

Lisp in Summer Projects Submission

Submission Date	2013-10-23 18:22:49
Full Name	Kálmán Kiss
Country	Hungary
Project Name	tubegame
Type of software	gui app
General category	game
LISP dialect	Common Lisp
GitHub URL	https://github.com/kiskami/tubegame
Did you start this project?	Yes, all the code is written by me
Project Description	I want to describe my project in this form.
Purpose	tubegame is a simple 3D space shoot'em up for Windows, written in Common Lisp (Clozure CL). The player commands a little space ship in a tube like level (labyrinth), and must destroy or bypass asteroids, and find the game ending ring.
Function	Its a simple 3D game using the Ogre 3D rendering engine, Bullet physics engine for collision detection, and OIS library for input handling.
Motivation	<p>Im a professional developer for about 13 years. In my daily job Im involved in developing JEE applications, security, electronic signature, and other IT fields. Game development and discovering and learning new programming languages has been always my hobby and "chill out" activity. :)</p> <p>Last year, I began to learn Common Lisp using Touretzkys excellent book, but other projects took my attention, and I couldnt proceed. I stumbled into "List in Summer Projects" in an rss digest from "Lambda the Ultimate" site somewhere around mid Aug. This gave me the impulse to continue with the book, and develop some significantly sized project</p>

in CL.

Audience

This project is the proof for myself, that I learnt CL to a level, that I can handle the project organization problems (asdf, packages, executable build) and typical technical problems in game development (main loop, entity system, rendering engine integration, physics, input, performance). Maybe it'll be usefull for other CL novices or curious nonlispers.

Methodology

The integration of Ogre 3D, Bullet and OIS is realized through llgs-engine project. This is a (on some places not so) thin C wrapper library for the C++ libraries. llgs-engine-cl is the CL cffi interface api module for it.

tubegame main entry point is tubegame:game-run in tubegame.lisp. This function initializes llgs-engine first, and then runs the main loop, with a simple two state gamestate (startscreen, game). Globals.lisp has the global constants and parameters, and the defstructs for the main game types (leveldata, entitydata).

The one-game-frame function (game.lisp) is called by the main loop, every frame. It loads and inits the level and player data, listens to input, performs collision detection and triggers game entity update.

Level data is stored on disk in Lisp readable format, in a lisp list. Game space is described as XZ planes in ascending Y order. Every (x,y,z) integer coord space-cube either has a tube element or nothing (nil). Tube elements are cubes with 1, 2 (tube), 3, 4, 5, 6 vents, and optionally contain asteroids or the game ending ring. Cubes can be rotated around X, Y, Z axes too. For example "(2 :rotx 90 :ast3 2 :ast1 1)" is a type 2 cube (tube) with rotation 90 deg, around X axis, and there are 2 type 3 asteroids and 1 type 1 asteroid.

The update-player function (player.lisp) delegates the tasks as movement, firing and hud update to subfunctions. The game-over function displays some statistics, and ends the game.

Asteroids (asteroid.lisp) are dumb entities, that can be damaged and explode eventually. Bullets (bullet.lisp) are forward flying objects, that die on collision and damage asteroids.

Utils.lisp contains some math, string formatting, entity list and physics object->entity map manipulation functions.

Conclusion

This isnt an OO style project, CLOS is not used, and needs study on my side. I didnt

strive to design a pure functional architecture and functions either. An OO rewrite is nearly trivial, given that all entity manipulation functions get the entity instance as parameter.

Collision detection has some issues, and game needs enrichment with more game elements, like gates, firing turrets, enemy ships, powerups, etc. Game needs more levels implemented, gameplay tweaks, balancing, multiplayer mode, etc.

Build Instructions

Requirements:

- Windows 7 or newer
- Clozure Common Lisp 1.8 or newer
- asdf loadable llgs-engine-cl library

This project needs llgs-engine-cl (<http://code.google.com/p/llgs-engine-cl/>).

llgs-enigne-cl is the Common Lisp cffi interface library to llgs-engine.

