Piattaforme Software per la Rete
Course Projects, part 1

Giovanni Agosta

Piattaforme Software per la Rete – Modulo 2

Outline

Rules of the Game

Short-term goal This is the goal that must be reached in order for the project to be evaluated.

Long-term goal While the long term goal needs not be reached within the project timeframe, the documentation and planning should reflect the goal, making it so that future development do not need to reengineer completely the application.

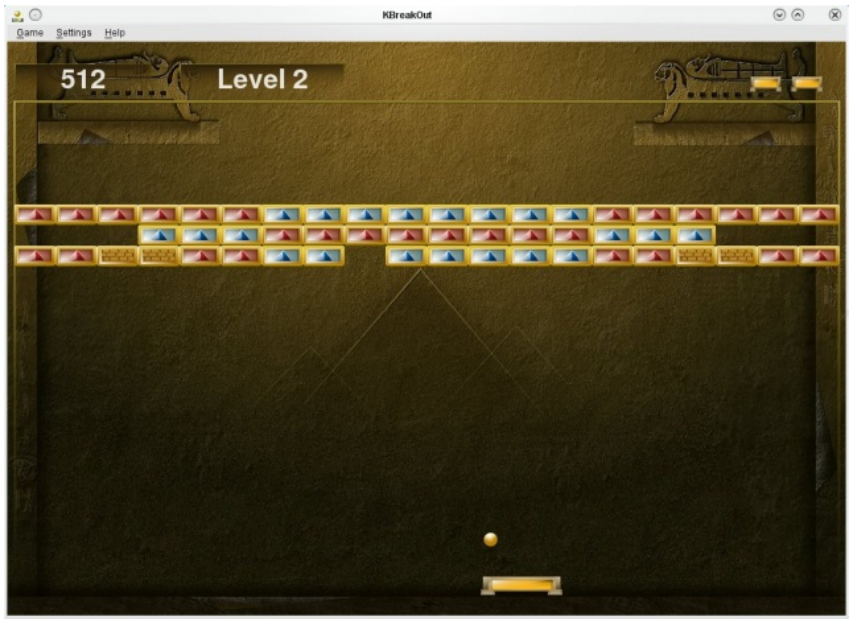
Technical details The required tools in terms of programming languages, libraries, documentation and version control tools. These are mandatory.

kbreakout Android port

Native app with Qt

Description

kbreakout is a Qt Breakout-like game. The goal of this project is to port the game to the Android platform.



kbreakout Android port

Native app with Qt

Goals and Techniques

Short-term goal Functional port of the kbreakout game.

Long-term goal Network multiplayer support.

Educational value Learn a major GUI framework (Qt), learn the Android native app development process.

Technical details C++ language, doxygen documentation, version control with google code or github; Android SDK.

Android app development

Tablesmith clone

Description

The Tablesmith shareware program [a] allows the user to generate random results from tables (e.g., random encounters). Our goal is to create a mobile application that replicates the same functionality. Each table is composed of entries containing text (the payload) and a probability. Probability is expressed as combination (sum) of uniformly-distributed random values (i.e., die rolls), using the standard die notation [b].

---

[a] http://www.mythosa.net/Main/TableSmith

[b] http://en.wikipedia.org/wiki/Dice_notation

Android app development
Tablesmith clone

Goals and Techniques

Short-term goal develop an Android App that allows the user to download tables and randomly select elements from them.

Long-term goal replicate the functionalities of Tablesmith, adapting the UI to handheld devices.

Educational value develop a complete application in the Android SDK (and you can sell it ;)).

Technical details Java language, Javadoc documentation, version control with google code or github; Android SDK.

Web App Re-engineering

PHP refactoring/debugging: Vaults of Pandius

Description

A large static web-site has been partially ported to a custom-made dynamic framework. The goal of the project is to bring the port to production state, allowing correct visualization, modification and creation of dynamic web pages.

Web App Re-engineering

PHP refactoring/debugging: Vaults of Pandius

Goals and Techniques

Short-term goal improve functionality to minimum (access, read, modify, administration), support at least standard monitor resolutions. Alternately, replace the custom framework with a standard one.

Long-term goal support rendering for mobile devices.

Educational value develop a web application serving a large website (600MB+ contents) using standard technologies.

Technical details PHP language, MySQL db, doxygen documentation, version control with google code or github.

P2012 Benchmark development

Bio-inspired algorithms

Description

STMicroelectronics Platform 2012 is a novel many-core accelerator for high end embedded systems. We want to compare its performance and programmability to those of GPUs and multi-core CPUs by developing a Particle Swarm Optimization algorithm. The P2012 SDK, including simulator and emulator, will be used for this work.

P2012 Benchmark development

Bio-inspired algorithms

Goals and Techniques

Short-term goal design and implement the algorithm using the P2012 SDK.

Long-term goal allow for different optimizations for P2012, GPUs and CPUs.

Educational value develop an application for an increasingly common architectural model (programmable parallel accelerators for high end embedded systems).

Technical details OpenCL API and programming language; doxygen documentation, version control with mercurial on POLIMI servers.

Hardware-accelerated DTLS

SPEAr C3 PolarSSL integration

Description

The STMicroelectronics SPEAr platform includes a C3 crypto accelerator. The C3 crypto accelerator is an ASIC crypto accelerator embedded in modern ARM Cortex platforms by STM and already has a working interface in the form of linux device files. The goal of the project is to implement the DTLS protocol (TLS over UDP) within the PolarSSL library. A board endowed with the C3 Accelerator and the properly configured Linux distribution will be made available via SSH.

Hardware-accelerated DTLS

SPEAr C3 PolarSSL integration

Goals and Techniques

Short-term goal Support a DTLS session with one cipher suite.

Long-term goal Support the entire DTLS protocol.

Educational value work with standard technologies for embedded encryption (PolarSSL) and develop for a state of the art high-end embedded processor.

Technical details C language; doxygen documentation; version control with mercurial on POLIMI server; PolarSSL.

Ettercap plugin

Retrieving group key from WPA2-PSK

Description

- Ettercap is a well known suite to check for the feasibility of man in the middle attacks on wired/wireless networks.
- The project aims at developing a plugin for the Ettercap tool able to retrieve the group key from a WPA2-PSK network, provided that the client running Ettercap is correctly authenticated

Ettercap plugin
Retrieving group key from WPA2-PSK

Goals and Techniques

Short-term goal Implement group key retrieval for WPA2-PSK

Long-term goal Obtain a proof of concept of the feasibility of the attack against WPA2-PSK

Educational value Understand the inner working of wireless security mechanisms - hands on work on a well established codebase

Technical details Compliance with the project coding style guidelines, git DVCS, basic knowledge of the mechanisms of WPA2 (or willingness to acquire it)

Legacy code reengineering
Reengineering Hunt

Description

- `hunt` is a small (~200kB codebase) multiplayer maze game with network support
- The codebase is significantly old and calls for a proper reengineering (removal of pre-ANSI style C, a few `goto`s and similar cruft)
- The purpose of the project is to reengineer the codebase targeting an x86 (32-64 bits) Linux system only, and adding the support for a basic password authentication on the server

Legacy code reengineering
Reengineering Hunt

Goals and Techniques

Short-term goal Perform a full reengineer of the code base

Long-term goal Obtain a portable, extensible version of the codebase, while retaining backward compatibility with the communication protocol

Educational value Understand the challenges of tackling someone else's code, understand efficient, low latency communications over the net

Technical details Legacy-code free version of the program, proper makefile and documentation, use of any DVCS (mercurial, git or bazaar) to coordinate the project efforts, working across the most diffused Linux distribution (Debian/RH based)

Embedded http Server

Arduino web server

Description

There is only a proof-of-concept http server for the Arduino platform. The goal of the project is to improve the http server to serve dynamically generated pages. The generator is a module that can linked into the http server, and is invoked when a page is requested. Static strings used in the module should be compressed with an appropriate algorithm. A sample generator module is also needed to demonstrate the functionality. The sample module will report the state of the sensors of the Arduino platform. Bring your own Arduino platform!

Embedded http Server

Arduino web server

Goals and Techniques

Short-term goal Develop the http server with an interface for a dynamic page generator module and the sample module.

Long-term goal Allow support of encrypted communication through https.

Educational value Work with a widespread embedded platform.

Technical details C language; doxygen documentation; version control with google code or github.

Code Tile Generation and Analysis

ARM assembly tile generation

Description

Code tiles are fragments of assembly code that can be used for dynamic code morphing, as they have the same semantics as a single instruction, but a different power profile. The goal of the project is to wrap the tile generation library (C code) in a Python shell to allow scripting of the tile library. A second tool is needed to analyze existing code and detect which (normalized) instructions are used, with the relative frequency (static).

Code Tile Generation and Analysis

ARM assembly tile generation

Goals and Techniques

Short-term goal Develop a wrapper for the tile generation library and a code analysis tool.

Long-term goal Support for different instruction set architectures (ISA).

Educational value Understanding of the ARM ISA; learning how to expose C libraries to the Python shell.

Technical details C language, ARM assembler language; doxygen documentation; versioni control with Mercurial.

Dynamic Code Re-Scheduling

Re-Scheduling ARM machine code

Description

To hide the actual operation of a function, it is useful to periodically (randomly) re-order the instructions, while preserving the semantics. A code scheduler should be implemented that is able to generate equivalent but re-scheduled functions. The scheduler must be integrated in an existing suite of side-channel countermeasures, and needs to be extremely fast.

Dynamic Code Re-Scheduling

Re-Scheduling ARM machine code

Goals and Techniques

Short-term goal Develop the code scheduler.

Long-term goal Support rescheduling of instructions that need recomputing of relative addresses.

Educational value Understanding of the ARM ISA; understand the nature of dependencies among machine instructions.

Technical details C and ARM assembly language; doxygen documentation; versioni control with Mercurial.