TÉTRAS
Formation en alternance

IUT ANNECY
UNIVERSITÉ SAVOIE MONT BLANC

Internet of Things (IoT)

Projet : Agents Battle Arenas

Mickaël Bettinelli
(mickael.bettinelli@univ-smb.fr)

Vue d’ensemble

Objectif : développer un jeu distribué embarqué

Sommaire :

1. Le jeu
2. Les règles
3. L’architecture du jeu
4. Evaluation du projet
5. Planning prévisionnel

Le jeu - vue d’ensemble

- *Battle royal* de personnages dans une arène
- Les personnages sont des agents développés par les étudiants
- Chaque groupe d’étudiants peut créer 3 personnages pour se battre

Images non contractuelles

Le jeu - vue d’ensemble



- Plusieurs arènes seront disponibles simultanément
- Les personnages peuvent :
  - Se battre
  - Naviguer entre les arènes
  - Voir qui se trouve dans leur arène
  - Se coordonner pour jouer en équipe (ou en former !)

Images non contractuelles

Le jeu - les personnages

Les attributs des personnages :

- Vie
- Force
- Armure
- Vitesse

Images non contractuelles

Le jeu - les personnages

Les personnages ont 4 actions possible :

- En **frapper** un autre
  - Inflige le nombre de points de force du personnage à la cible
- **Bloquer** un coup
  - Réduit les dégâts du nombre de points d'armure du personnage
- **Esquiver** un coup
  - Donne une chance de ne subir aucun dégât
- **Changer d'arène**
  - Le personnage devient hors de portée de ceux de son ancienne arène

Images non contractuelles

Le jeu - les combats

Un combat se déroule de manière synchrone :

- Des combattants se trouvent dans une arène
- Les personnages choisissent une des 4 actions possible
  - Si l'action est l'attaque ou le changement d'arène, ils doivent sélectionner une cible
- Les actions de tous les personnages sont exécutées de manière séquentielle au même moment

Images non contractuelles

Le jeu - les combats

Quelques règles sur les combats :

- Un personnage qui change d'arène peut prendre des dégâts avant de partir
- Si deux personnages ciblent un même combattant fragile, un seul des deux sera le tueur (le plus ancien arrivé dans cette arène).
- Un personnage dans une arène voit tous les autres combattant de cette même arène ainsi que leurs attributs

Images non contractuelles

Les règles - les personnages

- Chaque groupe peut créer 3 personnages
- Chaque personnage peut avoir au maximum 20 points d'attributs sur l'ensemble de ses caractéristiques
- La vie, la force et l'armure sont des valeurs de 0 à 20.
- La vitesse est comprise entre 0 et 10. Elle représente le % de chance de réussir une esquive.

Exemple.

Vie : 10
Force : 5
Armure : 3
Vitesse : 2

- Un personnage arrivé à 0 point de vie (pv) est considéré comme mort et ne peut plus jouer

Images non contractuelles

Les règles - les pièces d’or

- Tuer un autre personnage rapporte 10 d’or au tueur

Images non contractuelles

Le gagnant

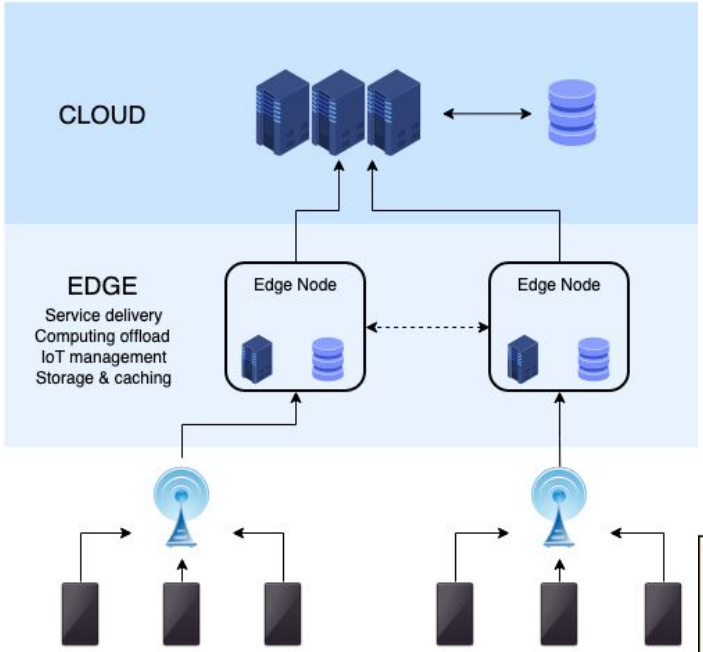
Le groupe d’étudiants gagnant est celui qui a le plus d’or à l’issue de 100 matchs

