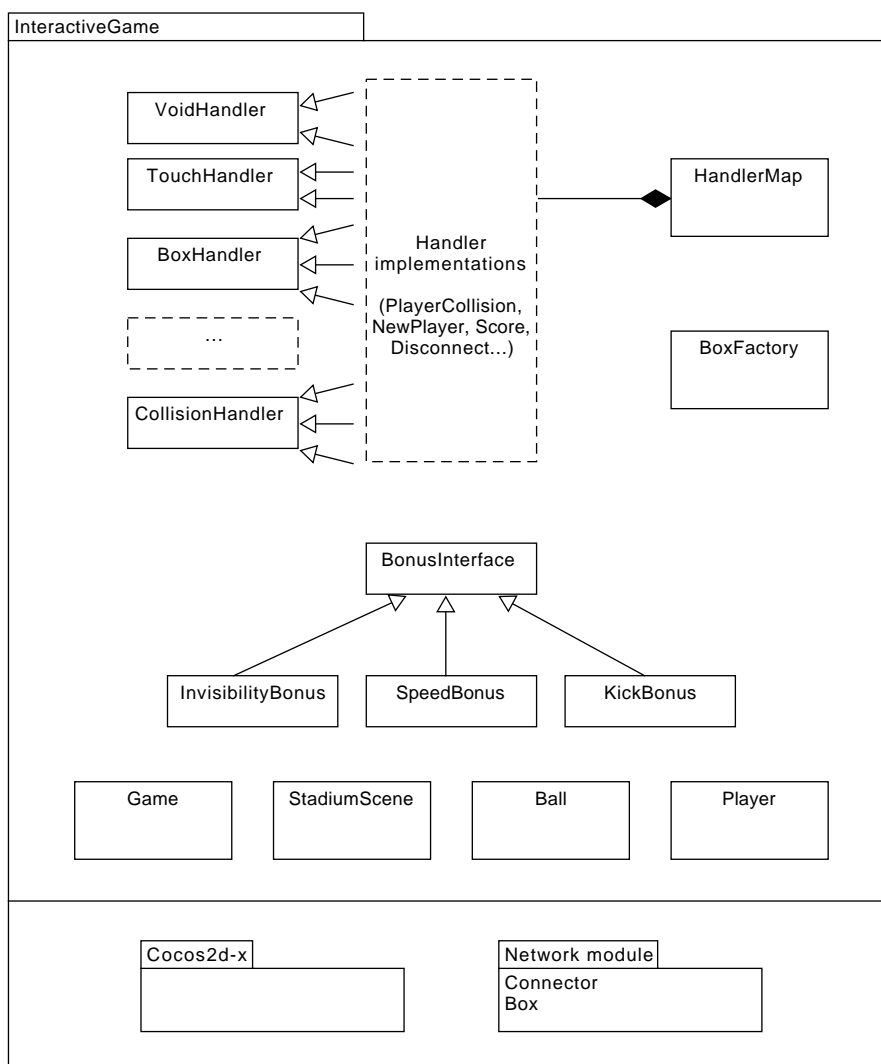
```
handlerMap->getVoidHandler(VOID_COUNTDOWN_FINISHED)->execute();
```

Podobně se volají metody v ostatních kategoriích, mění se pouze argumenty metody `execute()`. Použití handlerů výrazně zpřehledňuje celý systém a odděluje vznik události od jejich obsluhy, čímž je docíleno vyšší modularity celé aplikace.

4.2.5.1 Herní bonusy

Prvkem, který obohacuje celkovou hratelnost, jsou bonusy. Umožňují dočasné zvýhodnění hráče během zápasu změnou jeho vlastností na hřišti.

Obecnou definici herního bonusu obsahuje abstraktní třída `BonusInterface`. Mezi její základní metody patří `activate()` a `deactivate()`. Bonusy dále



Obrázek 4.2: Architektura hry

obsahují informace o době, jak dlouho budou aktivní, kdo vzal daný bonus a jakou má bonus grafiku. Návrh počítá se třemi implementacemi bonusů: rychlost, síla kopu a neviditelnost.

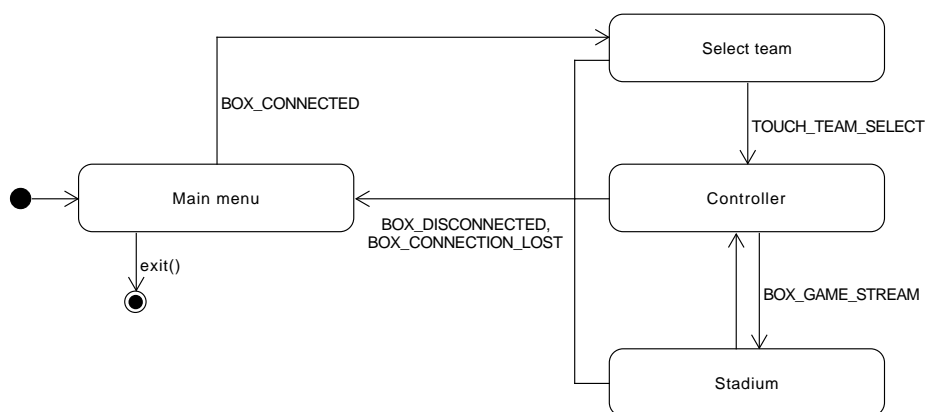
Bonus rychlosti zvyšuje hráči maximální rychlost, jakou se může pohybovat po hřišti. Může tak doběhnout míč lépe než protivník nebo se rychleji přiblížit k brance. Podobně jako rychlost zvyšuje bonus kopnutí sílu, která bude aplikována na míč. Pokud hráč s tímto bonusem vystřelí, míč letí rychleji a může doletět přes značnou část celého hřiště. Nejinteraktivnějším bonusem je neviditelnost. Hráč po sebrání bonusu zmizí z hlavní obrazovky a celé hřiště se mu

zobrazí na mobilní obrazovce a kromě toho může procházet skrz ostatní hráče. Neviditelný hráč získává velkou taktickou výhodu, může volněji manipulovat s míčem a překvapit protivníka svou pozicí na hřišti.

4.2.6 Architektura ovladače

Základní struktura ovladače je velmi podobná hře, jelikož využívá velké množství stejných komponent: Návrhový vzor Handler je použit jak ve hře, tak i v ovladači, obě aplikace spolu sdílí abstraktní části kódu, jen implementace konkrétních handlerů jsou odlišné. Stejně tak je využíván síťový modul a sdílené jsou i grafické zdroje. Proto se zde budu věnovat především odlišným prvkům.

