

Crazy Coder v0.5

(dokumentácia)

Autor: Juraj Mišúr

Predmet: Programování v Unixu [SWI015]

Prog. jazyk / Vývojové prostředí: C++ / vim :)

Škola / rok: MFF UK v Praze, zimný semester 2005/2006

Web: <http://misur.matfyz.cz/mff/prg020>

Užívateľská časť:

Princíp hry:

Program Crazy Coder je 2D hra na štýl Bombermana (pôvodne Dynablaster). Panák chodí po malej mapke a pokladá bomby, čím odstraňuje debničky v mape, namiesto ktorých sa mu potom môže zobrazit' bonus. Úlohou hry je prebiť sa k protihráčovi cez debničky a zabiť ho výbuchom bomby.

Prostredie:

Za prvé, hra sa dá kedykoľvek vypnúť normálnym zavretím okna, alebo po stlačení ESCAPE sa hra vypne do Menu, prípadne stlačením ESCAPE v Menu sa hra vypne úplne.

Na začiatku sa zobrazí Menu, v ktorom si môže hráč vybrať z troch položiek, a to hra dvoch hráčov na jednom počítači (stlačením klávesy 1), hra po sieti - server (klávesa 2) a hra po sieti - klient (klávesa 3).

Pri hre po sieti s vytvorením servera si hráč môže zvoliť port, cez ktorý chce hrať. Ak iba odklepne ENTER, port bude 99. Nasledovne server čaká, kým sa nepripojí klient.

Pri hre po sieti - klient si hráč najprv zadá IP adresu servera, potom port a čaká, kým sa hra na server nepripojí. To sa nemusí podariť, potom sa hra prepne späť do Menu. Ak sa to podarí, vykukne okno so skóre a na strane servera sa čaká na stlačenie ENTER. Potom sa už začne hra, hráči stoja v protiľahlých rohoch.

Pri hre dvoch hráčov na jednom počítači je ovládanie nasledovné:

Hráč	Modrý	Červený
pohyb doľava	šípka vľavo	klávesa A
pohyb doprava	šípka vpravo	klávesa D
pohyb hore	šípka hore	klávesa W
pohyb dola	šípka dole	klávesa S
položenie bomby	pravý ctrl	ľavý ctrl

Pri hre po sieti každý hráč ovláda svojho panáka šípkami a pravým CTRL.

Programátorská časť:

Program je rozdelený do týchto častí a zdrojákov:

Graphics: Gfx, CSprite, SFont

Main: Main, Defines

CBase: CBaseEntity, CBlock, CBomb, CBonus, CExplosion, CMap, CPlayer, CTile

Network: Client, Server, Network

Základný popis fungovania:

Vo funkcii **main** je switch, ktorý volá funkcie rôznych častí hry podľa nastavenia globálnej premennej **gameState**. Túto premennú menia funkcie podľa aktuálneho stavu hry a užívateľovho vstupu. Na začiatku je premenná na hodnote **MENU**, čo znamená že sa volá príslušná funkcia, ktorá zobrazí Menu a výber hry. Ak hráč zvolí hru po sieti (stav **CONNECT**), zavolá sa funkcia **Connect**, ktorá sa posnaží nadviazať spojenie s klientom/serverom. Ak je spojenie nadviazané, hra zmení stav na **NEW_ROUND** a znova sa cez **main** zavolá príslušná funkcia **NewRound**. Tá funkcia iba čaká na odklepnutie ENTER a zmení stav na **GAME**, pričom potom sa už spustia funkcie **GameCoreSingle** alebo **GameCoreMulti** v závislosti na typu hry.

V samotnej **GameCore** sa najprv inicializujú premenné hry. Hra je robená pre pevný počet FPS, v jednom frame sa zhodnotí vstup z klávesnice, potom sa zisťujú stavy samotných objektov v hre (bomby, debničky, explózie, hráči) a na základe toho sa vykonajú nejaké dôsledky.

Sieťový kód:

Na fungovanie siete sa používa `<winsock2.h>` pre Win a `<sys/socket.h>` pre Unix . Všetky blokujúce sieťové funkcie (**accept**, **connect**, **recv**) sú oddelené od hlavného procesu vo vlastnom vlákne **NetThread**. Ten zabezpečuje podľa stavu hry (**gameState**) spojenie, prípadne prijímanie packetov. Ak príde packet počas samotnej hry, **NetThread** zablokuje (pomocou **mutexu**) **deqPacket** (linked list obsahujúci prijaté packety) a vloží doň prichodzí packet. Následne **deqPackets** odblokuje. Samotná fcia **GameCoreMulti** každý frame na strane servera vykoná zmenu stavu lokálneho hráča, pošle klientovi túto zmenu a ak prišiel packet od klienta, tak mu odošle späť ten istý packet a vykoná zmenu stavu vzdialeného hráča.

