

B.Sc Computer Games Development, Year 4, FYP

Technical Design Document

Methods to mitigate latency in online games

Authors: Greg Cahilll
Project Supervisor: L.Shi
03/05/2020

Faculty of Science

Open-Book and Remote Assessment Cover Page

Student Name: Greg Cahill

Student Number: C00216328

Lecturer Name: Dr Lei Shi

Module: Project II (fyp)

Stage/Year: Fourth Year

Date: 03/05/2020

Declaration

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

CRC

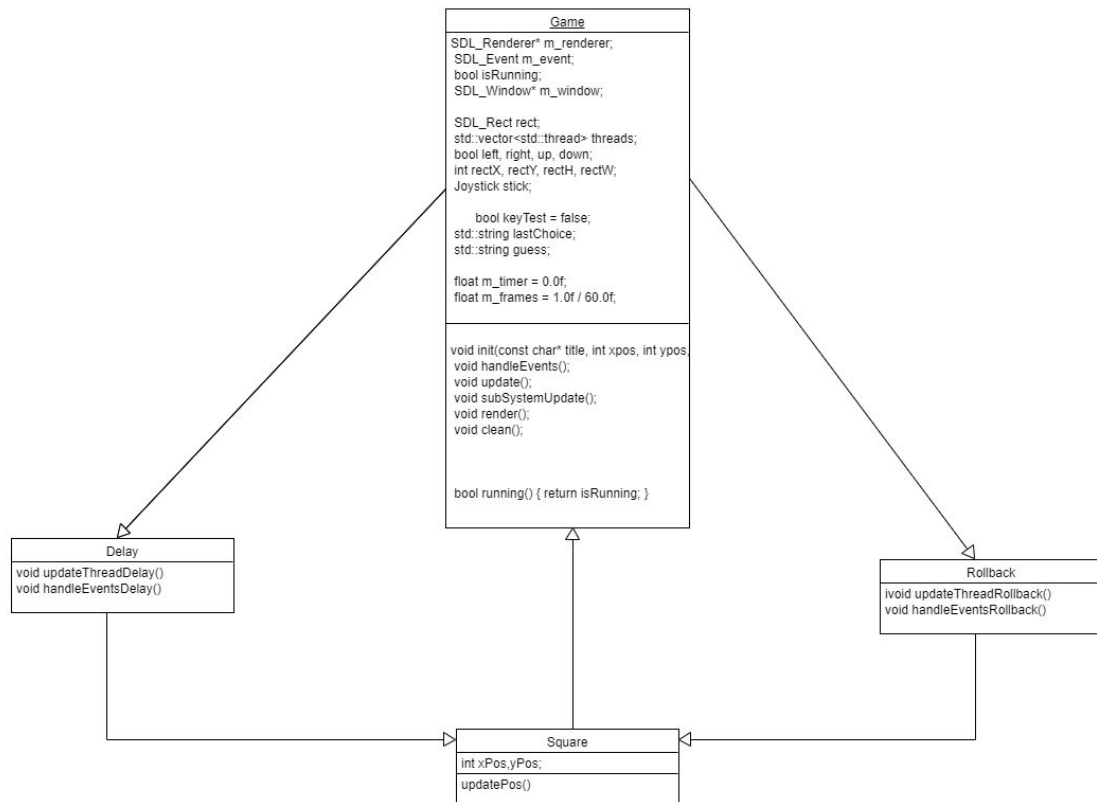
1

Game	
<ul style="list-style-type: none">• Responsibilities• Handles drawing to screen• handles input• updates variables as needed• closes the game when appropriate key is pressed	<ul style="list-style-type: none">• Collaborators• Delay• Rollback

Delay	
<ul style="list-style-type: none">• Responsibilities• Updates square position every x amount of milliseconds• handles arrow key input passed from game• handles running the two delay threads.	<ul style="list-style-type: none">• Collaborators• Game

Rollback	
<ul style="list-style-type: none"> Responsibilities Updates square position handles arrow key input passed from game handles running the two Rollback threads. handles dealing with guessing players next input based on previous Resets players position x frames when the guess is wrong. 	<ul style="list-style-type: none"> Collaborators Game

Architecture



Feature Design

Feature 0(setUp)

- Task 1: Create local and github repository.
- Task 2: Push repository to remote.
- Task 3: Include link to github on drive and invite team members.
- Task 4: Class templates.

Feature 1(Create small game on SDL)

- Task 1: Draw a rectangle shape.
- Task 2: Append rectangle with sprite.
- Task 3: Implementation to move in the cardinal directions.

Feature 2(Delay based)

- Task 1: Create thread to handle key presses
- Task 2: Create thread to update square position
- Task 3: Lock thread for variable amount of Milliseconds using sleep_for

Feature3(Rollback Based)

- Task 1 : Draw a line originating from the players position to a random point
- Task 2 : Stop tip of line when it collides with terrain
- Task 3 : Allow that tip to act as a fulcrum point to allow the player to swing
- Task 4 : Allow all points of the line to collide with terrain for realistic effect

Technology

External Libraries

SDL:

Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D. It is used by video playback software, emulators, and popular games including [Valve's](#) award winning catalog and many [Humble Bundle](#) games.

SDL officially supports Windows, Mac OS X, Linux, iOS, and Android. Support for other platforms may be found in the source code.

SDL is written in C, works natively with C++, and there are [bindings available](#) for several other languages, including C# and Python.

SDL 2.0 is distributed under the [zlib license](#). This license allows you to use SDL freely in any software.

I chose this particular library as we have used it in the past to develop games with online components, such as a team project known as ARGO, and SDL is very easily portable to other pieces of hardware, if it ever came to that.

Installation

[SDL Link](#)

SDL will be installed by linking the project with sfml SDL2, and including all the appropriate includes in the game.h, which can be used be the entire projects files (also add the developmental libraries from sdl_image, as it are needed to render objects to the window)

Technical Achievement

In terms of technical achievement, there wasn't a whole lot pushing boundaries, but i'm happy with the attempt I made to try to simulate an online game in an offline environment through the use of threading. Simple, but got the point across.

References

Referenced Publication	Citation	Reference
Study	(An analysis of continuous consistency models in real time peer-to-peer fighting games 2019)	Authors - Martin Huynh and Fernando Valerino, An analysis of continuous consistency models in real time peer-to-peer fighting games 2019 http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1322881&dswid=-395
Website	GGPO rollback netcode	https://github.com/pond3r/ggpo

Website	Making a multiplayer shooter in c++	http://www.codersblock.org/blog/multiplayer-fps-part-1
---------	-------------------------------------	---