Na rozdíl od hry ovladač obsahuje více herních scén (obrazovek). Konkrétně se jedná o hlavní menu, obrazovku s výběrem týmu, ovládací obrazovka a scéna s vykresleným hřištěm. Přechody mezi jednotlivými obrazovkami a událostmi, které vyvolají přechody, popisuje diagram 4.3.



Obrázek 4.3: Události přechodů mezi obrazovkami

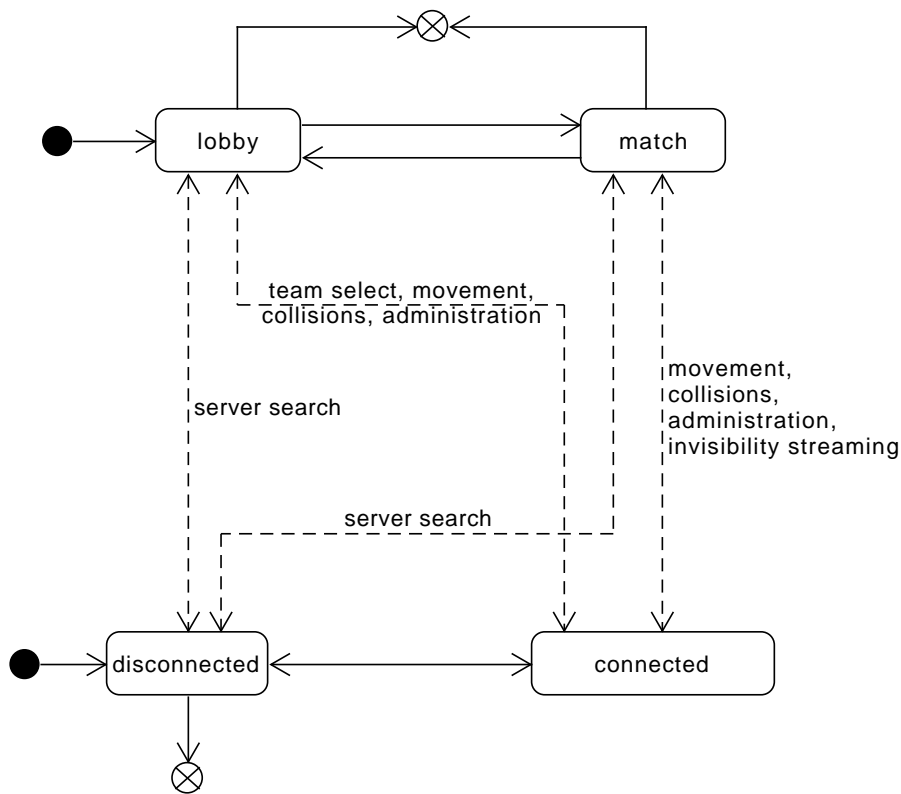
Hlavní menu je první zobrazenou scénou po spuštění ovladače. Obsahuje nastavení hráčovy přezdívky, aktivace vibrací a seznam nalezených serverů. Po kliknutí na vybraný server naváže ovladač spojení s hrou a zobrazí volbu týmu. Hráč si může zvolit ze dvou týmů (červený nebo modrý), případně si nechat určit tým automaticky. Po přiřazení týmu se zobrazí ovládací obrazovka, které dominuje velké tlačítko pro kopnutí do míče. Pokud vezme hráč během zápasu bonus neviditelnosti, zobrazí se mu na mobilním displeji totožné hřiště jako na hlavní obrazovce. Kopnout do míče při neviditelnosti lze dotykem libovolně na obrazovce, aby se předešlo stínění prstem v místě, kde zrovna probíhá hra. V případě odpojení se ovladač vrátí do hlavního menu a v případě ztráty spojení zobrazí chybovou hlášku.

4.2.7 Komunikace mezi ovladačem a hrou

Vzhledem k použití handlerů v systému je možné návrh síťové komunikace realizovat s jejich využitím: přijetí síťové zprávy vyvolá herní událost a následně proběhne její zpracování. Vzhledem k tomu, že síťové zprávy i události jsou rozlišeny identifikátorem, je možné tyto identifikátory konsolidovat a použít pro událost i zprávu totožný identifikátor.

Nejběžnějším scénářem komunikace je sled následujících kroků: vyhledání her v lokální síti, připojení na zvolenou hru, volba týmu, spuštění hry a poté herní komunikace. V případě aktivace bonusu neviditelnosti začne hra posílat informace o svém stavu danému ovladači, aby byl schopen vykreslit kompletní hru na své obrazovce.

Stavový diagram 4.4 přehledně zobrazuje jednotlivé stavy hry i ovladače z hlediska síťové komunikace a zachycuje, kdy se posílají důležité informace.



Obrázek 4.4: Komunikační stavy hry a ovladače

Nedílnou součástí návrhu celého systému je zpracování dat ze zpráv. Některé zprávy žádná data neobsahují, informační hodnotu představuje pouze identifikátor Boxu. Jiné obsahují jednoduchý textový řetězec a jejich zpraco-

vání je pak triviální, ale existují i zprávy s komplexní strukturou a tím pádem i složitým zpracováním. Takovou zprávou je například stav hry posílaný ze serveru pro vykreslování hry při neviditelnosti. Zpráva musí obsahovat informace o všech viditelných hráčích (pozice, rychlost a směr pohybu, barva týmu), míči a některé další informace. Takováto zpráva musí mít definovanou pevnou strukturu. Protože se obsah může měnit (např. když se změní počet hráčů), její návrh vyžaduje hlubší analýzu.

Původním záměrem, jak zpracovávat zprávy, bylo vytvořit vlastní parsovací systém pro každou zprávu. Ukázalo se však, že tento způsob je nedostatečný především u složitějších zpráv. Jako lepší varianta se nabízelo využití existujících technologií pro serializaci dat. Z několika vhodných řešení jsem si zvolil knihovnu *Protocol Buffers* vyvíjenou společností Google. Jedná se o multiplatformní knihovnu s otevřeným zdrojovým kódem, která umožňuje definici struktury zprávy, její serializaci a deserializaci a použití více programovacími jazyky současně [23]. Použití Protocol Buffers je velmi jednoduché. Nejprve je třeba definovat strukturu zprávy s určenou syntaxí a poté v příkazové řádce vygenerovat soubory s požadovaným zdrojovým kódem (v tomto případě C++). Tyto vygenerované soubory se poté importují do projektu a je možné je přímo využívat.

