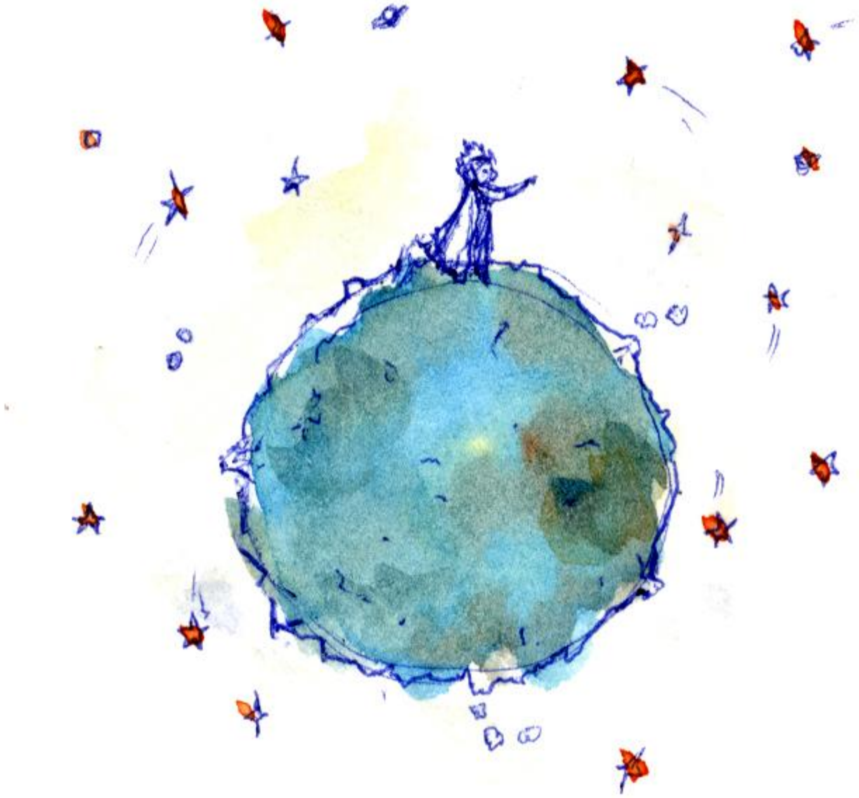
Le Petit Project

The Named Data Networking Multiplayer Online Game



A Design Document for REMAP UCLA

Authored by Phoebe Choo

Version # 1.0

Tuesday, November 13, 2012

Table of Contents

1.0. Project Overview v

1.1. Named Data Networking (NDN) v

1.2. Multiplayer Online Game (MOG) vi

1.3. Design History vi

1.4. Ongoing Discussion Points and References vii

2.0. Game Overview viii

2.1. Game Concept viii

2.2. Target Audience viii

2.3. Game Flow Summary viii

2.4. Look and Feel viii

3.0. GAMEPLAY AND MECHANICS ix

3.1. Gameplay ix

3.1.2. Game Progression ix

3.1.3. Challenge/Puzzle Structure ix

3.1.4. Objectives ix

3.2. Mechanics ix

3.2.2. Movement ix

3.2.3. Objects ix

3.2.4. Actions x

3.2.5. Combat x

3.2.6. Economy x

3.3. Screen Flow x

3.4. Replaying and Saving x

3.5. Cheats x

4.0. Story, Setting and Character xi

4.1. Narratives xi

4.2. Game World xi

4.2.1. Home World xi

4.2.2. Asteroid #1 xi

4.3. Characters xii

4.3.1. Character #1 xii

5.0. Interface xiii
5.1. Audio xiii
5.2. Music xiii
5.3. Help System xiii
6.0. Technical xiv
6.1. Target Hardware xiv
6.2. Development hardware and software xiv
6.2.1. Plug-ins xiv
6.3. Network xiv
6.3.1. Data Sources xiv
6.3.2. Synchronization xiv
6.4. Secondary Software xiv
7.0. Appendices xv

0.1. Prologue- About this Design Document

A Game Design Document (GDD) is the blueprint from which a game is to be built. As such, every single detail necessary to build the game should be addressed in the document (or support documents).