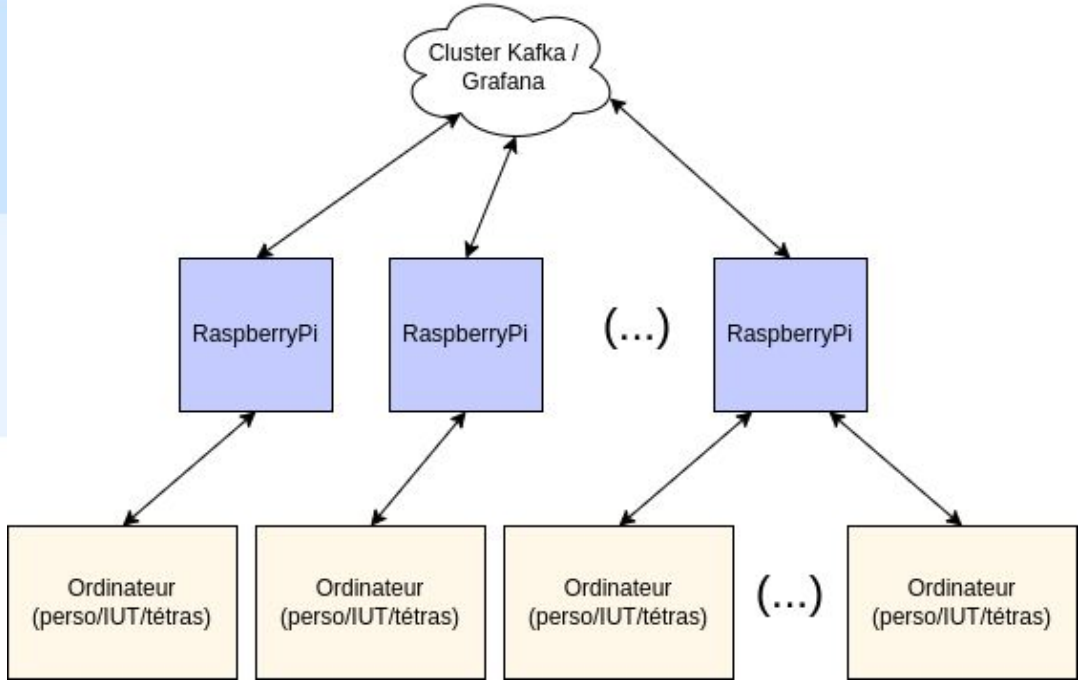
Visualisation des matchs

Objectifs : visualiser les statistiques en temps réel de la partie, par exemple :

- Les équipes qui gagnent le plus d’or
- Les personnages qui font le plus de morts, de dégâts, qui en reçoivent le plus, *etc.*
- Les arènes où se trouvent les personnages

