Projet Spicy Invader

UX, OO, DB

Ethan Schafstall

28.08.2023-03.11.2023

Table des matières

[1. Introduction 1](#_Toc135905380)

[2. Protocole MQTT 2](#_Toc135905381)

[3. Infrastructure virtuelle 3](#_Toc135905382)

[3.1 Paquets 3](#_Toc135905383)

[3.2 IPs et masques 3](#_Toc135905384)

[4. Implémentation 8](#_Toc135905385)

[4.1 Activation de service mosquitto 8](#_Toc135905386)

[4.2 Commandes de Communication 8](#_Toc135905387)

[4.3 Tests du Communication 9](#_Toc135905388)

[4.3.1 Test RaspBroker allumé 9](#_Toc135905389)

[4.3.2 Test RaspBroker éteinte 10](#_Toc135905390)

[5. Conclusion 12](#_Toc135905391)

# 1. Introduction

L’objectif de ce projet est de mettre en place le protocole de messagerie MQTT (Message Queuing Telemetry Transport) dans an environnement virtuel avec plusieurs VMs tournant L’OS Rasbian qui communique entre eux via le réseau virtuel.

Durent le projet nous verrons les notions de virtualisation, la configuration d’un réseau, l’OS rasbian, le protocool MQTT, et le service de mosquitto.

Ce projet simule une situation réelle ou deux machines, géographiquement séparateur ou les machines pouvez communiquer, avec le Protocol MQTT, via une machine qui les relis.

Cette situation va être créer avec la virtualisation, l’OS Raspian, et le service mosquitto.

# 2. Expérience utilisateur (UX)

# 3. Programmation orientée objet (OO)

# 4. Base de données (DB)

## 4.1. Importation données et schéma de base de données

## 4.2. Gestions des utilisateurs

## 4.3. Requêtes de sélection

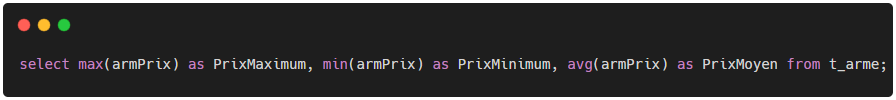
Les requêtes de select SQL servent à récupérer des donnés depuis la base de données du jeu, pour ensuite afficher/ou en servir pour l’UX, ainsi que le fonctionnement du jeu.

### 4.3.1

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

### 4.3.2



### 4.3.3

### 4.3.4

### 4.3.5

### 4.3.6

### 4.3.7

### 4.3.8

### 4.3.9

### 4.3.10