**Naming**

gameX: coordinate in the game

scrX: coordinate in the screen

csvX: data from csv file

Floor goes to Bottom:

Target, Switch goes to Middle;

Player, NPC goes to TOP

You can devlare a parent class and store children class element. However if you want to use chld class method you need to do polymorphism

**Programmming**

**Undo:**

**Skeletong’s move will not be reversed**

World.update:

Update order: Skeleton -> player -? Rogue and Mage

Render Order:

Floor, target, switch is at bottom, rest are at top

Rogue.update:

We assume rogue move every time player try to move. To make game easier.

That is, player keep pressed right, rogue will move

Turn around is counted as a action, other wise it is too hard to play

**Change fro UML**

Remove Tile.canGoThrough

Adding a collision function to all sprite

**USELESS Function**

/\*\*

\* **TODO** rewrite the pushing stone function

\* This function used to perform pushing stone

\*/

**public** **static** **boolean** pushStone(**float** nextX, **float** nextY, **int** stoneID, **int** input) {

/\*this function take the planning x and y as input

\* and check whether this stone can be pushed

\* if successful, return true

\* else, return false

\*/

**for**(**int** i = 0; i < *sprites*.length; i++) {

Position spriteGamePosition = *sprites*[i].gamePosition;

**float** BlockedX = spriteGamePosition.gameX;

**float** BlockedY = spriteGamePosition.gameY;

String tileType =*sprites*[i].tileType;

**if** (nextX == BlockedX & nextY == BlockedY) {

// if we find the block, we try to push it

**if** (tileType.equals(***WALL***) | tileType.equals(***STONE***)) {

**return** **false**;

}

}

}

*sprites*[stoneID].update(input);

**return** **true**;

}