Infrastructure du projet

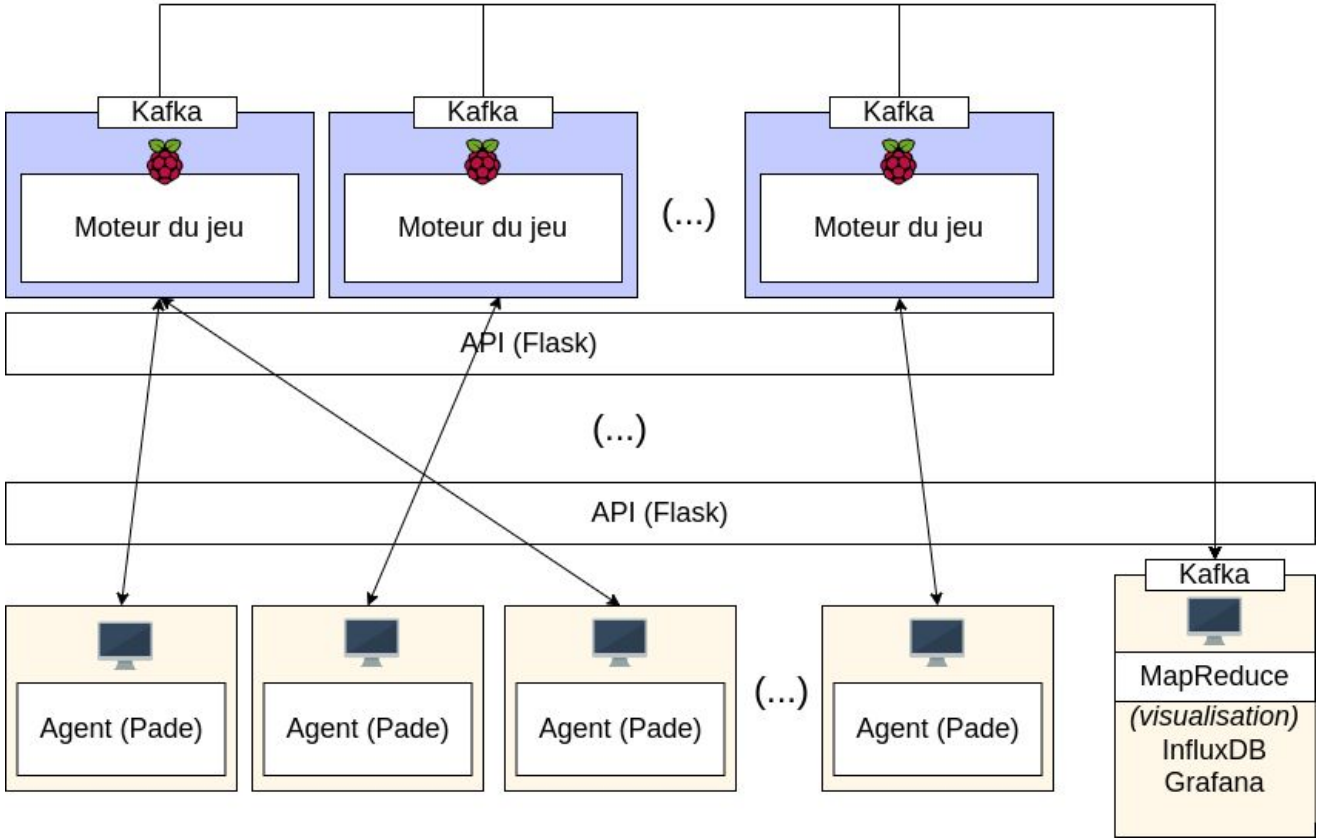




En faisant une analogie avec le continuum cloud-edge-IoT :

- Les ordinateurs sont des IoT
- Les raspberries sont des machines edge

Architecture logicielle



Les frameworks que nous allons utiliser :

- Flask
- Kafka
- InfluxDB
- Grafana

Evaluation

2 notes :

- Un **rapport** par groupe (max 5 pages) dans lequel indiquer :
  - Les choix de conception de l'API
  - Les choix des métriques visualisées avec Grafana
  - L'usage de Kafka

  Tous vos choix doivent être justifiés.

- Une **soutenance** de 10mn/groupe (5mn présentation + 5mn questions):
  - Les choix de modélisation de vos agents (type d'agent, modèles, etc.)
  - La stratégie choisie

Evaluation (barème temporaire)

Le rapport (/12) :

- Conception de l'API (choix des routes, des méthodes) (/3)
- Usage de Kafka pour la transmission des données (/3)
- Métriques visualisées (quantité et pertinence) (/2)
- Intégration des frameworks de visualisation (InfluxDB + Grafana) (/4)

La soutenance (/8) :

- Modélisation de l'agent (justification des choix techniques) (/6)
- Implication dans la réalisation d'une stratégie efficace (/2)

Bonus de performance en arènes

Un classement des meilleures équipes sera réalisé à la fin des matchs

1er !		+2 points
2nd		+1 points
3ème		+0.5 points

Planning des séances

Tâches/séances	1	2	3	4	5
Présentation du travail Conception + implémentation **API Flask** *(travail 1, Groupe, Tous)* Vérification de la connectivité des RPi	4h				
Intégration du moteur à l'API *(travail de groupe)*		1h			
Conception d'agents *(travail de groupe)* Test du moteur		3h			
Création du client de **visualisation** *(travail 1, Groupe, Tous)* (InfluxDB, Grafana)			4h		
Mise en place de **Kafka** *(travail de groupe)*				3h	1h
Test du jeu (2h) + soutenances (1h) **lancement des matchs !** (1h)				1h	3h

Quelques détails techniques

Conception de l’API

Fonctionnalités à implémenter :

- Récupérer les informations sur un personnage
- Récupérer les informations sur une arène
- Faire entrer un personnage dans une arène
- Choisir une cible à un personnage
- Choisir l’action d’un personnage
- Lancer le jeu

Conception des agents

Agent réactif

Percevoir l'environnement

Agir sur l'environnement à l'aide de règles

Agent cognitif

Percevoir l'environnement

Agir sur l'environnement

Mémoriser des informations

Générer des plans d'actions

Inférer des connaissances

Agent cognitif et apprenant

Percevoir l'environnement

Agir sur l'environnement

Mémoriser des informations

Inférer des connaissances

Générer des plans d'actions

Apprendre de ses expériences précédentes

Utilisez le langage que vous préférez !

Installation PADE

pip install pade --use-deprecated=backtrack-on-build-failures