

If you are going to use a method another module has, make sure they have it under their Methods Other Modules Call List!

If you are going to make a method for another module to use, make sure they have it under their Methods We Will Call on Other Modules List!

If a method is on both lists, don't make it vary from it's purpose without telling whoever uses it, and don't expect it to do something it doesn't yet without telling the people who made it!

If a methods is not on both lists, talk to whoever would be using it before implementing it--otherwise someone might have to change their implementation! Any method not on both lists should be preceded by `/////` to denote this.

Authoring

Methods Other Modules Call:

`WorldData.getTileData(int x, int y)`: Returns the `TileData` object of the given coordinates

`TileData.getImageName()`: Returns the image name of the given tile--calling `Data's getImage` method with this name would yield the tile's image

`TileData.getGridObjectDatas()`: Returns the list of `GridObjectDatas` present on this tile

`GridObjectData.isSteppable()`: Returns whether or not the `GridObjectData` is steppable

`GridObjectData.getTileData()`: Returns the `Tile` (given by a `TileData` object) this `GridObjectData` is on

`GridObjectData.getImageName()`: Returns the image name of the given tile--calling `Data's getImage` method with this name would yield the tile's image

Methods We Will Call On Other Modules:

`Data: storeImage(String name, File image)`

`getImage(string name)`

`storeWorldData(String name, WorldData world)`

`getWorldData(String name)`

`storeGridObjectData(String name, GridObjectData t)`

`getGridObjectDatas()`

Data

Methods Other Modules Call:

Authoring Environment will call on `Data`:

`storeImage(String name, File image)`

`getImage(string name)`

`storeWorldData(String name, WorldData world)`

`getWorldData(String name)`

`storeGridObjectData(String name, GridObjectData t)`

`getGridObjectDatas()`

Player will call on Data:

getWorldData()

getImage(String s)

Player

Methods Other Modules Call:

Methods Player Calls:

From Engine:

/////Tile: Tile()

/////Engine: Engine()

From Data:

getWorldData(int x, int y)

getTileData ()

getImageName()

getGridObjectDatas()

GridObjectData()

isSteppable()

getTileData()

getImageName()

getImage(String s)

Game Engine

Methods other modules call:

/////Tile: isSteppable()

Engine: gameloop(),checkCollisions(),checkGoals()

/////Sprite: doConversation(),setImage/Animation(),move(),getXPosition(),getYPosition()

/////DialogueBox:showDialogue()

/////setGoals() - set the goals for this game

/////NPC: giveItem() (triggered by event?)

/////Player: useItem(),die(),storeItem()

/////World:initializeBattle() - Arena mode

/////Items:isEquipable(),isUsable()

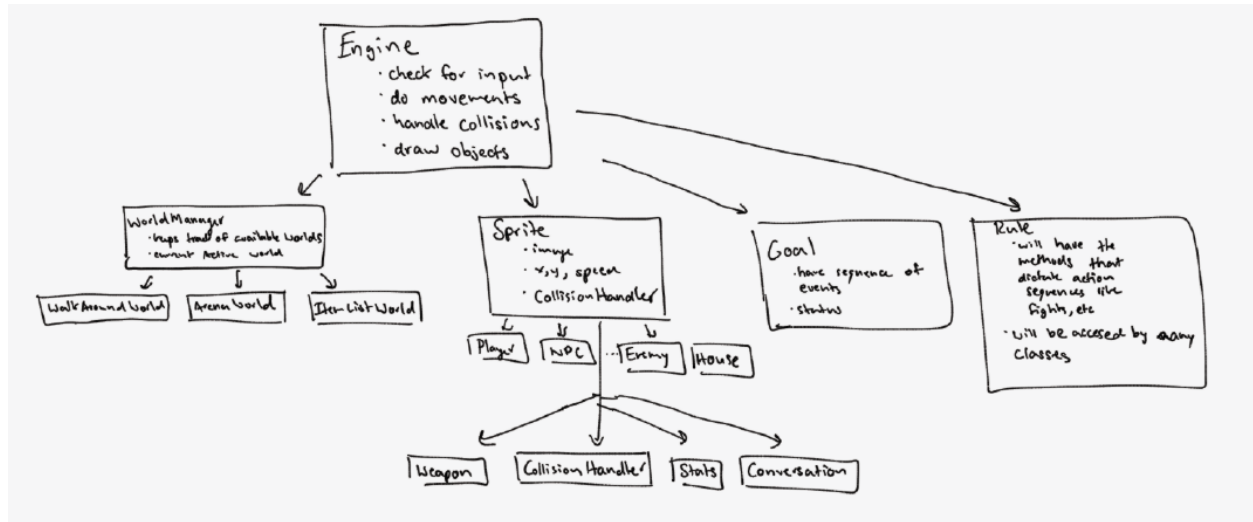


Figure: UML diagram for Engine

Notes:

Elements

Objects:

“Tile” objects

“Non-tile” objects: This would include buildings, structures, players, enemies,

Doors,

Menu:

Character has items, levels, inventory, attributes.

Turn Based combat/battle system

Quests:

Cutscenes

MapOptions: does the map move, does the map follow you, multiple maps.

Music: Let music be uploaded, be played during certain time frames.

Tile properties:

Whether it allows movement across it, and for what types of things (ie, enemies can cross but player cant, etc)

What objects can be placed on it: NPC, enemy, Player, structure, item.

Whether it initiates a cutscene, quest, battle