The GDD is also a reference document. Members of the development team should be able to use the document to find specific information for their specific needs. As such a manuscript may grow sizably in order to document every piece of the game, some information might not appear in the GDD itself but instead would appear in supplemental documents such as an Art Bible or Test Plan.

Last but not least, a GDD is meant to be a living document. Just as an artist changes the design of his painting every time he takes his brush to the canvas, a game evolves as code and art are created. In this project, the game should evolve naturally to improve its purposes and to adapt to its users as well. The GDD is then the communication tool from which all the members of the team can follow that evolution.

1.0. Project Overview

This is a project funded by the National Science Foundation (NSF) and developed by the Research Center for Media, Engineering and Performance (REMAP) in collaboration with the University of California, Los Angeles (UCLA).

Key Objectives [1]

- ✓ Explore the synchronization mechanism of more complex game events such as inter-user and user-object interactions.
- ✓ Evaluate the scalability of NDN-based peer-to-peer games.
- ✓ Study authentication and anti-cheating problems in more detail and provide some preliminary solutions based on NDN's signature-based trust model.

1.1. Named Data Networking (NDN)

"While the Internet has far exceeded expectations, it has also stretched initial assumptions, often creating tussles that challenge its underlying communication model. Users and applications operate in terms of content, making it increasingly limiting and difficult to conform to IP's requirement to communicate by discovering and specifying location. To carry the Internet into the future, a conceptually simple yet transformational architectural shift is required, from today's focus on where -- addresses and hosts -- to what -- the content that users and applications care about.

The **Named Data Networking (NDN)** project aims to develop a new Internet architecture that can capitalize on strengths -- and address weaknesses -- of the Internet's current host-based, point-to-point communication architecture in order to naturally accommodate emerging patterns of communication. By naming data instead of their locations, NDN transforms data into a first-class entity. The current Internet secures the data container. NDN secures the contents, a design choice that decouples trust in data from trust in hosts, enabling several radically scalable communication mechanisms such as automatic caching to optimize bandwidth. The project studies the technical challenges that must be addressed to validate NDN as a future Internet architecture: routing scalability, fast forwarding, trust models, network security, content protection and privacy, and fundamental communication theory. The project uses end-to-end testbed deployments, simulation, and theoretical analysis to evaluate the proposed architecture, and is developing specifications and prototype implementations of NDN protocols and applications. NDN Technical Report NDN-0001 "Named Data Networking (NDN) Project" is a slightly modified version of the NDN project proposal.

The NDN project was funded by NSF in September 2010 as one of the four projects under NSF's Future Internet Architecture Program."

Excerpt from http://named-data.net/

[1] Adapted from LPP Poster.pdf by Zhening Qu and Jeff Burkes, (insert date??)

1.2. Multiplayer Online Game (MOG)

A multiplayer/ multi-user online game (MOG) is one in which more than one person can play in the same game environment at the same time, and able to interact with other individuals over some form of computer network.

This project was envisioned in line with developing applications using the new network approach of NDN, as an experiment in scaled synchronized scene graphs, assets and states.

> ***Huge amounts of data packets need to be exchanged at high frequency in MOGs. This makes a MOG an appropriate test for NDN distribution efficiency. Synchronization mechanisms are needed for MOGs to maintain game state consistency, and they exhibit fundamental influences on the game performances. Emerging NDN primitives can thus be exploited and evaluated in the game context. MOGs are also good test cases for the NDN security model as player authentication and cheating prevention are of special importance.*** [1]

In addition, the development of a peer-to-peer (P2P) system using non-centralized architecture that engages the public is of interest.