Využití Google Protocol Buffers elegantně vyřešilo problém zpracování zpráv a umožnilo velmi jednoduše specifikovat komplexní struktury dat posílané po síti.

V definici zprávy lze určit, zda se proměnná ve zprávě musí vyskytovat (parametr `required`), nebo se nutně vyskytovat nemusí (`optional`), či může existovat několik instancí (`repeated`), ale také nemusí být žádná. Následující ukázka ilustruje definici jednoduché zprávy v systému Protocol Buffers:

```
message Person {
    required string name = 1;
    required int32 id = 2;
    optional string email = 3;
    repeated string address = 4;
}
```

Pokud je nutné změnit strukturu zprávy, stačí upravit soubor s definicí a znovu vygenerovat zdrojové soubory. Není potřeba ručně měnit algoritmus parsování zprávy, jako by tomu bylo při použití vlastního primitivního parsování.

4.2.8 Shrnutí

Ve fázi návrhu jsem vytvořil základní strukturu síťového modulu včetně zpracování zpráv, hry a ovladače. Následující fází je implementace: naprogramování navržených funkcionalit a vytvoření funkčního systému.

4.3 Implementace

Přesto, že implementace zabrala značnou část tvorby této práce, není možné obsáhnout všechny části tvorby hry a ovladače, proto se zde budu zabývat především těmi částmi, které jsou něčím zajímavé, komplikované nebo důležité pro celý systém.

4.3.1 Vývojové prostředí

Celý vývoj probíhal v operačním systému OS X, konkrétně ve vývojovém prostředí Xcode. Jednalo se o mou první zkušenost s tímto prostředím i s operačním systémem OS X, proto byla tvorba v počátcích komplikovaná, později však vývoj probíhal bez problémů a začal jsem dokonce využívat výhod, které mi toto prostředí nabízelo.

Kromě vývojového prostředí jsem hojně využíval příkazovou řádku, především pro kompilaci projektu pro operační systém Android.

4.3.2 Struktura projektu, integrace knihoven

Prvním krokem každé implementace je založení projektu. Podle návodů jsem vytvořil prázdný projekt s herním engine Cocos2d-x. Do tohoto projektu jsem poté importoval knihovnu RakNet a vytvořil modul pro síťovou komunikaci. Přesto, že vývoj tohoto modulu probíhal z důvodu praktičnosti sloučeně se zbytkem hry, byl vyvíjen tak, aby jej bylo možné snadno importovat do libovolné jiné aplikace.

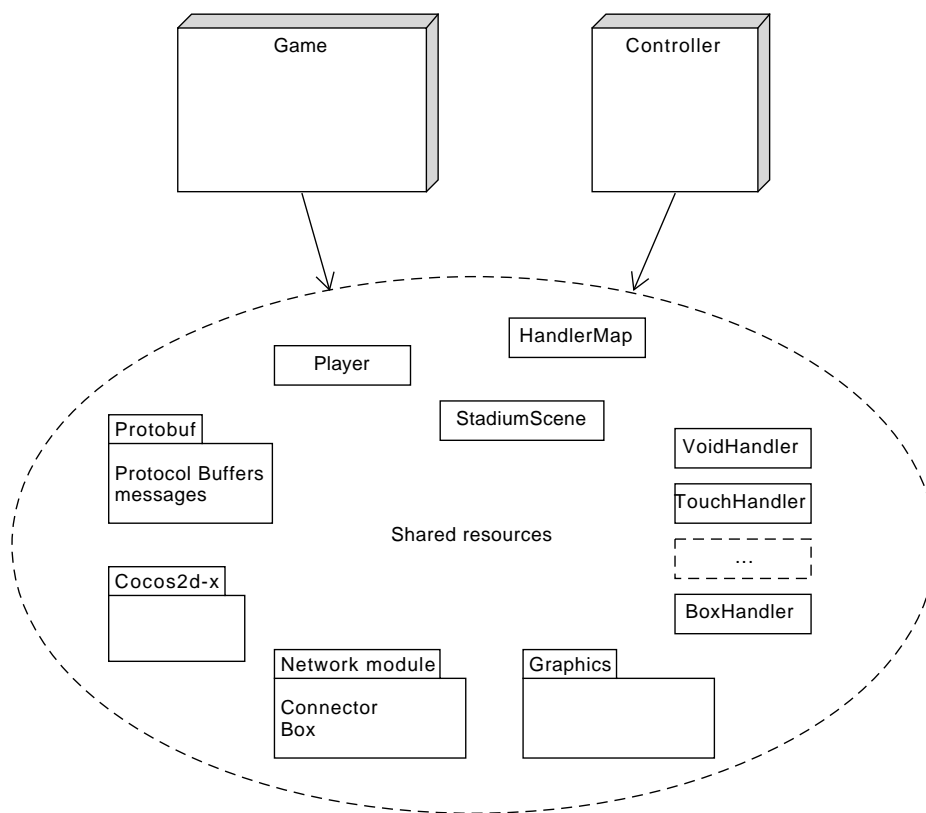
Vytvořil jsem tedy 2 projekty: jeden pro hru na PC a druhý pro mobilní ovladač. Oba projekty odkazují na stejné zdrojové soubory síťového modulu a změna v něm se tak projeví v obou aplikacích. Kromě toho sdílí projekty také několik dalších souborů, z nichž nejvýznamnější je objekt `StadiumScene` obsahující implementaci fotbalového hřiště. Sdílení některých souborů ilustruje obrázek 4.5.

4.3.3 Síťový modul

Implementace síťového modulu probíhala bez větších komplikací. Během studování knihovny RakNet jsem vytvořil testovací aplikace *klient* a *server*, abych si vyzkoušel práci s knihovnou. Velmi kvalitním zdrojem informací byla sada ukázkových programů nacházející se přímo v distribuci knihovny. Programy jsou velmi dobře zdokumentované a pomáhají pochopit daný kód.

