Input: A map of string to string representing the association of layers to their respective filenames in the repository.
Output: Layers, which is an object containing all layers or components used to build the game.

```
function loadXmlComponents
    layers : Layers;
    jaxbContext : JAXBContext;
    file : File;
    unmarshaller : Unmarshaller;
    for layer ϵ layers – {lesson, challenge}
        jaxbContext ← JAXBContext.newInstance(layer.class)
        unmarshaller ← jaxbContext.createUnmarshaller();
        file ← File(xmlFiles.layer);
        layers.layer ← unmarshaller.unmarshal(file);
    jaxbContext ← JAXBContext.newInstance(lesson.class)
    unmarshaller ← jaxbContext.createUnmarshaller();
    lessons : Lesson[];
    for lessonFile ϵ xmlFiles.lessons
        lesson : Lesson;
        file ← File(lessonFile);
        lesson ← unmarshaller.unmarshal(file);
        lessons ← lessons ∪ {lesson};
    jaxbContext ← JAXBContext.newInstance(challenge.class)
    unmarshaller ← jaxbContext.createUnmarshaller();
    challenges : Challenge[];
    for challengeFile ϵ xmlFiles.challenges
        challenge : Challenge;
        file ← File(challengeFile);
        challenge ← unmarshaller.unmarshal(file);
        challenges ← challenges ∪ {challenge};
    learningObjectives : LearningObjective[];
    for (lesson ϵ lessons) && (challenge ϵ challenges)
        learningObjective : LearningObjective;
        lessonActs : LessonAct[];
        lessonAct : LessonAct;
        lessonAct.lessonScreens ← lesson;
        lessonAct.challengeScreens ← challenge;
        lessonActs ← lessonActs ∪ {lessonAct};
        learningObjective.lessonActs ← lessonActs;
        learningObjectives ← learningObjectives ∪ {learningObjective};
    layers.learningObjectives ← learningObjectives;
    call wireUpLayers;
```