1.3. Design History

27th September 2012

- Qu Zening proposes Le Petit Prince as the background story. "MMO: player actions is decided by on-line data, objects in the game are dynamically created by pulling data from the web."
- E-mail list created: ndn-mog@lists.cs.ucla.edu

October 2012

- *Egal Car: A Peer-to-Peer Car Racing Game* was developed using Egal, C# bindings for Xerox PARC's CCNx library, including its new Sync protocol, in Unity.
- Jeff Burke suggests re-interpreting Le Petit Prince for a non-commercial MOG experimental release. Unity 3D is decided as the convenient development tool of choice for game architecture.
- Repository created: https://github.com/remap/ndn-lpp
- Redmine created: http://sea.remap.ucla.edu:8080/login

November 2012

- V1.0 of GDD is drafted by Phoebe Choo

1.4. Ongoing Discussion Points and References

[1]***Goals:***

- ✓ Maintain game consistency among players
- ✓ Exploit emerging Sync primitives
- ✓ Inform NDN-based game development and synchronization protocol choices

1.4.1. Realtime data sources mapped into game objects

- ○ http://openflights.org/data.html
- ○ http://openflights.svn.sourceforge.net/viewvc/openflights/openflights/data/airports.dat
- ○ Flight Patterns of America by Aaron Koblin: http://www.aaronkoblin.com/
- ○ Jonathan Harris: http://www.number27.org/
- ○ Mark Hansen and Ben Rubin: http://earstudio.com/
- ○ http://www.nasa.gov/open/plan/data-gov.html
- ○ http://www.thesheepmarket.com/
- ○ Flickr images

1.4.2. Mapping real-World Coordinates into Game World

- ○ OpenStreetMap tiling scheme (based on the Mercator projection) http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

1.4.3. Development

- ○ Embedding Mono: http://www.mono-project.com/Embedding_Mono
- ○ P/Invoke, invoking C from C# and vice versa

1.4.4. MMO Networking toolkits

- ○ http://unity3d.com/unity/engine/networking
- ○ Separate from socket comm mechanisms?

1.4.5. Storyline

- ○ Le Petit Prince: http://en.wikipedia.org/wiki/Le_petit_prince

2.0. Game Overview

Game prototype much be rich in (1) user-user interaction (2) user-object interaction. Therefore it is important that we put both kinds of interactions into the gameplay…

1. User-user Interactions: users send requests to each other; if a request is granted, the requester's experience will grow; such requests could be a drawing of a sheep, but we don't really have to implement them in the front end for now (as long as the request packets are sent out, the gameplay will be able to move on automatically).
2. User-object Interactions: each asteroid maintains a certain number of objects; users interact with these objects to gain experience; such objects could be seedlings that come out of the earth of the asteroid and users should try to get rid of them.

Excerpt from LPP: Gameplay & Namespace (draft) by Zening Qu, 21st October 2012

2.1. Game Concept

What is the moral of the story? A journey of self-discovery/ to find true love/ the way home?

2.2. Target Audience

Anybody with access to the internet via a computer, console? Platforms? Age range?

2.3. Game Flow Summary

How does the player move through the game? Flowchart, etc.

Draft of interface, and summary of player movement in game itself.

2.4. Look and Feel

Basic look and feel of the game?

Saint-Exupéry painted the original illustrations, in a simple but elegant watercolor style.

3.0. GAMEPLAY AND MECHANICS

Considerations:

a) Using NDN
b) Peer-to-peer architecture rather than a centralized one
c) Real-time data sources mapped into game objects

Players begin on a home world, with an avatar they choose to represent themselves.
They are then expelled from this home world to which they endeavor to return, by gathering the necessary resources to do so.
To gather these resources they visit various virtual worlds, and interact with other NPCs and/or players.
Once they have sufficient resources, they return to their home world…

3.1. Gameplay

How does the game work?
How does player interact with already-existing data-driven art?

3.1.2. Game Progression

Do the players level up? After they return to the home world what happens?