4.3.4 Připojení nového hráče

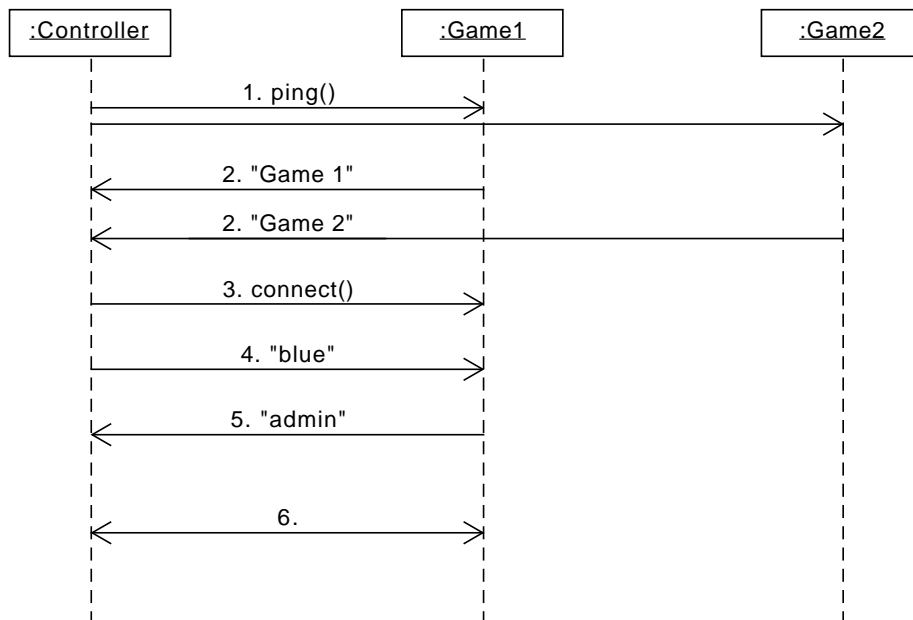
Stěžejním krokem síťové komunikace je navázání spojení s ovladačem. V této sekci popíšu typický průběh navazování spojení z pohledu systému:



Obrázek 4.5: Sdílení částí aplikace mezi hrou a ovladačem

1. Hledání serverů: Ovladače periodicky posílají dotazovací zprávu na všesměrovou adresu v lokální síti.
2. Odpověď serverů: Herní servery, které přijaly dotazovací zprávu, odpovídají zprávou obsahující název serveru.
3. Připojení na server: Ovladač zobrazuje nalezené servery a po kliknutí na vybraný server s ním naváže aktivní spojení.
4. Volba týmu: Ovladač je sice spojen se serverem, hráč se však stále nezobrazil na hřišti, musí si totiž nejprve zvolit tým. Provedená volba se odešle na server a hráč je zobrazen na hřišti.
5. Admin: Pokud je hráč první připojený, server mu odešle dodatečnou informaci, že má ovladač práva na správu hry.
6. Spojení je navázáno, ovladač může volně komunikovat se hrou.

Číslování jednotlivých kroků koresponduje pro lepší přehlednost s kroky v diagramu 4.6, který názorně ilustruje připojení nového hráče.



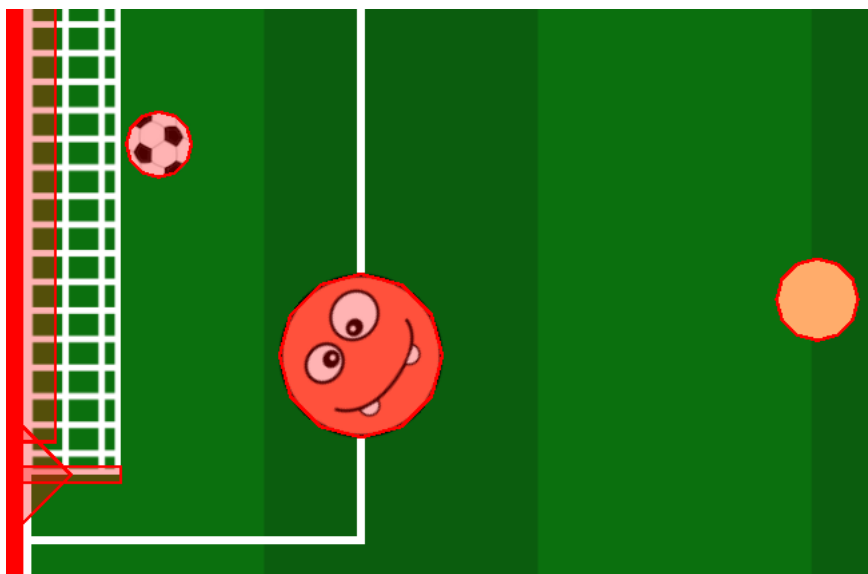
Obrázek 4.6: Připojení nového hráče

4.3.5 Fyzikální engine

Jednou z funkcí knihovny Cocos2d-x je simulace chování objektů za pomoci integrovaného fyzikálního engine. Herní princip fotbalu vychází ze simulace reálného světa, proto je zde fyzikální engine hojně využíván. Nejdůležitější prvky v tomto případě tvoří objekty, jejich kolize a detekce překryvu.

Fyzikální objekt obsahuje 2 elementy: tvar a definice materiálu. Materiál má 3 složky: hustota, pružnost a tření. Ve hře jsou fyzikálními objekty hráči, míč, branky, okraje hřiště a detektory gólu. U každého fyzikálního objektu lze definovat, s jakým typem jiných fyzikálních objektů může kolidovat (například srážka dvou hráčů). Použití této definice je výhodné pro detekci gólu (obrázek 4.7): do prostoru branky jsou umístěny neviditelné fyzikální objekty reprezentující detektory gólu, se kterými sice nekoliduje ani hráč, ani míč, ale je detekován překryv dvou objektů. Pokud se míč překryje s detektory, vyvolá se událost gólu a je přičten bod příslušnému týmu. Pohyb míče i případného hráče v brance je však neovlivněn.

Dalším příkladem je bonus neviditelnosti: Pokud jej hráč vezme, změní se jeho definice kolizí a přestane kolidovat s ostatními hráči. Přesto, že je na



Obrázek 4.7: Zobrazení fyzikálních objektů a detektoru gólu

hřišti neviditelný, jeho fyzikální reprezentace se pohybuje po hřišti a kolize s míčem tak funguje zcela normálně.

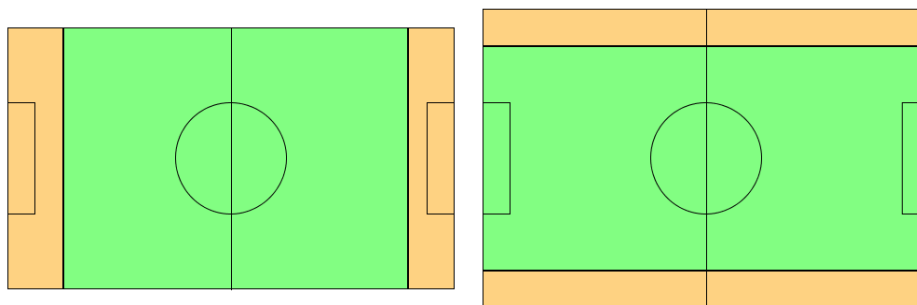
4.3.6 Neviditelnost

Stěžejní funkce, která výrazně obohacuje interaktivitu v porovnání s klasickými ovladači, je projekce kompletní herní scény na mobilní obrazovku při aktivním bonusu neviditelnosti. Její realizace s sebou přinesla několik komplikací, které jsou podrobně rozebrány v následující části.

