1. A discussion on how your design for assignment 1 helped or hindered your extensions made in this assignment

Change:

Consecutive instances of platform -> one instance of platform with width of levelWidth,

Because of performance issues.

JSONReader has now be renamed to JSONInterpreter and be moved to model package. As the name suggest, it is now responsible to not only reading JSON file, but also using EntityFactory to initialize the Level and assign Entity(s) to the Level.

Kept:

Level is still responsible for how the ballboy move.

GameEngine still handle all the entity interaction and instruct the entity in the currentLevel to progress.

Entitys are still split into MovableEntity and StillEntity, this help reducing execution time in LevelImpl.tick(). Tick() determine the interaction between Entitys, and this segregation introduce a first round of type check, so each iteration can run the first round of type check before type check for each type. For instance, only MovableEntity should invoke move() on each iteration.

1. Rationalise changes you have made to your assignment 1 design

BlockedBackground will now take the config as JSON states to enable different appearance for level, as the specification required.

To enable the ballboy “continuously bouncing”, Enemy and Ballboy now will be affected by gravity.

Ballboy now hold a few more attributes include

* landed; if landed is true, Ballboy now will bounce() on each tick(). Landed will be turned to true every time collide with instances of Platform. Landed will be turned to

1. A discussion on each design pattern you have used including
2. Where you used it (be explicit as to what classes are involved and in what roles)
3. What this pattern does for your code in terms of SOLID/GRASP principles
4. What overall benefits this pattern provides (be specific to your code, not the pattern in general)
5. What drawbacks this pattern causes (be specific to your code, not the pattern in general)

A picture containing text, sky

Description automatically generated