

Peter Holotík

Prechod hráča na server 13.0.2

Postupné simulovanie a reprodukovanie chyby, ktorá spôsobuje pád serveru (13.0.2). Oprava bola riešená formou zmeny platformy, na ktorej server bežal.

Dôvod:

Pravidlá fakultného RoboCup zápasu nedovoľujú ani jednému tímu zhodiť server a tým znemožniť hranie. Preto je našou prioritou, ak sa chceme zúčastniť turnaja, odstrániť tento problém.

Umiestnenie v programe:

Všetky zmeny v programe na vystopovanie chyby, mali charakter komentára, ktorý znemožnil vykonávať určité správanie hráča. Žiadna z týchto zmien nebola trvalo zavedená do programu, išlo o prechodné a testovacie zmeny, ktoré sa potom vrátili naspäť.

Server je robený v dvoch verziách, jedna pre Windows a druhá pre Linux. Moje testovanie prebiehalo nasledovným spôsobom, najprv som sa pokúsil spustiť simuláciu nášho hráča a tímu jahodových princov, no táto simulácia sa ani nezačala, lebo hráč jahodových princov hneď padol. Náš hráč sa tváril spokojne. Preto som hľadal hráča, čo by bežal na novom serveri bez pádu hneď na začiatku a túto podmienku splnil hráč Loptošov.

Spustil som simuláciu hry medzi našim hráčom a Loptošmi. Hráči hrali uspokojivo a všetko vyzeralo v poriadku, no zrazu sa stala prekvapivá vec. Monitor sa zastavil a padli všetci hráči aj so serverom zhruba v 800tom cykle.

Zistil som, že tento pád servera sa prejavuje aj keď nechám hrať náš tím osamote bez protihráčov, ďalej som testoval túto chybu len týmto spôsobom. Žiaľ aj po mnohých hodinách testovania som nedokázal identifikovať problematickú časť hráča, čo spôsobovala pád servera.

Po tomto, som sa spojil s kolegami z konkurenčného tímu a dozvedel som sa, že ich hráč síce beží na linuxovom RoboCup serveri bez pádu, no rovnako padá na windowsovej verzii serveru. Teda má rovnaký problém ako náš hráč. Podobne pôvodný UTTP hráč beží bez pádu na Linux RoboCup serveri. Tento problém sa vyrieši tak, že fakultný RoboCup pobeží na Linux serveri a ani jeden hráč by nemal mať problém. Náš nový hráč ešte musí byť dodatočne otestovaný na Linux serveri. Zaznamenanú chybu pádu servera som odoslal implementačnému tímu RoboCup.

Oprava metódy na zisťovania prítomnosti nepriateľských hráčov v okolí hráča

Metóda dáva do indexu poľa štandardne hodnotu -1 a keď svet vráti prázdne pole o cudzích hráčoch, tak pristupuje na prvok poľa s indexom -1 .

Dôvod:

- Hráči padajú v dôsledku tejto metódy, keď nie je v ich okolí žiaden nepriateľský hráč.

Umiestnenie v programe:

/Player/ModuleManager/Behaviours/OutPlayOpponentBehaviour.cpp

Prekreslenie rozhodovacích stromov s loptou a bez lopty

Prekreslenie rozhodovacích stromov a oprava drobných chýb, na ktoré som narazil pri prekresľovaní.

Dôvod:

- Plánovali sme urobiť zmeny v hlavnom rozhodovacom strome, bolo nutné predtým urobiť analýzu terajších stromov.

Umiestnenie v programe:

/Player/ModuleManager/Behaviours/PlayOnWithoutBallBehaviour.cpp

/Player/ModuleManager/Behaviours/PlayOnWithBallBehaviour.cpp

Prerobenie metódy dribblepoint

Metóda vypočíta optimálny bod na driblovanie

Dôvod:

- Metóda vyrátala, priemernú rizikovosť na každý smer, kam sa mohol hráč s loptou vydať. Zmena bola taká, že nevyrátala priemer, ale maximum z možných smerov a potom sa vybralo minimum. Táto zmena nakoniec nebola implementovaná, pretože ju plánujeme nahradiť.

Umiestnenie v programe:

/Player/Evaluating/EvaluatingPlayerList.cpp

Oprava bugov v umbrella kick mode

Odkopnutie lopty na pseudo-náhodnú pozíciu a center lopty na nesprávnu pozíciu pre súradnicu y.

Dôvod:

- Metóda pristupovala na prvky v poli, ktoré do poľa nepatrili.

Umiestnenie v programe:

/Player/ModuleManager/Behaviours/PassToStrategicPositionBehaviour.cpp

/Player/ModuleManager/Behaviours/PassToRandomUmbrellaPositionBehaviour.cpp

Andrej Škuba

Obzeranie sa hráčov

Princíp, ktorý sme implementovali sa zakladá na nasledujúcej myšlienke. Hráč pred prijatím lopty, hneď po tom, čo sa rozhodne, že loptu, ktorá smeruje k nemu prijme, sa obzrie smerom doľava o 90° a následne doprava o 180° pri najširšom uhle pohľadu a tým vezme na vedomie pozície protihráčov nachádzajúcich sa v jeho okolí. Po tomto úkone sa hráč rozbehne smerom k vyrátanému bodu prijatia lopty. V prípade, že je lopta pred obzeraním sa bližšie ako 6 metrov, tak sa hráč nebude okolo seba obzerať, pretože by sa mohlo stať, že by počas obzretia sa lopta preletela okolo hráča a ten by ju nezachytil. Taktiež ak je lopta vzdialená od hráča viac ako na 15 cyklov, tak sa hráč po 15tich cykloch opätovne obzrie, či sa za ním nenachádzajú protihráči, pretože medzičasom sa mohli za ním objaviť.

Dôvody pre zmenu:

- V doterajšej verzii hráča sa pri zachytávaní lopty stávalo, že po prijatí lopty sa hráč vybral spolu s loptou smerom, v ktorom lopta prichádzala k hráčovi a pritom nezaregistroval, že za ním stoja protihráči. Tí mu následne loptu odobrali a častokrát to bol aj dôvod inkasovana gólu. Naším cieľom bolo túto taktickú chybu odstrániť a vylepšiť tak prehľad hráča o jeho blízkom okolí, najmä protihráčoch.
- Toto správanie zlepšuje prehľad hráča o jeho protivníkoch v okolí a zlepšuje rozhodovanie sa o nasledovnom smere pohybu s loptou tak, aby o túto loptu hráč neprišiel.

Umiestnenie v programe:

Samotný kus kódu obsluhujúci toto správanie bol začlenený do triedy `InterceptBallBehaviour`, konkrétne do metódy `Behave`.

V triede `Myself` sme vytvorili metódy zabezpečujúce prácu s hodnotou premennej, v ktorej udržiavame aktuálny počet krokov k lopte. Ak je tento počet nulový znamená to, že sme pri lopte, prípadne, že sme prekročili 15 krokov na ceste k lopte a že hráč sa musí znova obzrieť. S touto premennou pracujú metódy

- `GetLastLookAroundTime()` - vráti počet cyklov, pred ktorými sa obzrel za seba
- `SetLastLookAroundTime(int steps)` - nastaví počet cyklov, pred ktorými sa obzrel za seba
- `DecreaseLastLookAroundTime()` - zníži počet cyklov, pred ktorými sa obzrel za seba o jeden

Rastislav Kršák

Zlepšenie vykopávania lopty brankárom

Po úprave zdrojového kódu sa berie do úvahy nielen pozícia, odkiaľ sa lopta vykopáva, ale i miesto kam smeruje. Každé dvojici bodov, ktorá je tvorená z miesta výkopu lopty (jeden z niekoľkých bodov nachádzajúcich sa na vertikálnej úsečke v priestore bránkoviška) a miesta, kam lopta smeruje (pozície spoluhráčov) sa priradí váha. Váha vyjadruje vhodnosť výkopu lopty týmto smerom a pri jej výpočte sa berú do úvahy faktory ako uhlová vzdialenosť k najbližšiemu nepriateľovi od miesta výkopu (čím vyššia, tým vyššia je váha), počet nepriateľov v okolí spoluhráča (čím menej, tým vyššia je váha), vzdialenosť k miestu výkopu lopty (čím ďalej, tým lepšie). Na realizovanie výkopu sa potom vyberie dvojica bodov s najvyššou váhou.

Pri vykopávaní lopty na konkrétneho hráča sa navyše zohľadňuje aj vzdialenosť od brankára. Podľa toho sa prispôsobuje rýchlosť (sila) prihrávky a to pomerne k hraničnej vzdialenosti, za ktorú sa prihráva sto percentnou silou.

Z rozhodovania brankára sú pri vykopávaní vylúčení hráči, ktorí sa nachádzajú v priestore medzi brankárom a jeho bránou a hráči, v ktorých blízkosti sa nachádza nepriateľ. Ak nie je možné prihrať ani jednému spoluhráčovi, tak sa aplikujú pôvodné funkcie na vykopávanie lopty, ktoré sa však predtým používali len výnimočne. Konkrétne ide výkop lopty realizovaný použitím funkcií využívajúcich gradientové mapy. Brankár v tomto prípade loptu kopne do priestoru tak, aby lopta nesmerovala na nepriateľa a šla čo najďalej do priestoru ihriska.

Dôvody pre zmenu:

- brankár príliš často vykopáva loptu na rovnakú pozíciu na ihrisku
- brankár vôbec nezohľadňuje aktuálnu situáciu na ihrisku, t. j. rozmiestnenie hráčov pri vykopávaní lopty

Umiestnenie v programe:

Player/ModuleManager/Behaviours/GoalieBehaviours/KickOffBehavior.cpp

Player/ModuleManager/Behaviours/GoalieBehaviours/KickOffBehavior.h

Tomáš Kučečka

Nadstavba modelu sveta

Vytvorenie nadstavby nad svetom hráča za účelom lepšej prehľadnosti hráča o situácii na ihrisku. Hráč si vedie informácie o pohybe svojich protihráčov na ihrisku v podobe kružníc ich možného pohybu v čase. Tieto kružnice sú osekávané pohľadmi hráča tak, ako sa hráč obzerá počas priebehu hry – bolo implementované uchovávanie si pohľadov za určitý počet cyklov.

Ak hráč teda v čase t uvidel nepriateľa na určitej pozícii, v čase $t+1$ sa okolo tohto hráča vytvorí kružnica, ktorá predstavuje jeho možnú pozíciu v danom čase (hráč sa môže vyskytovať v ľubovoľnom bode vnútri kružnice). Táto kružnica sa bude s časom zväčšovať. Hráč môže polomer kružnice orezať na základe obzerania sa po ihrisku. Kalkulácia orezávania prebieha ale až v závislosti od konkrétnej informácie, ktorú chceme vydolovať zo sveta. Napríklad orezávanie má zmysel pri nájdení najvhodnejšieho miesta prihrávky alebo zistenia, či je pozícia voľná.

Dôvody pre zmenu:

- Lepšia informácia o prostredí hry (protihráči v hre)
- Možnosť využitia pri prihrávaní si, driblovaní, atď.

Umiestnenie v programe:

`Player/Player/World/Enemy.cpp`

`Player/Player/World/Enemy.h`

V rámci triedy *Enemy* bola vytvorená podtrieda *Triangle* reprezentujúca pohľad hráča.

Ďalšie zmeny, ktoré si táto implementácia vyžiadala sa týkali doplnenia do triedy *World*. Jednalo sa o vytvorenie inštancie tejto metódy a o aktualizáciu jej dátových štruktúr. Trieda *Enemy* využíva aj existujúce štruktúry, ktoré poskytujú informácie o stave sveta, a to:

- *World->enemies* – Pole reprezentujúce zoznam protihráčov na ihrisku s informáciou o ich polohe, atď.
- *World->friends* – Pole reprezentujúce zoznam spoluhráčov na ihrisku s informáciou o ich polohe, atď.

Prihrávky hráčov

Implementácia nového vyhodnocovania prihrávok medzi hráčmi na základe informácií, ktoré poskytuje vyššie opísaná nadstavba modelu sveta. Výstupom novo vytvorenej metódy je vrátenie pozície, na ktorú by mal hráč prihrať. Kľúčom výpočtu pozície sú informácie o pozíciách spoluhráčov a nepriateľov v hre.

Vyhodnotenie prebieha zistením, či je možné na nejakú pozíciu, kde má hráč spoluhráča prihrať tak, aby k prihrávke nemohol pribehnúť protihráč. Do úvahy sa berú aj viaceré penalizácie za prihrávky smerom vzad, za spoluhráčov, o ktorých máme nepresné informácie a podobne.

Dôvody pre zmenu:

- Hráči používaním pôvodnej funkcie na výpočet cieľa prihrávky prihrávali aj nepriateľom
- Existujúca funkcia na výpočet bodu prihrávky je neprehľadná, slabo zdokumentovaná a ťažko by sa v nej využili informácie

Umiestnenie v programe:

`Project/Player/Player/World/Enemy.cpp`

Konkrétne sa jedná o tieto metódy v rámci triedy *Enemy*:

```
Point Enemy::getPassTargetPosition()  
bool Enemy::canPassSafely(Point& target, float probability)  
float Enemy::getPassProbabilitySuccess(Point& origin, Point& target)  
float Enemy::recalculateEnemyCircleBorders(const Myself* me, const Player*  
enemy, const Point* target)
```

Metóda *getPassProbabilitySuccess(...)* vracia pravdepodobnosť úspešnosti prihrávky medzi dvoma bodmi. Táto metóda je využívaná metódou *getPassTargetPosition(..)*, ktorá vracia bod, na ktorý je navhodnejšie poslať prihrávku na základe ohodnotenia, ktoré obdrží od tejto metódy. Metóda *getPassTargetPosition(..)* využíva aj metódu *recalculateEnemyCircleBorders(...)*, ktorá vracia dĺžku, o ktorú je možné nepriteľa posunúť ďalej od miesta prihrávky. Podrobnejšie informácie o fungovaní týchto metód je možné nájsť v samotnom zdrojovom kóde.

Úprava rozhodovania hráča s loptou

Ide o úpravu správania sa hráča v prípade, ak má pod kontrolou loptu. Úprava sa týkala zjednodušenia rozhodovacieho stromu a odstránenie niektorých zúfalých akcií. Zúfalá akcia znamená, že hráč loptu odkopne náhodne, pretože nevie s ňou nič iné spraviť (žadná iná akcia v rámci stromu nemá splnené podmienky). Ďalšia úprava sa týkala doplnenia rozhodovania sa hráča na základe vytvorenej nadstavby modelu sveta.

Dôvody pre zmenu:

- Snaha o zjednodušenie rozhodovacieho stromu
- Odstránenie nadbytočných a nezmyselných podmienok a akcií
- Integrácia vyššie opísaných vylepšení hráča

Súčasný stav a ďalšia práca

Všetky vyššie uvedené vylepšenia sa testujú na hre tímu Kukuričné deti proti pôvodnému tímu UTPP a taktiež aj na hre proti tímu Princovia.

Výsledky zatiaľ dosahujú slabú kvalitu, pričom ale bolo odstránených už viacero chýb v implementácii. Pre odstránenie ďalších nedostatkov sa plánuje doimplementovať modul pre vizualizáciu sveta. Taktiež sa počas implementácie existujúcej funkcionality objavili ďalšie možnosti rozšírenia.

Jaroslav Bálík

Vhadzovanie lopty vo free kick móde

Ide o opravenie pôvodnej funkcionality hráča pri vykopávaní lopty. Toto vykopávanie lopty nastane vtedy, keď sa lopta dostane mimo hraciu plochu, alebo keď je rozhodcom ohlásený faul.

Dôvody pre zmenu:

- Pri vhadzovaní je hráč číslo 5 určený na to, aby rozohral loptu. Traja ďalší hráči sa postaví tak, aby loptu pri vhadzovaní mohli chytiť. Avšak problém nastáva vtedy, keď hráč číslo 5 loptu nevidí a teda sa k nej nemôže priblížiť.

Umiestnenie v programe:

`/Player/ModuleManager/Behaviours/OurFreeKickWithoutBallBehaviour.cpp`

Konkrétne sa jedná o tieto metódy v rámci triedy:

`bool OurFreeKickWithoutBallBehaviour:behave()`

Na rozohrávanie je aj naďalej určený hráč číslo 5, ale spolu s ním hráč číslo 6 ide po loptu. Hráč, ktorý k lopte príde ako druhý, sa postaví tak, aby prihrávku mohol chytiť.

Vypisovanie štatistiky kouča v release móde

Ide o opravu pôvodnej funkcionality kouča. Kouč vypisuje do súbory `Debug/Timova statistika.txt` súhrnné štatistiky, ktoré sumarizujú priebeh zápasu a správanie sa jednotlivých hráčov.

Dôvody pre zmenu:

- Celkové tímové štatistiky kouč vypisoval iba v debug móde. Debug mód však spomaľoval priebeh zápasu.

Umiestnenie v programe:

`/Coach/CoachKernel.cpp`

Metóda:

`Coachkernel::~Coachkernel()`

Kouč po tejto oprave vypisuje celkové štatistiky v release aj debug móde.