4.3.6.1 Poměr stran hřiště

Při snaze o co nejvěrnější zobrazení stejného hřiště v mobilním telefonu i ve hře se logicky nabízí otázka, jak efektivně sdílet zdrojový kód mezi oběma platformami. Použití identického herního enginu tuto potřebu zjednodušuje, ale je potřeba vytvořit dostatečně obecné třídy a umožnit jejich modifikaci pro potřeby dané platformy. Jako první byla vytvořena třída `StadiumScene` a `Player` pouze na PC. Ukázalo se však, že přes veškeré výhody herního enginu existují komplikace, které zabraňují bezproblémovému použití stejného zdrojového kódu:

První komplikací bylo použití herních handlerů přímo ve `StadiumScene`. Mobilní platforma disponuje odlišnými implementací handlerů, proto nebylo možné projekt zkompilovat. Oprava této chyby spočívala v přesunu použití handleru do třídy `Game`. Třída `StadiumScene` se tak stala nezávislou na herních událostech a bylo ji možné importovat do mobilní aplikace.



Obrázek 4.8: Přesahy hřiště při různých poměrech stran

Druhý, avšak komplikovanější problém představovala rozdílnost herních obrazovek, zejména různorodost poměrů stran. Návrh hřiště počítá s tím, že se hra bude snažit přizpůsobit rozměry hřiště monitoru tak, aby byla vyplněna celá plocha obrazovky. Například pokud hra běží na monitoru s poměrem stran 16:9, herní hřiště bude mít totožný poměr stran. Komplikace nastávají, pokud se do hry připojí zařízení s odlišným poměrem stran. Vzhledem k tomu, že rozměry mobilních displejů jsou značně rozdílné, je velká pravděpodobnost výskytu tohoto problému. Hřiště v mobilním displeji se, podobně jako u PC, vykreslí přes celou obrazovku a při rozdílných poměrech stran vznikají odchylky od skutečné polohy. To způsobuje nepřesnou hru či přesah objektu „za hřiště“.

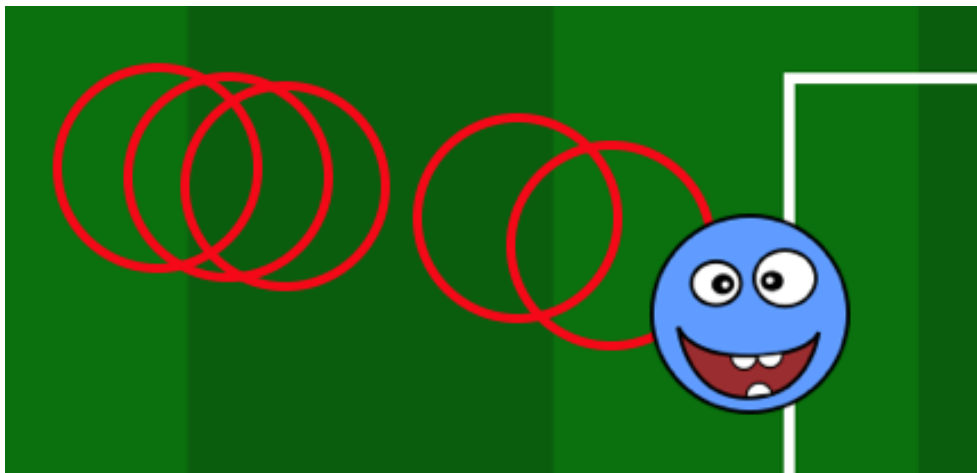
Řešení tohoto problému spočívá ve dvou krocích. Prvním z nich je odeslání rozměrů hřiště do mobilního telefonu. Hřiště se vykreslí s totožnými rozměry jako na PC. To ale způsobuje přesah hřiště přes ty strany, které jsou na mobilním displeji menší než na PC. Přesahy ilustruje obrázek 4.8, oranžová barva vyznačuje přesahující oblasti v mobilním displeji. Pro zobrazení těchto oblastí jsem upravil vykreslování scény, aby se hřiště posouvalo v závislosti na pozici hráče a byly tak skryty jen nejvzdálenější části herní plochy. Tyto úpravy ale musely být vytvořeny nad samotnou třídou `StadiumScene`, aby neovlivnily vykreslování na PC.

4.3.6.2 Vykreslování hráčů v mobilní obrazovce

Jako velmi zajímavou část během vývoje považuji ladění vykreslování hráčů na mobilním displeji. Aktivní bonus neviditelnosti periodicky posílá informace o hře, jejíž součástí jsou také data o jednotlivých hráčích. Na počátku vývoje obsahovaly tyto údaje pouze souřadnice hráče na hřišti. Tyto souřadnice pak používá ovladač pro nastavení pozice hráče na mobilním displeji. Při testování se však ukázalo, že tento návrh je téměř nepoužitelný: pohyb hráčů byl velmi trhaný, nepomohlo ani zvýšení frekvence posílání zpráv (z 30 na 60 zpráv za vteřinu) a je proto potřeba vykreslování optimalizovat. Obrázek 4.9 znázorňuje

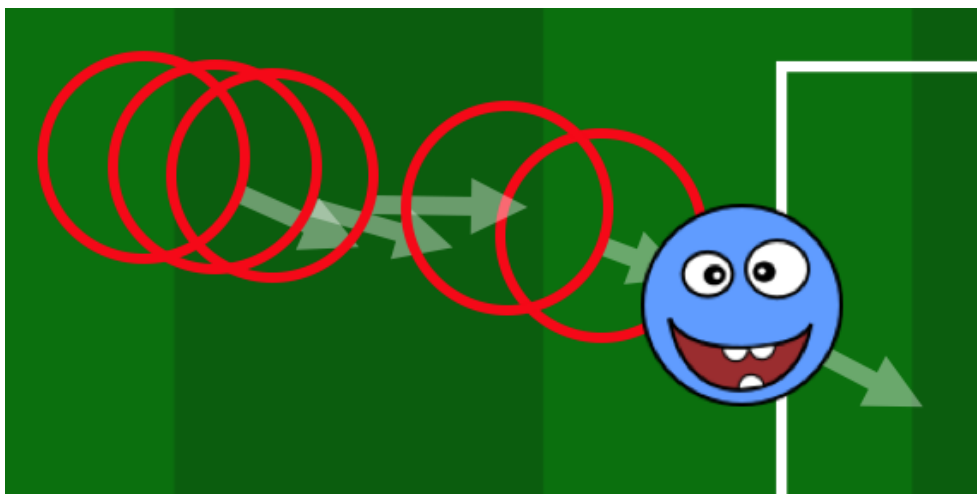
4. TVORBA SYSTÉMU

pozice hráče přijaté po síti a ilustruje nedostatečnost tohoto řešení. Jednotlivé pozice jsou nepravidelně vzdáleny. Překreslování tak netvoří dojem plynulosti.



