Sieťová hra Bulánci

Programátorská dokumentácia

Dominik Bobák

Marek Gajdoš

5ZYP31

Anastasiia Tsipushtanova

5ZYP32

SÚBORY:

Bullet.h, Bullet.c

Štruktúry:

1. BULLET\_STRUCT

* int x – aktuálna súradnica vertikálna
* int y – aktuálna súradnica horizontálna
* enum Direction direction – aktuálny smer pohybovania
* int shooterId – id postavy ktorá vystrelila

Funkcie verejné:

1. void MoveBullet(BULLET\_STRUCT\* bullet, enum Direction direction) – Posúva strely podľa zadaného smeru

Client.h, Client.c

Štruktúry:

1. SOCKET\_DATA
   * int sockfd - socket
   * struct sockaddr\_in serv\_addr – adresa servera: dĺžka, port, hostname
   * struct hostent\* server – informácia o danom hostovi: názov, ip-adresa atď.
2. GAME\_DATA
   * SOCKET\_DATA \* socketData
   * MAP\* map – hracie pole s hráčmi
   * bool gameStarted – kontrola štartu hry
   * bool isRunning – kontrola ci hra pokračuje
   * const int id – odkaz na SOCKET\_DATA
   * pthread\_mutex\_t \* gameMutex – možnosť blokovať pristúp pre iné procesy
   * pthread\_cond\_t \* startGame\_c – kontrola kondície

Funkcie verejné:

1. void init\_Client() – inicializuje klienta a spojenie so serverom;

Funkcie pomocné:

1. void ShowMap(MAP\* map) - vykreslí mapu v GUI
2. CLIENT\_MESSAGE GetClientMessage() – vráti smer hráča/strely podľa výberu používateľa
3. void\* Thread\_ClientSend(void\* data) – funkcia posiela od klienta request na server, vráti NULL
4. void\* Thread\_ClientReceive(void\* data) – funkcia prijíma odpoveď od servera, vráti NULL
   * void\* data – vstupne GAME\_DATA
5. int ConnectToServer - pokúsi sa pripojiť na server a vráti id klienta zo serveru

Common.h, Common.c

Definované typy:

1. MAX\_PLAYERS 4 – maximálny počet hráčov
2. MIN\_PLAYERS 2 – minimálny počet hráčov
3. MAX\_BULLETS 50 – maximálny počet
4. DEFAULT\_PORT 10025 – pôvodne číslo portu serveru
5. GAME\_TIME 600 – trvanie hry
6. SCREEN\_WIDTH 500 – šírka okna
7. SCREEN\_HEIGHT 500 – výška okna
8. WALL\_THICKNESS 10 – hrúbka rámu
9. CELL\_WIDTH 20 – veľkosť postavy
10. BULLET\_SIZE 5 – veľkosť strely

Štruktúry:

1. CLIENT\_MESSAGE
   * enum Direction direction – zapíše smer objektu
   * enum MessageType type – zapíše typ objektu: postava ci strela

Funkcie verejné:

1. int GetPort() – vráti číslo portu pripojenia
2. char\* GetHostname() – vráti adresu serveru
3. int getIntFromCharPtr(char \* arg) – prekonvertuje číslo portu zo stringu na číslo.

Enums.h, Enums.c

ENUMS:

1. enum WeaponType
   * PISTOL
2. enum KeyType
   * UNDEF,
   * LEFT\_ARROW,
   * RIGHT\_ARROW,
   * UP\_ARROW,
   * DOWN\_ARROW,
   * SPACEBAR,
   * BACKSPACE,
   * ESCAPE
3. enum Color
   * YELLOW,
   * BROWN,
   * GREEN,
   * ORANGE
4. enum Direction
   * UP,
   * DOWN,
   * LEFT,
   * RIGHT
5. enum MessageType

* PLAYER,
* BULLET,
* CANCEL,
* UNKNOWN

1. enum ResponseType
   * ACK,
   * WAIT,
   * PLAY,
   * UPDATE,
   * STOP,
   * ERR

Funkcie verejné:

1. char\* WeaponType\_toString(enum WeaponType type) – prekonvertuje hodnotu Enum na reťazec
2. char\* Direction\_toString(enum Direction type) – prekonvertuje hodnotu Enum na reťazec
3. char\* MessageType\_toString(enum MessageType type) – prekonvertuje hodnotu Enum na reťazec
4. char\* ResponseType\_toString(enum ResponseType type) – prekonvertuje hodnotu Enum na reťazec

GUI.h, GUI.c

ENUMS:

1. Game\_State
   * NOT\_PLAYING = 0,
   * PLAYING = 1,
   * PAUSED = 2,
   * GAME\_OVER = 3

Štruktúry:

1. Game:
   * SDL\_Renderer \*renderer – používa sa pri vykresľovaní okna
   * SDL\_Window \*window – chráni podmienky okna
   * int running – kontrola behu hry
   * Game\_State state – status hry
   * MAP\* map – aktuálna mapa na vykresľovanie
   * SDL\_Event event – používa sa pre zisťovanie stlačenia klávesy

Funkcie:

1. void draw\_map(Game\* game) – vykreslí mapu
2. void update\_title(Game\* game,char \* message) – obnoví hlavičku okna
3. int getIntFromInputBox() – prekonvertuje text vyskakovacieho okna na int
4. void initialize(Game\* game) – inicializuje parametre hry
5. void terminate(Game\* game, int exit\_code) – zruší hracie okno
6. void sdl\_ellipse(SDL\_Renderer\* r, int x0, int y0, int radiusX, int radiusY) - vykreslí elipsu pre postavu
7. enum KeyType handle\_input(Game\* game) – kontroluje ci je niečo načítane z klávesnice
8. void draw\_player(Game\* game, PLAYER\_STRUCT \* player) – vykreslí postavu
9. void draw\_bullet(Game\* game, BULLET\_STRUCT \* bullet) – vykreslí strelu
10. void draw\_walls(Game\* game) – vykresli rámik okna

main.c

Funkcie:

1. int main(int argc, char \*argv[]) – ak sú zadane argumenty, spusti server, ak nie sú zadane, spusti client

Map.h, Map.c

Štruktúry:

1. MAP

* PLAYER\_STRUCT players[MAX\_PLAYERS] – informácia o všetkých hráčoch
* BULLET\_STRUCT bullets[MAX\_BULLETS] - informácia o všetkých strelách
* int numberOfPlayers – aktuálny počet hráčov
* int numberOfBullets - aktuálny počet striel
* const int mapY – výška mapy
* const int mapX – šírka mapy
* enum KeyType lastKey – posledná stlačená klávesa

Player.h, Player.c

Definované typy:

1. MAX\_LIVES 5 – maximálny počet životov

Štruktúry:

1. PLAYER\_STRUCT
   * int x – aktuálna súradnica vertikálna
   * int y – aktuálna súradnica horizontálna
   * int lives – aktuálny počet životov
   * bool isAlive – kontroluje či hráč je aktívny
   * enum Direction – aktuálny smer pohybovania
   * int hits – počet zásahov
   * int id – id hráča

Funkcie:

1. void MovePlayer(PLAYER\_STRUCT\* player, enum Direction direction, int maxX, int maxY, int offset) – posúva pozíciu hráča na poli
   * PLAYER\_STRUCT\* player – informácia o hracovi
   * enum Direction direction – smer sa v ktorom hráč posúva
   * int maxX – maximálna hodnota x
   * int maxY – maximálna hodnota y
   * int offset – posunutie od kraju

Server.h, Server.c

Štruktúry:

1. BINDING\_DATA
   * SOCKET\_DATA \* socketData – údaje procesov
   * bool waitingForConnection – kontrola ci čaká na spojenie
   * bool gameStarted – kontrola štartu hry
   * pthread\_mutex\_t \* bindingMutex – možnosť blokovať pristúp pre iné procesy
   * pthread\_cond\_t \* playerConnected\_c – podmienková premenná že hráč bol pripojený
   * pthread\_cond\_t \* connectPlayers\_c – podmienková premenná pre pripojenie hráčov
2. SOCKET\_DATA
   * int sockfd – popisovač socketu
   * int cli\_sockfd[MAX\_PLAYERS] – pole popisovačov
   * socklen\_t cli\_len[MAX\_PLAYERS] – pole veľkostí popisovačov
   * bool validSockets[MAX\_PLAYERS] – kontrola či je socket platný
   * struct sockaddr\_in serv\_addr – adresa socketu na serveri
   * struct sockaddr\_in cli\_addr[MAX\_PLAYERS] – adresy klientov
   * int connectedClients – počet pripojených klientov
3. GAME\_DATA
   * SOCKET\_DATA \* socketData – informácia o sockete
   * MAP\* map – hracie pole s hráčmi
   * bool isRunning – kontrola ci hra pokračuje
   * pthread\_mutex\_t \* gameMutex – možnosť blokovať pristúp pre iné procesy
   * pthread\_cond\_t \* startGame\_c – podmienková premenná k štartu hry
4. RECIPIENT\_DATA
   * GAME\_DATA \* gameData – údaje príjemcu
   * int id – id príjemcu

Funkcie verejné:

1. void init\_Server() – inicializuje server a spojenie s klientom

Funkcie pomocné:

1. int CreateBinding(SOCKET\_DATA \* socketData, int port) - spraví pripojenie na základe údajov socketu a čísla portu
2. void SetupPlayers(MAP \* map) – nastaví hráčov na plochu
3. int StartGame(SOCKET\_DATA \* socketData) – spusti hru pre aktuálny počet hráčov
4. void\* Thread\_ServerSend(void\* data) – funkcia posiela od servera request klientovi, vráti NULL
5. void\* Thread\_ServerReceive(void\* data) – funkcia prijíma odpoveď od klienta, vráti NULL
6. void MoveBullets(MAP\* map) – funkcia posúva strely na hracom poli
7. void StartRequests(BINDING\_DATA \* bindingData) – čaká na pripojenie od klientov, a keď sa pripoja, označí že hra začala
8. void \* Thread\_ConnectPlayers(void \* data) – spracuje pripojenie klientov na server