

MoToX - programátorská dokumentace

Kateřina Nevolová
katka.nevolova@gmail.com
73502469

11. dubna 2012

MoToX je jednoduchá hra pro linux napsaná v jazyce C++ za pomoci knihoven GLUT a OpenGL pro grafiku. Program se sestává z několika modulů, z nichž se většina stará o nějaký konkrétní subsystém. Veškeré napojení na GLUT a hlavní cyklus programu je popsán v modulu `main`, který se dále odkazuje na game obstarávající herní logiku. Většina modulů pak sdílí společný formát:

- inicializační a deinicializační funkci
- funkci pro posun herního stavu o nějaký časový úsek
- případně funkci pro vykreslení pomocí OpenGL
- lokální data uložená v globální proměnné, většinou nějakém kontejneru

1 Soupis modulů

1.1 `main.cpp`

Obsahuje funkci `main` s GLUTovými funkcemi pro vytvoření a obsluhování OpenGL okna apod., sbírá a pamatuje si klávesnicový vstup. Snímá přesný čas hry a předává ho dál herní logice do modulu `game`. Periodicky vyvolává překreslení scény.

1.2 `game.cpp`

Inicializuje (a při vyvolání nového levelu resetuje) všechny součásti hry. Pamatuje si čas a status hry a zvolenou mapu. Funkcí `init_map` komunikuje s modulem `maplist` funkcí `maplist.get_file_name`, podle vybraného mapového souboru potom načte level funkcí `read_map_file`. Funkce `draw_game` kreslí celou hru a je přepínána podle statusu hry (hráč je právě v menu, ve hře, vyhrál, či prohrál). Dále komunikuje s modulem `text` funkcemi `draw_string` a `text.length` pro výpis statistik a jiného textu hry a jeho pohodlné zarovnávání. Funkce `update_game` posune všechny herní součásti o specifický časový interval a zpracovává klávesnicový vstup z modulu `main`. Udržuje souřadnice kamery (hráč

je stále ve středu hracího pole). V případě hráčovy smrti (`moto_out_of_map`, `moto_touches_enemy` - funkce z modulu `moto`) či vítězství (funkce `enemies_left` - funkce z modulu `enemy`) dává možnost spustit hru znovu, nebo se hrou dobrovolně skončit.

1.3 `moto.cpp`

Modul `moto` se stará o samotnou motorku, kreslí ji (funkce `draw_moto`) a udržuje parametry jejích kol a hlav. Funkce `update_moto` slouží k ovládání samotného stroje, je volána z modulu `game` a zpracovává konečný klávesnicový vstup. Pomáhá si funkcí `rotate` (odtud) a `spring` (z modulu `ball`) pro snadnější vybalancování a soudržnost kol a řeší kolize kol s okolím funkcí z `map.cpp` `collide_ball_with_walls`. Komunikuje s modulem `bullets` funkcí `add_bullet`. Funkce `get_moto_position` a `moto_out_of_map` jsou volány opět z `gamu` a slouží pro účely kamery a pro kontrolu pozice motorky. Funkce `moto_touches_enemy` zjišťuje pomocí funkce z `enemy.cpp` `ball_touches_enemy` zda nedošlo ke kolizi s nepřáteli, popřípadě zavolá `big_explosion` z modulu `particles`.

1.4 `enemy.cpp`

List `enemies` uchovává pozici, rychlost, životy a další parametry nepřátel. Funkce `move_enemies` posunuje nepřátele o časový tik, nechá je kolidovat s prostředím funkcí `collide_ball_with_walls` a pokud je někdo z nepřátel mimo mapu, smaže ho. Funkce `draw_enemies` nepřátele kreslí. Kolize se střelami hráče jsou ošetřovány funkcí `try_to_damage_enemy`, která nalezne cíl a případně smaže nepřítele a udělá výbuch (`big_explosion`) z partikul. Funkce `ball_touches_enemy` zjišťuje kolizi kolečka motorky a nepřátel (volána z modulu `moto.cpp`). Funkci `enemies_left` používá modul `game`, aby zjistil, zda už hra neskončila vítězstvím.