Obrázek 4.9: Trhané vykreslování hráče

Pro docílení plynulejšího pohybu jsem přidal do zprávy o hráčích informaci o vektoru pohybu. Z dat se tak dala rekonstruovat nejen pozice, ale i rychlost a směr pohybu. Hráč na mobilní obrazovce se tedy pohyboval i v případě, že delší dobu nebyla přijata zpráva s aktualizovanými údaji. Toto řešení vylepšilo celkový dojem z vykreslování, stále se však objevoval trhaný pohyb a občasné uskakování hráčů, byť v menší míře.

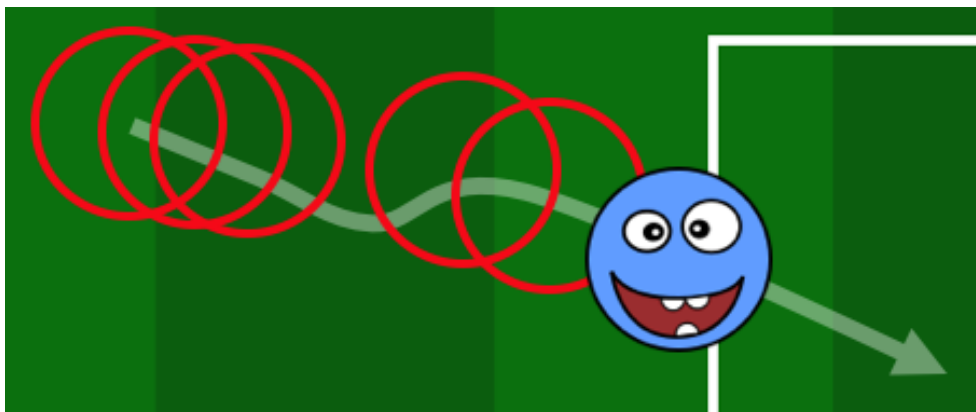


Obrázek 4.10: Vykreslování hráče s predikcí pohybu

Protože bylo při tvorbě zprávy s daty o hráčích nastaveno odesílání a přijímání zpráv bez zajištění spolehlivosti, kromě ztrát zpráv mohlo také docházet

ke změně pořadí přijatých zpráv. Ztráta zpráv v tomto případě nepředstavuje zásadní problém, jelikož neaktuální zprávy nejsou z hlediska hry informačně hodnotné, komplikaci představuje změna pořadí. „Starší“ zprávy, které se dostanou před ty novější, mohou způsobit „přeskakování“ hráčů: Pokud přijdou dvě zprávy v opačném pořadí, hráč se nejprve přesune na starší místo (do protipohybu) a poté zpět na správné místo. Tento problém vyřešilo jednoduché nastavení spolehlivosti zprávy na `UNRELIABLE_SEQUENCED`, což způsobilo zahazování neaktuálních paketů a elegantně vyřešilo problém s přeskakováním.

Zprávy o hráčích nyní poskytují všechny informace potřebné pro vykreslení. Další vylepšování plynulosti se proto týkaly ovladače. Predikce pohybu s využitím aktuální rychlosti vykazuje nedostatky ve chvíli, kdy hráč změni rychlost nebo směr pohybu a zpráva s touto informací přijde se zpožděním. Tato situace způsobuje zmiňované trhání. Způsobem, jak eliminovat toho trhání, je plynulá změna pozice pomocí animace objektu. Při zpracování nově přijaté zprávy se tak vypočítá vektor změny přijaté pozice od pozice na mobilním displeji. Z toho se vytvoří animace, která přesune objekt hráče relativně o vzdálenost vypočítaného vektoru. Zároveň však zůstane zachována predikce pohybu.



Obrázek 4.11: Vykreslování s predikcí pohybu a plynulou korekcí

Kombinace všech těchto vylepšení splňuje požadavky na plynulost pohybu, který je tak téměř identický jako na hlavní obrazovce. Výsledné řešení ilustruje obrázek 4.11.

4.3.7 Shrnutí

Ve fázi implementace byl podle návrhu vytvořen funkční systém. Obě aplikace se podařilo vytvořit podle očekávání, stěžejní části a případné komplikace zde byly rozebrány do větších podrobností.

4.4 Testování

Následující sekce se věnuje testování software a to jak automatické, tak manuální. Je sice zařazena až po fázi implementace, některé testy však probíhaly souběžně s implementací.

4.4.1 Unit testy

Základní formou testování jsou automatizované testy, tzv. *unit testy*. Pomocí nich se kontroluje elementární funkčnost jednotlivých komponent a funkcí. Pro implementaci jsem použil testovací framework *Catch* [24]. Mezi jeho výhody patří velmi snadná integrace a velké množství testovacích maker. Následující ukázka zobrazuje jeden z automatických testů síťového modulu, který kontroluje správnou inicializaci třídy `Box`:

```
TEST_CASE("GameNet tests")
{
    SECTION("Box")
    {
        GameNet::Box * box = nullptr;

        // string to box
        box = GameNet::Box::create("test");
        REQUIRE( box->getData().compare("test") == 0 );
        box->deallocate();

        // packet to box
        RakNet::Packet * packet = new RakNet::Packet();
        packet->data = new unsigned char[5];
        packet->length = 5;
        memcpy(packet->data, "\0test", 5*sizeof(char));
        box = GameNet::Box::create(packet);
        REQUIRE( box->getData().compare("test") == 0 );

        // custom dealloc - not the "real" RakNet packet
        delete [] box->getPacketData();
        delete [] packet->data;
        delete box;
    }
}
```

Vzhledem k vývoji v prostředí Xcode jsou automatické testy implementovány na platformách iOS a Mac OS X. Toto řešení splňuje požadavky uvedené v návrhu hry, je zahrnuta alespoň jedna mobilní a jedna počítačová platforma.

