

# MQTT JSON

---

## ROUTE

Transmetteur -> Receveur

Description

```
{  
  
}
```

## map

### OpenCV -> GameEngine, Joueur

Informations de la map (longueur, largeur), informations obstacles (angle orientation, longueur, largeur):

A RENVOYER A CHAQUE X SECONDES ?

```
{
  "mapWidth": int,
  "mapHeight": int,
  "checkpoints": [
    {
      "id": int,
      "x": int,
      "y": int,
    },
    ...
  ],
  "obstacles": [
    {
      "id": int
      "angle": float,
      "x": int
      "y": int
    },
    ...
  ]
}
```

## player/register

### Joueur -> GameEngine

Informations de base du joueur qui se connecte à la partie (salon, attente des joueurs)

```
{
  "uuid": str, // utiliser QUuid de Qt
  "pseudo":str,
  "controller": str, // ia, keyboard, controller, vr, phone // pour le
graphique
  "vehicle": str,
  "team": int // null automatiquement si partie libre
}
```

## player/control

### Joueur -> GameEngine

```
{
  "uuid": str,
  "angle": float, // en rad
  "power": int, // [-100%;100%]
  "buttons": { // état des boutons
    "banana": bool,
    "bomb": bool,
    "rocket": bool
  }
}
```

# game

## GameEngine -> Joueur

Informations de la partie en temps réel (coordonnées des joueurs, leur véhicule, leur vitesse... )

Informations des items placés sur la map en temps réel (coordonnées).

```
{
  "players": [
    {
      "uuid": str,
      "pseudo": str,
      "color": str,
      "team": int
      "x": int,
      "y": int,
      "angle": float, // en rad
      "speed": int,
      "vehicle": str, // type de vehicle : bike, car, truck
      "items": { // nombre d'objet que le joueur possède
        "banana": int,
        "bomb": int,
        "rocket": int
      },
      "lastCheckpoint": int, // dernier point de passage passé
      "currentLap": int,
      "status": str, // driving, accident
      "controller": str // ia, keyboard, controller, vr, phone // pour
                        // de l'affichage seulement
    },
    ...
  ],
  "items": [
    {
      "x": int,
      "y": int,
      "angle": float, // en rad
      "status": str // banana : placed; bomb : flying, waiting,
                    // exploding; rocket : flying, exploding
      "type": str // banana, bomb, rocket
    },
    ...
  ],
  "elapsedTime": int,
  "infoMessage": str, // ex : "la partie commence dans 10 secondes"
  "status": str // waiting, progress, ended, paused
}
```

# game/properties

## GameEngine -> Joueur

Initialisation de la partie (propriété de la partie)

Taille de la carte

Nombre d'items

A RENVOYER A CHAQUE X SECONDES ?

```
{
  "lapsNb": int, // nombre de tour à faire
  "teamNb": int, // nombre d'équipe dans la partie

  "circleRadius": int,
  "rectangleWidth": int,
  "rectangleHeight": int,
  "checkpointRadius": int,

  "bananaNb": int, // -1 = infinie
  "bananaCd": int, // temps entre deux utilisations
  "bananaTtl": int, // temps avant que la banane disparaisse
  "bananaRadius": int, // taille de la banane
  "bombNb": int, // -1 = infinie
  "bombCd": int, // temps entre deux utilisations
  "bombTtl": int, // temps avant explosion
  "bombRadius": int, // taille de la bombe
  "bombExplosionRadius": int // taille de l'explosion
  "rocketNb": int, // -1 = infinie
  "rocketCd": int, // temps entre deux utilisations
  "rocketSpeed": float, // vitesse d'une roquette
  "rocketRadius": int // taille d'une roquette

  "vehicleOptions" {
    "<vehicle>": { // vehicle : bike, car, truck
      "maxSpeed": int,
      "acceleration": float,
      "weight": int,
      "steeringAngle": float, // angle de braquage, en rad
      "width": int, // largeur du véhicule
      "height": int // longueur du véhicule
    }
  }
}
```

UUID : V4 obligatoirement généré par une fonction sur le client - RFC4122

## Mise à jour 29/09 :

### Ajout de la taille des véhicules

/game/properties vehicleOptions : + width

/game/properties vehicleOptions : + height

### Ajout de la taille des objets

/game/properties : + bananaRadius

/game/properties : + bombRadius

/game/properties : + bombExplosionRadius

/game/properties : + rocketRadius

### Modification de la taille des obstacles / checkpoints

/game/properties circleRadius : float -> int

/game/properties rectangleWidth : float -> int

/game/properties rectangleHeight : float -> int

/game/properties checkpointRadius : float -> int

### Précision

Angle : en radian

Color : mot (ex: "red", "blue", ...) ou hexadécimal (ex: "#387639")

### Modification du nom des topics

le / au début des topics disparaît