1.5 `bullets.cpp`

Obhospodařuje list s pozicí, rychlostí a způsobovaným poškozením střel. Funkcí `update_bullets` je updatuje, nechává vykreslovat (`add_particle`), koliduje se stěnami funkcí `collide_ball_with_walls` z modulu `map`, popřípadě nechá vyrobít `small_explosion`. Maže střely pokud jsou pryč z mapy (`out_of_map`) nebo pokud zasáhly nepřítele (`try_to_damage_enemy`).

1.6 `ball.cpp`

Modul `ball` se stará o hru na té nejnížší úrovni. Funkcí `collision` (volanou z modulu `map`) se stará o kolizi kuličky s libovolnou přímkou - napomáhá si funkcí `normalize` a `wall_collision`, která už konečně řeší veškeré přepočítání pozic a rychlostí při kolizích. Funkce `spring` volaná z modulu `moto` se stará o přitažlivost kuliček a pružení.

1.7 map.cpp

Modul map udržuje a vykresluje list stěn a překážek. Funkce `collide_ball_with_walls` je volána téměř ze všech modulů hry a stará se po zavolání funkce `collision` z `ball.cpp` o kolize kola s mapou. Funkce `out_of_map` udržuje hranice mapy pro hru podobně jako funkce `map_bottom`.

1.8 maplist.cpp

Tento modul inicializuje (funkcí `maplist_init`) a spravuje adresář map, ověřuje zda jsou soubory v adresáři maps skutečně mapami (funkcí `process_filename`) a pokud ano přidává je do vektoru map. Udržuje počet map. Funkcemi `maplist_get_name` a `maplist_get_file_name` pomáhá modulu game pohodlně načítat požadované mapy a vykreslovat interface.

1.9 particles.cpp

Modul pro kreslení částicového systému (explozí apod.). Všechny částice jsou uloženy v listu, každá má svoji pozici, rychlost (směr), barvu, trvanlivost a jeden ze tří předem určených tvarů. Partikly jsou zde kresleny a updatovány. Různé moduly hry volají funkce `small_explosion` a `big_explosion` pro různé typy explozí.

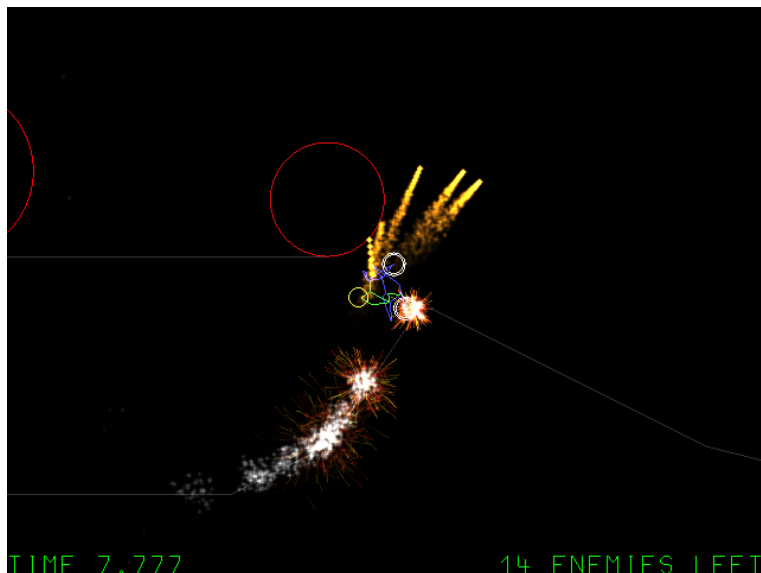
1.10 text.cpp

Jednoduchý nástroj pro kreslení a zjištění délky textu, font je ručně poskládaný z vektorů.

2 Kompilace a spuštění hry

Ke kompilaci jsou potřeba knihovny GLUT a OpenGL. Program se sestaví pomocí Makefile příkazem `make`. Skompilovaný program je poté připraven ke spuštění příkazem `./motox`.

3 Screenshot s popisem



Motorku ovládáme šipkami, mezerníkem a klávesou x. Stroj vykresluje modul moto. Modul game pomocí fontů z modulu text kreslí informace o hře (čas hry a počet nepřátel, které je potřeba eliminovat) v dolní části okna. Výbuchy, kouř a střely obhospodaruje modul bullets a vykresluje modul particles. Šedé plošinky pochází z modulu map a kreslí je game. Nepřátelské objekty reprezentují červené kružnice, ke kterým není radno se přibližovat.