Rozšíření testování i na Android je komplikovanější a přesahuje požadavky této práce.

Při vývoji pro více mobilních platforem by však bylo vhodné automaticky testovat všechny platformy, aby se zamezilo chybám způsobeným např. odlišnou architekturou či hardware.

4.4.2 Testy paměti

Dalším systémovým testem je kontrola správné alokace paměti. Vzhledem k tomu, že celý systém je napsán v jazyce C++, nejsou paměťové chyby automaticky kontrolovány. Herní engine Cocos2d-x sice pracuje s pamětí efektivně, přesto však během vývoje docházelo v aplikacích ke špatné dealokaci. Za následek této chyby bylo nekontrolovatelný růst spotřeby operační paměti.

Velké zásluhy na vyřešení těchto problémů mají integrované nástroje ve vývojovém prostředí Xcode. Ty umožňují kontrolovat správu paměti za běhu programu a zobrazují špatně uvolněnou paměť v reálném čase dokonce i s detekcí, na jakém místě v kódu k úniku došlo. Tímto způsobem bylo možné efektivně odstranit příčiny chyb a zajistit optimální správu paměti.

4.4.3 Testování hratelnosti

Důležitou fází při vývoji počítačových her představuje testování hratelnosti. Jeho cílem je nastavení parametrů hry pro vyvážení obtížnosti, ovládání a dalších metrik. Testování hratelnosti se účastnilo 12 lidí a jejich poznatky byly konzultovány během hry či po ní.

4.4.3.1 Pohyb hráče

V případě fotbalu bylo nutné najít optimální model pohybu hráčů, aby hra nepůsobila jako pomalá nebo naopak příliš dynamická. V počátcích návrhu měli hráči nízkou rychlost a hra s nižším počtem hráčů byla příliš zdlouhavá. Problém vyřešilo mírné zvýšení maximální rychlosti hráčů.

Jako druhý poznatek při ovládání hráči reportovali nepohodlné ovládání hry v případě aktivního bonusu neviditelnosti. Aby viděli na displej, měli tendence naklánět telefon méně než při sledování hlavní obrazovky a dosahovali tak nižších rychlostí. Toto bylo vyřešeno úpravou zpracování dat z akcelerometru: pro dosažení maximální rychlosti stačí naklonit telefon méně než doposud. Hráči tak mohou sledovat obrazovku z lepšího úhlu.

4.4.3.2 Okraje hřiště

Další problém během hraní se vyskytl v situacích, kdy se míč dostal do okrajových částí hřiště: hrany a rohy hřiště, případně vnější strana branek. Pokud míč následoval jeden nebo více hráčů, nebylo snadné míč dostat z rohů zpět do hry. Tento problém negativně ovlivňoval herní zážitek.

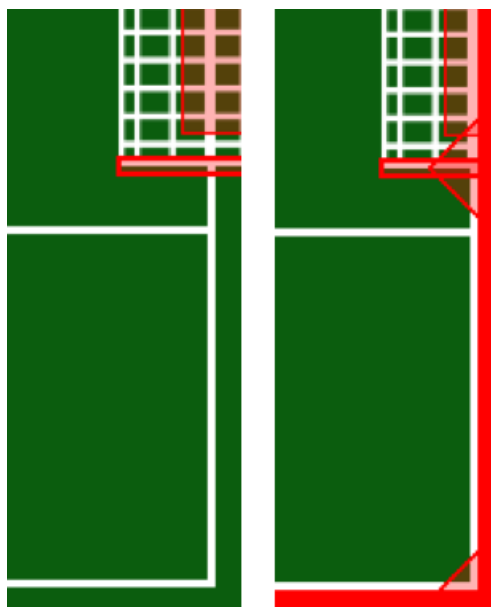


Obrázek 4.12: Testování hratelnosti

Cílem vyřešení toho problému bylo předejít či eliminovat zaseknutí míče v rozích a jeho snadné odsunutí z okraje hřiště. Pro tyto účely jsem využil možností fyzikálního enginu a hřiště rozšířil o další neviditelné objekty:

- Zdvojení okrajů hřiště: první okraj je určen pro míč a druhý (mírně přesahující „za hřiště“) pro hráče. Ti se tak mohou dostat mírně za okrajovou čáru, což umožnilo lepší manipulaci s míčem.
- Rohy hřiště: zabránit zaseknutí míče v rozích se podařilo přidáním neviditelných objektů do rohů, které kolidují pouze s míčem.
- Branky: podobně jako v rozích jsem přidal neviditelné objekty ke hranám branek.

Opravu tohoto problému ilustruje obrázek 4.13. Řešení elegantně eliminovalo problémy s míčem a vylepšilo celkovou hratelnost.



Obrázek 4.13: Oprava okrajů hřiště

4.5 Dokumentace

Nedílnou součástí každého softwarového produktu tvoří dokumentace. Jedná se o soubor textů a dalších materiálů, které provázejí celý vývoj projektu. V této práci tvoří nejdůležitější části dokumentace požadavků, informační diagramy jednotlivých komponent a dokumentace zdrojového kódu. Kromě toho je součástí také návod na kompilaci a spuštění aplikací.

Dokumentace požadavků je uvedena v kapitole 4.1. Obsahuje popis základních funkcionalit potřebných pro vytvoření funkčního systému.

Dokumentace zdrojového kódu zlepšuje přehlednost a srozumitelnost kódu především pro vývojáře. V definici metod a rozhraní vysvětluje účel jejich použití a v implementaci pomáhá např. s pochopením algoritmů či složitějších oblastí kódu.

Kromě textového popisu je důležité také znázornění vztahů mezi komponentami a jejich spolupráce. Vhodně navržený diagram může pomoci pochopit problematiku lépe než velké množství textu. Množství diagramů v předešlých částech práce tak tvoří součást projektové dokumentace.

4.6 Možnosti rozšíření

Jako v mnoha softwarových projektech zde existuje prostor pro další vývoj. Některé prvky nebyly implementovány, ať už z důvodu časové náročnosti, nebo protože nespádají do nutných požadavků práce.

Jedním z návrhů na vylepšení je grafika. Lepší zpracování obrázků, vytvoření více animací a použití částicových efektů by hru vizuálně oživilo a pomohlo tak k intenzivnějšímu hernímu zážitku.

Dalším možností rozšíření je použití zvukových efektů. Ty nebyly z časových důvodů implementovány, jejich nasazení umocní celkovou atmosféru hry.

