Documentación Mensajes del server.

# MENSAJES MANDADOS POR EL SERVER:

1. CONECCTIONSTATUS:
   1. Se manda cuando el server ha recibido la petición de conexión del servidor.
   2. Campos:
      1. conectionStatus -> true
2. DRAWMAP:
   1. Se manda en la creación de la sala para pintar el mapa
   2. Campos.
      1. event -> “DRAWMAP”
      2. posX -> Array con las posiciones X de todos los objetos
      3. posY -> Array con las posiciones Y de todos los objetos
      4. height -> Array con todas las alturas de los objetos
      5. width -> Array con todas las anchuras de los objetos
      6. myType -> Array con el tipo de objeto pasado a String entre:
         1. GROUND,WALL,OBSTACLE,POWERUP,OBSTACLEPOINT,SLOPE,GENERICPOWERUP,DOOR,TRAPDOOR,TRAMPOLINE,FINISH
3. FINISH:
   1. Se manda cuando se ha acabado la carrera con el resultado
   2. Campos:
      1. event -> “FINISH”
      2. winner -> boolean si has ganado o no
      3. time -> tiempo que has hecho
      4. maxTime -> Tiempo máximo para ganar el circuito en caso singlePlayer
4. GROUNDCOLLISION:
   1. Se manda cuando se entra en contacto por primera vez con un suelo para poner la animación.
   2. Campos:
      1. event -> “GROUNDCOLLISION”
5. LOSESHIELD:
   1. Se manda cuando teníamos escudo y lo perdemos
   2. Campos:
      1. event -> “LOSESHIELD”
6. OBJECTUSED:
   1. Se manda cuando se ha usado un objeto para quitar el indicador de que todavía se tiene dicho objeto y pintar su activación en la pantalla
   2. Campos:
      1. event -> “OBJECTUSED”
      2. type -> String con los siguientes tipos:
         1. SHIELD, STAMINA, WEIGHT, LETUCCE, SPEED, INK, NULL
7. OBSTACLEUPDATE:
   1. Se manda cuando ha habido una actualización en el estado de los fuegos.
   2. Campos:
      1. event -> “OBSTACLEUPDATE”
      2. id -> int con el índice del obstáculo
      3. estate -> String con los siguientes estados:
         1. ACTIVE,NOTACTIVE,PREACTIVATE
8. OBSTACLECOLLISION:
   1. Se manda cuando el caracol choca con un obstáculo para la animación de cansado
   2. Campos:
      1. event -> “OBSTACLECOLLISION”
9. SLOPECOLLISION:
   1. Se manda cuando se entra en contacto por primera vez con una cuesta para poner la animación.
   2. Campos:
      1. event -> “SLOPECOLLISION”
      2. degrees -> int con los créate de la cuesta
10. SNAILUPDATE:
    1. Se manda información sobre si hay que cambiar las animaciones si te quedas sin stamina
    2. Campos:
       1. event-> “SNAILUPDATE”
       2. runOutStamia -> true o false si te quedas sin stamina
       3. recoverStamina -> true o false si recuperas la stamina
11. TAKEPOWERUP:
    1. Se manda cuando el caracol colisiona con un power up. Te manda el power up que se ha creado, se debe borrar la caja sorpresa del escenario.
    2. Campos:
       1. event -> “TAKEPOWERUP”
       2. type -> una String con el tipo de power up creado entre:
          1. {SHIELD,STAMINA,WEIGTH,LETUCCE,SPEED,INK,NULL}
12. TICK:
    1. Se manda cada refresco de la sala con las posX e Y de nuestro caracol y su estamina
    2. Campos:
       1. event -> “TICK”
       2. posX -> float con tu posX
       3. posY -> float con tu posY
       4. stamina -> float con tu stamina
13. UPDATEDOOR:
    1. Se manda un evento para cambiar la animación del phaser cuando la puerta se abre o cierra.
    2. Campos:
       1. event -> “UPDATEDOOR”
       2. id -> un int con el id
14. UPDATETRAMPOLINE:
    1. Se manda un evento para cambiar la animación del phaser cuando el trampoline se activa.
    2. Campos:
       1. event -> “UPDATETRAMPOLINE”
       2. id -> un int con el id
       3. estate -> String con los siguientes estados:
          1. ACTIVE,NOTACTIVE,PREACTIVATE
15. UPDATETRAPDOOR:
    1. Se manda un evento para cambiar la animación del phaser cuando la trampilla se abre o cierra.
    2. Campos:
       1. event -> “UPDATETRAPDOOR”
       2. id -> un int con el id
16. WALLCOLLISION:
    1. Se manda cuando se entra en contacto por primera vez con una pared para poner la animación.
    2. Campos:
       1. event -> “WALLCOLLISION”
17. WAITINGROOMSTART:
    1. Se manda mientras se espera a que la sala se llene.
    2. Campos:
       1. event -> “WAITINGROOMSTART”
       2. roomName -> String con el nombre

# MENSAJES MANDADOS POR EL CLIENTE:

1. SINGLEPLAYER:
   1. Se recibe el nombre del jugador y la sala, se envia al darle a que quieres jugar solo.
   2. Campos:
      1. playerName
      2. roomName
2. UPDATEINPUT:
   1. Se recibe cada vez que el jugador interactua con el cliente
   2. Campos:
      1. isSprinting
      2. useObject

# Mensajes sin implementar cliente:

Mensaje con el mapa y la ambientación elegida.