It provides a simple and low level api for rendering 3D scenes, handling input, collision detection, etc in Common Lisp.

llgs-engine (<http://code.google.com/p/llgs-engine/>) is a low level C wrapper around the Ogre 3D rendering engine, OIS input and Bullet physics engine.

Get the source code from <http://code.google.com/p/tubegame/> (with Mercurial) or <https://github.com/kiskami/tubegame> (git).

To build the executable binary, run the following command in source directory:

```
[YOUR CCL INSTALLATION]wx86cl.exe -l make-exe.lisp
```

Test Instructions

There are no unit tests, just play it. :)

Execution Instructions

Requirements:

- Windows 7 or newer
- DirectX 9 or higher/OpenGL
- 3D Hardware Accelerator

Download the binary distribution with dependency dlls and game assets from <http://www.filedropper.com/tubegame-01>

With the binary package unzipped, change directory to bin and run tubegame.exe, or run ccl, load the asdf package and run main function:

```
tubegamebin>cclwx86cl.exe
Welcome to Clozure Common Lisp Version 1.9-r15764
(WindowsX8632)!
? (require "tubegame")
"tubegame"
NIL
? (tubegame:game-run)
```

**** Notes

The game must be run from a directory (bin), where the

Ogre dlls, config files
and lgs engine dll reside!

This program requires some Microsoft Visual C++ 2010
runtime components.

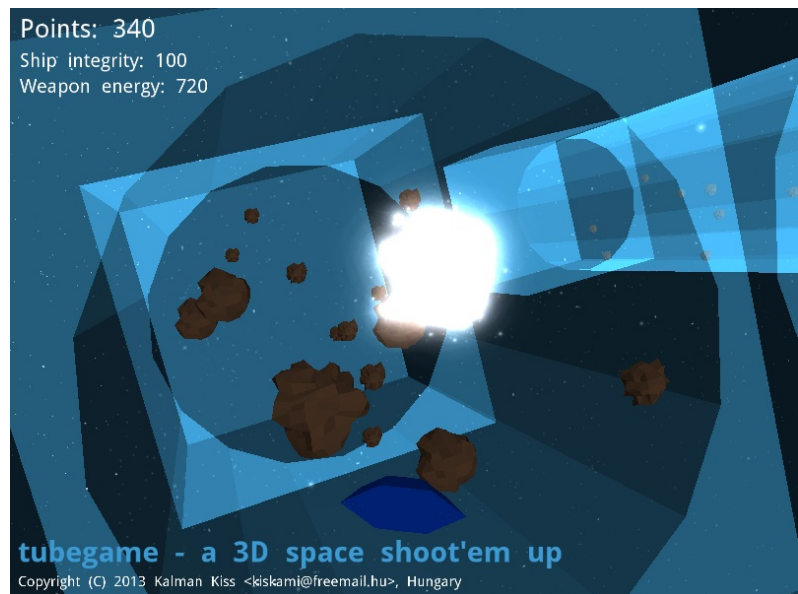
If you get a MSVCP100.dll missing error, please download
and install the
redistributable package from Microsoft:
<http://www.microsoft.com/en-us/download/details.aspx?id=5555>

Describe any bugs or caveats

- Command line argument parsing doesnt work.
CCL:*UNPROCESSED-COMMAND-LINE-ARGUMENTS*
seems
to be empty, when running the saved exe.
- Collision detection and handling is not perfect. Its possible
to leave the level, when
stuck on wall, and sometimes its impossible to escape from
asteroid when collided.

- Only a few gameplay elements implemented: level building
blocks, asteroids,
bullets, explosions, simple hud, game ending ring.
- Demo level is large, its nearly impossible to find the game
ending ring. :)

Screen shots



[tubegame-screenshot-09262013_233854846.png](#)

□
[tubegame-screenshot-09302013_003521728.png](#)

Official

I have read rules and have abided by them.
I am 18 years of age or older.
I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran,
Myanmar (Burma), North Korea, Sudan, or Syria.