Podobných návrhů na další vývoj existuje celá řada, například: nové typy herních bonusů, integrace herního serveru do mobilní obrazovky (server by tak mohl být použit v mobilním zařízení například při připojení do externí obrazovky), vytvoření hráčů s umělou inteligencí (využitelné, když je připojen lichý počet hráčů) a mnoho dalších.

4.7 Využití v dalších hrách

Kromě rozšíření stávající hry umožňuje systém snadno vytvořit odlišnou hru. Nejzásadnějšími kroky je vytvoření herních tříd, vlastní množiny událostí a zpráv a poté tvorba příslušných handlerů využívající tyto třídy a zprávy.

Závěr

V této bakalářské práci jsem se zabýval zkoumáním možností, jak ovládat PC hru pomocí chytrého telefonu. V první kapitole jsem popsal nejběžnější herní ovladače a jejich schopnosti a porovnal je se senzory chytrých telefonů v druhé kapitole. Součástí teoretické části bylo také nalezení podobných existujících řešení a diskuse jejich vlastností.

Všechny tyto kapitoly sloužily jako teoretický základ pro analýzu a realizaci softwarového projektu. Ten měl za cíl vytvořit systém pro interaktivní ovládání PC hry a demonstrovat jeho schopnosti v jednoduché hře. Tohoto cíle se podařilo dosáhnout vytvořením hry s tematikou fotbalu. Softwarový projekt byl rozdělen na standardní fáze: analýza, návrh, implementace a testování.

Systém vytvořený v této práci pro podporu interaktivních her ulehčuje případný návrh dalších her. Není omezen na konkrétní herní žánr, jeho modularita jej předurčuje k použití v libovolném typu interaktivní hry.

V rámci bakalářské práce jsem využil poznatky získané při studiu, především z oblastí softwarového projektu, objektového modelování a síťové komunikace. Kromě toho považuji za osobní přínos první větší zkušenost s herním enginem a vývojem her. Z dlouhodobého hlediska bych chtěl využít získané zkušenosti pro další výzkum způsobů ovládání her a jejich interakce s cílem propojit herní svět s realitou.

Literatura

- [1] Purcaru, B. I.: *Game vs. Hardware, The History of PC gaming*. Bogdan Ion Purcaru, 2014.
- [2] Halfbrick: Fruit Ninja - the greatest fruit-slicing game in the world! Dostupné z: www.fruitninja.com
- [3] iQuest: Mobile Gamepad - Aplikace pro Android ve službě Google Play. Dostupné z: <https://play.google.com/store/apps/details?id=com.mmh.mobilegamepad&hl=cs>
- [4] Apple: Bezdrátové streamování obsahu z iPhoneu, iPadu a iPodu touch přes AirPlay. Dostupné z: <https://support.apple.com/cs-cz/HT204289>
- [5] N-Dream: AirConsole. Dostupné z: <http://www.airconsole.com>
- [6] Rylich, J.: *Počítačové hry jako fenomén nových médií*. Diplomová práce, Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví, Studia nových médií., 2011.
- [7] 24kupi: Android smart phone watch - 24kupi pro edition. Dostupné z: <http://www.24kupi.com/24kupi-pro-edition-watch>
- [8] Wikipedia: Game Controller - Wikipedia, the free encyclopedia. Dostupné z: https://en.wikipedia.org/wiki/Game_controller
- [9] Scott, C.: Repetitive Strain Injury: How to prevent, identify and manage RSI. Dostupné z: <https://web.eecs.umich.edu/~cscott/rsi.html>
- [10] Juránek, M.: Prostředky automatického řízení - akcelerometr | VŠB Technická univerzita Ostrava. Dostupné z: http://homel.vsb.cz/~jur286/prostredky_aut_rizeni/preklad.htm

- [11] Kean, S.; Hall, J.; Perry, P.: *Meet the Kinect: An Introduction to Programming Natural User Interfaces*. ITPro collection, Apress, 2012, ISBN 9781430238898. Dostupné z: <https://books.google.cz/books?id=Cfxnzjf9phAC>
- [12] Gray, L.: *How Does a Touch Screen Work?* High-Tech Science, Gareth Stevens Publishing LLLP, 2013, ISBN 9781482465266. Dostupné z: <https://books.google.cz/books?id=BtDnAgAAQBAJ>
- [13] Bluestacks: Accelerometer Games Archive - AndroidTapp. Dostupné z: <http://www.bluestacks.com/blog/app-reviews/archive/tag/accelerometer-games.html>
- [14] Jayachandra, Y.: *SMARTPHONE FRONTIERS: Technologies, Applications and Markets*. McGraw-Hill Education, ISBN 9789351340232. Dostupné z: <https://books.google.cz/books?id=ldieBAAAQBAJ>
- [15] Fraden, J.: *Handbook of Modern Sensors - Physics, Design and Applications*. Springer New York Heidelberg Dordrecht London, čtvrté vydání, 2010.
- [16] Chakravarthi, M.: Free Hand Control Of Computer Game With A Proximity Sensor. Dostupné z: <http://myelectricalcomputer.blogspot.cz/2014/02/free-hand-control-of-computer-game-with.html>
- [17] Leach, R.: *Introduction to Software Engineering*. Taylor & Francis, 1999, ISBN 9780849314452. Dostupné z: <https://books.google.cz/books?id=XW8P1P7sC0kC>
- [18] Wieggers, K.: *Požadavky na software*. Computer Press, Albatros Media a.s., 2016, ISBN 9788025140727. Dostupné z: <https://books.google.cz/books?id=UQ3qCwAAQBAJ>
- [19] JenkinsSoftware: RakNet - Multiplayer game network engine. Dostupné z: <http://www.raknet.net>
- [20] Gregory, J.: *Game Engine Architecture, Second Edition*. Taylor & Francis, 2014, ISBN 9781466560017.
- [21] Goldberg, M.; Vendel, C.: *Atari Inc: Business is Fun*. Syzygy Company Press, 2012, ISBN 9780985597405. Dostupné z: <https://books.google.cz/books?id=3FwGMtRafrAC>
- [22] Buschmann, F.; Meunier, R.; Rohnert, H.; aj.: *Pattern-Oriented Software Architecture, A System of Patterns*. Pattern-Oriented Software Architecture, Wiley, 2013, ISBN 9781118725269.

- [23] Google: Protocol Buffers | Google Developers. Dostupné z: <https://developers.google.com/protocol-buffers/>
- [24] Nash, P.: philquared/Catch: A modern, C++-native, header-only, framework for unit-tests, TDD and BDD. Dostupné z: <https://github.com/philquared/Catch>

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	bin	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF