note : l'utilisateur pourra créer de nouveaux types qui seront soit hérités de Object, soit d'une classe héritée de Object en Javascript, les types ne sont pas définis mais je les ai mis pour plus de clarté

La grammaire haut niveau permet l'héritage multiple. Il reste à savoir si ce sera implémentable. A priori oui puisque la notion d'héritage n'apparait pas dans la grammaire bas niveau (les attributs seront des ressources et des paramètres d'entités, il suffira de les créer les uns à la suite des autres en veillant bien à ne pas mettre des doublons).
ex :
maison is construction and breakable

type = classe

Object

mass : double
position : coordonnee(double, double, double)
orientation : coordonnee(double, double, double)
taille : coordonnee(double, double, double)
isFix : boolean
isTraversable : boolean
//generated : boolean (destroyed)
touch (object) : boolean

Autres mots-clé représentant des classes :

Game

gameStarted : boolean
gameEnded : boolean
defeat : boolean
victory : boolean
score : int
gravity : double ou (double,double,double)

en + : il n'y en a qu'un

Counter

value : double
unite : String

en + :

Camera

position (double, double, double)
orientation (double, double, double)
isFix : boolean
fov : double
type : String
active : boolean

en + : type peut être "firstPerson", "thirdPerson" ou "free"
active sert lorsqu'il y a plusieurs caméra pour savoir laquelle capture

Time

value : double
unit : String

en + : unit est soit 'min', 'sec' ou 'ms'

- list (un groupe d'élément ordonné, sert à l'utilisateur pour ne pas avoir à rappeler chaque objets, ex : armee is list of 10 character)
- player | ally | enemy | neutral (correspond à ne rien préciser)
- multiple (à la différence d'une liste où on ne pourra pas enlever ou ajouter d'éléments hors déclaration, signifie qu'on peut générer plusieurs fois cet élément pendant le jeu, par contre on ne peut pas accéder aux éléments en particulier, sauf dans les règles du jeu, ex : jangoFett is character enemy multiple), les types sont multiple par définition, ainsi que les enemy, ally et neutral

pendant le jeu : generate 10 jangoFett in zoneGenerationClones
generate 5 obstacle on circuit1
si on ne définit pas multiple, il ne sera pas possible de générer plusieurs objets.

generate in zone / on objet / at coordonnees, pour générer avec une position aléatoire avec une restriction
- in : à l'intérieur d'une zone en veillant à ce qu'il n'y ait aucune collision lors de la génération
- on : juste au dessus de l'objet en veillant à ce qu'il n'y ait aucune collision lors de la génération
- at : génère le premier objet aux coordonnées indiquées et les suivants autour de lui en veillant à ce qu'il n'y ait aucune collision lors de la génération

Character

name : String
description : String
life : int
lifeMax : int
lifeMin : int
nbOfLives : int
magic : int
magicMax : int
magicMin : int
level : int
attack : int
defense : int
jumpForce : int
maxJumpsInTheAir : int
money : int
class : String
race : String
acceleration : int
inventory : List<Object>
//dead : boolean (alive)
moveWithCamera : Camera

en + :pour maxJumpsInTheAir, ex : pacman=0, mario=1, kirby=5, persoCheaté=1000
Si le perso a une accélération, ses déplacements se feront par rapport à sa vitesse, sinon ils se feront en direct, à l'appui et au relâchement de la touche

attribut name, l'attribut servira lors d'un éventuel affichage
exemple :
gandalf is character
gandalf has name at "Gandalf le Blanc"

Vehicle

name : String
description : String
acceleration : int
maxSpeed : int
minSpeed : int
boost : int
maxBoost : int
boostInterval : int

SpaceCraft / Plane

name : String
description : String
acceleration : int
maxSpeed : int
minSpeed : int
boost : int
maxBoost : int
boostInterval : int

en + : déplacements grâce à l'accélération
Ce qui diffère d'un personnage, c'est le contrôle en fonction de la caméra : appuidroite : le perso va à droite de l'écran (grâce à moveWithCamera), le véhicule tourne à sa droite (cela peut être vers gauche de l'écran si la caméra est braquée sur le pare-brise), l'avion fait une rotation dans le sens horaire

Obstacle

name : String

en + : isFix = true; isTraversable = false;

Weapon

name : String
nbMunitions : int
nbMunitionsMax : int
shootPower : int
shootInterval : Time(int, String)
reloadTime : Time(int, String)

en + : L'unité du Temps peut être en ms, sec ou min.

Projectile

speed : int
damages : int
damageZone : Zone

en + : dès que l'objet est généré, il se déplace jusqu'à sa destruction, damageZone est utilisé pour un missile par exemple

Sword

name : String
damages : int

en + :

Zone

name : String

en + : créera une forme englobante englobante (invisible), isTraversable = true;

Ground

type : String

en + : type pourrait être normal, neige, goudron (pour les coeff de frottement) et eau pour creer un bloc traversable mais c'est juste une idée

Bonus

name : String
description : String
value : int
unit : String
collectors : List<Objet>
typesCollectors : List<Type>
generators : List<Objet>
typesGenerators : List<Type>

en + : collectors correspond aux objets qui pourront récupérer l'objet, typeCollectors pour les types, ex: un plane en particulier peut récup le bonus et tous les vehicles peuvent récup le bonus.
generators correspond aux objets qui génèrent l'objet lorsqu'ils sont détruits, typeGenerators pour les types,

Checkpoint

name : String

en + : c'est une zone particulière car si l'objet contrôlé par le joueur meurt, il sera re-généré dans cette zone

Breakable

life : int
breakers : List <Object>
typesBreakers : List<Type>

en + : un objet qui a 2 états soit normal, soit cassé (ex : une porte, une fenêtre), si la vie descend à 0, le booléen broken passe à true, il y aura 2 objets 3D pour chaque état, si broken est à true, isTraversable passe à true, l'objet 3D n'est pas destroyed

Construction

name : String
entrances : List<Zone>

en + : pour des rpg par ex, on ne crée que l'extérieur d'un bâtiment et dès qu'on entre par une entrée, on est automatiquement téléporté dans une pièce en 3d

Room

name : String
exits : List<Zone>

en + : une pièce, les collisions sont gérées différemment (par face) car si le personnage est à l'intérieur, il est déjà en collision

Ball

name : String
number : int

en + : volume englobant sphérique, permet les jeux de billard, volley, foot, bowling

...

...

en + :

si les types ont des pts de vie (comme character ou breakable) / (attack et damages sont des attributs

enemy < touches > ally : enemy < damages/attack > ally
enemy < touches > player : enemy >> damages/attack >> player
enemy < touches > enemy : rien
ally < touches > player : rien
ally < touches > ally : rien
player < touches > player : impossible
neutral < touches > enemy : neutral << damages/attack << enemy
neutral < touches > ally : rien
neutral < touches > player : rien
neutral < touche > neutral : rien

exemple avec un projectile (pas de pts de vie mais damages) et un personnage
projectile ally < touches > character enemy : enemy < damages/attack > ally : character << damages << projectile
projectile ally < touches > character ally : rien