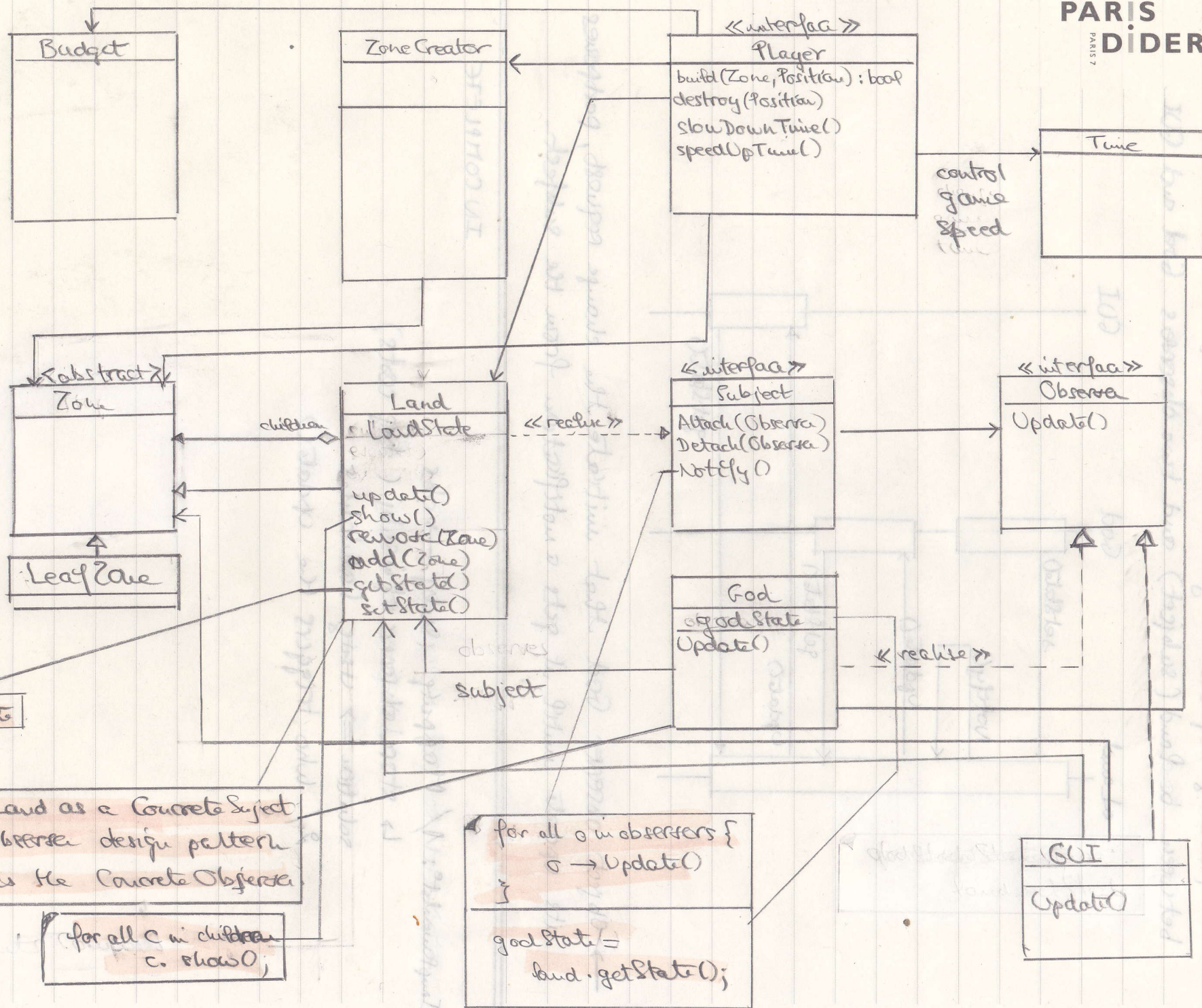


IN COMPLETE



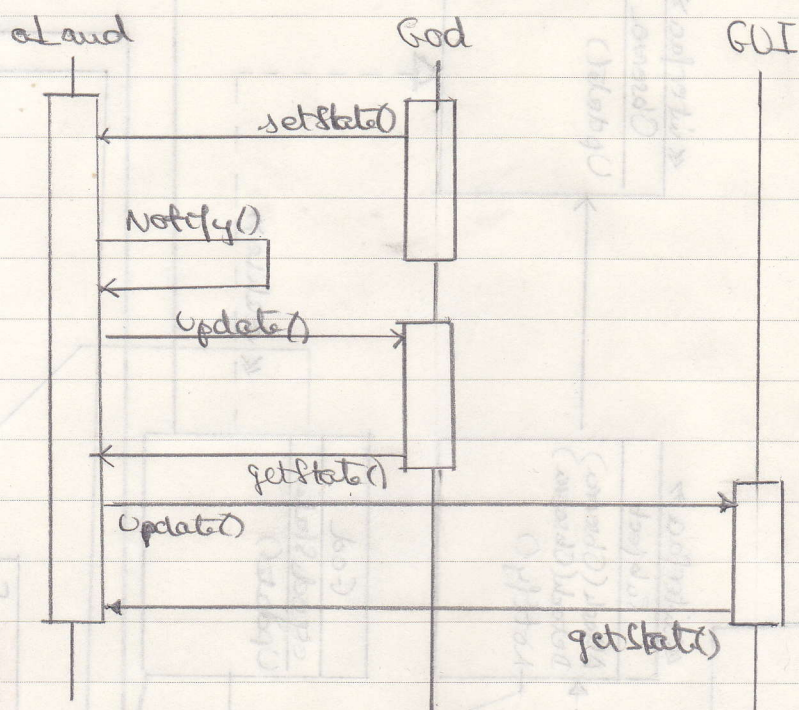
Return LandState

We consider Land as a Concrete Subject for Subject-Observer design pattern and God is the Concrete Observer.

for all c in children
c.show();

for all o in observers {
o → Update()
}
godState =
land.getState();

The following sequence diagram illustrates the collaboration between a Land (subject) and two observers God and GUI.



→ Object observer God that initiate the change request, postpones its update until it gets a notification from the subject.

IN COMPLETE

Implementation: 1/ Mapping subj → obs

↳ store observers ref in subj (big costs)

solution ⇒ using hash tables

2/ Who triggers the update()

- Zone: (component) Abstract class
 - declares the interface for objects in the composition.
 - implements default behavior for the interface common to all classes as appropriate.
- LeafZone (ResidentialZone, CommercialZone, IndustrialZone ---)
 - represent leaf objects in the composite
 - leaf has no children!
 - EX: ResidentialZone can't have a class that inherits from it.
- Land: - defines behavior for components having children
 - stores child components
 - implements child-related ops in the Zone interface
- GUI, Player and Zone Creator are Client for Zones