Klient pri zmene pohybu hráča pošle packet serveru, ale nevykoná zmenu stavu svojho hráča. Zmena stavu oboch hráčov sa na klientovi vykonáva až po obdržaní packetov zo servera (to znamená že sa zamkne **deqPackets**, pozrie sa, či niečo došlo; ak áno tak vykoná zmenu stavu hráčov na základe hodnôt v packete, ktorý obsahuje aj hodnotu o ktorého hráča to vlastne ide; nakoniec sa **deqPackets** odomkne).

Programové časti:

V každom `<source>.cpp` sú definované funkcie objektu, ktorý je deklarovaný v `<source>.h`

Graphics:

Gfx - grafické funkcie, inicializácia SDL, nastavenie rozlíšenia obrazovky

CSprite - objekt typu sprite (tzn. objekt, ktorý obsahuje nejakú grafickú bitmapu), funkcie na vykresľovanie a zadávanie parametrov

SFont - funkcie na vykresľovanie písma, viď poznámka

Main:

- Main** - základný switch spomenutý vyššie, ďalej funkcie ako **Menu**, **NewRound** (výpis skóre a čakanie na enter), **Connect** (volanie **GetIP** a **GetPort**, slúži na nadviazanie spojenia), **DrawXXX** (vykresľovanie rôznych vecí), **GameCoreSingle** (jadro lokálnej hry), **GameCoreMulti** (jadro hry po sieti, správanie servera/klienta), **NetThread** (vlákno pre ovládanie spojenia - nadväzovanie spojenia, prijímanie packetov, kontrola spojenia)
- Defines** - rôzne **enum** a **#define**, napr. **GAME_STATE** {MENU, CONNECT, NEW_ROUND, GAME, GAME_QUIT}, **PACKET** {GAME_STATE, PLAYER, MOVE, PLANT}

CBase:

CBaseEntity - základný objekt, z ktorého dedia všetky ostatné **CXXX** objekty

```
class CBaseEntity
{
public:
    CBaseEntity();
    CBaseEntity(int posX, int posY);
    virtual ~CBaseEntity();
    virtual ENTITY_STATUS Think();           //vykonanie zmeny pohybu/akcie/stavu
    ENTITY_STATUS Status();                 //vráti stav objektu
    CSprite *GetSprite();                   //vráti sprite, ktorý objekt obsahuje
    void SetSprite(CSprite *tmp);           //uloží sprite do objektu
    int GetMapX();                          //poloha objektu v mierach mapy
    int GetMapY();
    int GetX();                             //poloha objektu v absolútnych mierach
    int GetY();

protected:
    CSprite *spr;                           //sprite objektu
    int x, y;                               //absolútna poloha
    int w, h;                              //výška, šírka
    int mapX, mapY;                         //poloha v mape
    ENTITY_STATUS status;                   //stav objektu (alive/deadanim/dead)
    int animFrame;                          //frame animácie
    int animTick;                          //pomocný frame
};
```

- CBlock** - debnička; v prípade úmrtia vygeneruje nejaký bonus
- CBomb** - bomba; v prípade, že dotiká (umrie), vygeneruje explóziu na základe polohy v mape a dĺžke plameňa
- CBonus** - bonus; môže byť **BOMB/SPEED/FLAME**
- CExplosion** - časť celkovej explózie zaberajúca jedno políčko mapy
- CMap** - mapa; obsahuje funkcie na určovanie, kde sa nachádzajú objekty na mape a operácie s nimi, základ je jedno políčko o veľkosti **TILE_SIZE**
- CPlayer** - objekt panák; funkcie **Thnik** a **NetThink** na určovanie stavu, zmenu

pohybu atp.

C Tile - políčko mapy, cez ktoré sa nedá prejsť panákom

Network:

Client - spustenie a ukončenie klienta (**startupClient**, **shutdownClient**)

Server - spustenie a ukončenie servera (**startupServerForListening**, **shutdownServer**)

Network - objekt **C Packet**, ktorý obsahuje buffer s údajmi, ktoré sa posielajú po sieti

Poznámka:

Hra je programovaná tak, aby behala pod Windows a Unix. Na pohodlnú kompiláciu pod Win je priložený solution pre Visual Studio .NET 2003. Unixáci použijú Makefile.

Tento program je distribuovaný s knižnicami a zdrojovými kódmi SDL (<http://www.libsdl.org>), licencia **LGPL**

SFont.c a SFont.h sú zdrojové kódy SFontu (<http://www.linux-games.com/sfont/>), licencia **GPL**