

**Názov semestrálnej práce**

Pong

Vypracoval(a): **Pavol Konečný**

Študijná skupina: **5ZYI36**

Predmet: **Princípy operačných systémov**

Cvičiaci: *Ing. Patrik Rusnák, PhD.*

Obsah

[1. Štruktúra projektu 3](#_Toc124892924)

[2. CMakeLists 3](#_Toc124892925)

[3. client\_game a server\_game 3](#_Toc124892926)

[4. Adresar network 3](#_Toc124892927)

[5. Adresar collision 3](#_Toc124892928)

[6. Kde sme použili vlákna 4](#_Toc124892929)

[7. Synchronizačný problém 4](#_Toc124892930)

[8. Socket medzi clientom a serverom 4](#_Toc124892931)

[9. Problémy, ktoré sme v semestrálnej práci riešili 4](#_Toc124892932)

# Štruktúra projektu

Obrázok, na ktorom je text

Automaticky generovaný popis

# CMakeLists

Obsahuje postup ako buildnut projekt.

# client\_game a server\_game

Obsahujú main funkciu, inicializujú NetworkObject a chod programu ďalej predajú **game**.

**game** obsahuje hlavný chod programu ako je napríklad: vykresľovanie, update, input-handling, atď.

# Adresar network

Obsahuje **client**, **server** a abstraktnú triedu **network\_object** - ktoré predstavujú sieťový prvok. Client aj server implementujú rovnaký interface zadefinovaný abstraktnou triedou network\_object.

# Adresar collision

Obsahuje **dynamic** a **static** **collision** **object**. Obidve tieto triedy implementujú rovnaký interface zadefinovaný **collision**\_**object**. Tieto objekty definujú collision object v hre.

# Kde sme použili vlákna

Použil som jedno hlavné vlákno a jedno sieťové.

Hlavné vlákno rieši všetko. Napríklad vykresľuje hru, spracováva vstup užívateľa, updatuje pohyb.

Sieťové vlákno zase počúva na sockete. Ak príde správa, spracuje ju a updatne polohu hráča.

# Synchronizačný problém

Nenastal žiadny problém. Pretože napríklad premenná ku ktorej pristupujú obe vlákna je **opponentPos.** Hlavne vlákno tuto premennú len číta, a sieťové vlákno prepisuje synchronizačný problém tu nie je potreba riešiť, pretože nas zaujíma len okamžitý stav premennej

# Socket medzi clientom a serverom

Server si vytvorí socket a začne čakať na pripojenie klienta. Client si vytvorí socket a pokúsi sa pripojiť k serveru. Následne obe aplikácie medzi sebou komunikujú.

.

# Problémy, ktoré sme v semestrálnej práci riešili

1. Paralelizácia – Tento problém som vyriešil využitím dvoch vlákien.
2. Vykresľovanie na obrazovku – Na vykresľovanie som použil knižnicu ncurses.
3. Multiplayer – Ten som vyriešil vďaka použitiu socketv medzi clientom a serverom.

Ďalším problémom by sme mohli považovať používanie niekoľkých metód, ktoré fungujú na linuxe. Ale keďže Pong hráme na školskom friose, ktorý beží na linuxe, tak tieto metódy fungujú.

Makefile – Daný problém som nevyriešil , pretože som nenašiel ako importovať knižnice.

Najväčší problém bol , že som na semestrálnu prácu bol sám. Tým pádom som veľa času strávil zisťovaním ako čo spraviť , opraviť a podobne.