3.1.3. Challenge/Puzzle Structure

Unfolding story once milestone achieved? Special graphics/animations unlocked when challenge solved?

3.1.4. Objectives

What are the objectives of the game for the player? Gather as many resources as possible? Return to home world?

3.2. Mechanics

What are the implicit and explicit rules?

3.2.2. Movement

How do players move through the game- arrow buttons, flying, no movement except for changing scenes?

3.2.3. Objects

What objects are there in the world, how can they be interacted with?

3.2.4. Actions

What players can do, i.e. Picking up, carrying, re-combining, dropping, talking, reading…

3.2.5. Combat

Is there any conflict or competition?

3.2.6. Economy

Is there an economy to the game, what are the assets worth?

3.3. Screen Flow

Ideally a graphical description of how each screen relates to each other, and then also a description of how each screen works, e.g. main screen, options screen, interaction screen, tutorial

3.4. Replaying and Saving

How do players return to the game, will replay be an option…

3.5. Cheats

Non-standard methods used to create advantages beyond normal gameplay. Originally for play-yesting purposes, but now evolved to be commonly accepted as walkthroughs/guidebooks, special bonuses to be unlocked by players, or simply to make the game more accessible to the casual gamer.

4.0. Story, Setting and Character

What is the player up against? Who is holding the resources we are trying to discover? Where is the secret hidden?

In LPP: humans, animals, and planets.

Other suggestions: Actual planets (travelling a star sign?), micro-organisms (see the video game Flow on the PS), aliens/monsters (think Ugly Dolls, Pokemon), mythical creatures or other animals (players could be centaurs meeting dinosaurs), superheroes (players gain powers with experience?).
E.g. the player could be an angel that lost his wings and fell out of heaven to land on prehistoric Earth and eventually meet Miss. Libra by discovering nearby star constellations who then provides him a way back to heaven by making him new wings after he finds all the materials he needs from other players etc.

4.1. Narratives

What the player sees, specific details like scripts or cut scenes. If this section gets too long it may be put into the Story Bible.

4.2. Game World

General look and feel of the world.

4.2.1. Home World

4.2.1.1. General Description

4.2.1.2. Physical Characteristics

4.2.1.3. Connections to other areas

4.2.2. Asteroid #1

4.2.2.1. General Description

4.2.2.2. Physical Characteristics

4.2.2.3. Connections to other areas

4.3. Characters

Details of NPCs in the game, or types of player avatars. Any active opponents that requires strategic decision making? Enemies- Villians/monsters?

4.3.1. Character #1

4.3.1.1. Back Story

Why is he there?

4.3.1.2. Characteristics

How does the character look? Animations?

4.3.1.3. Abilities

What can the character do? What’s his relationship to other characters, and how does he interact? Statistics? Artificial Intelligence? Player/collision detection, pathfinding- if necessary, there can be a separate section detailing AI.

5.0. Interface

Visuals- HUD? Control system- how does the player control the game, what are the specific commands?

5.1. Audio

Sounds effects, voiceovers, etc.

5.2. Music

Any soundtracks used for background music…

5.3. Help System

Tutorials, hints/ tips players can access.

6.0. Technical

This may require reference to other documents or a Technical Bible.

6.1. Target Hardware

Platforms? Mac only?

6.2. Development hardware and software

Game Engine- Unity 3D? Scripting language..

6.2.1. Plug-ins

6.2.1.1. ccnx

Egal, C# bindings for Xerox PARC's CCNx library?

6.3. Network

Network module?

http://unity3d.com/unity/engine/networking

6.3.1. Data Sources

Asteroid data from airports – how to instantiate an unchanging but spatially organized (distributed) repo of the asteroid locations.

6.3.2. Synchronization

Namespace?

- Refactor sync from Egal car for use in LPP
- Publish data into NDN namespace for use in NDN-mog asset sync

6.4. Secondary Software

Editor, installer, updating of software…

7.0. Appendices