

CptS 487

Software Design and Architecture

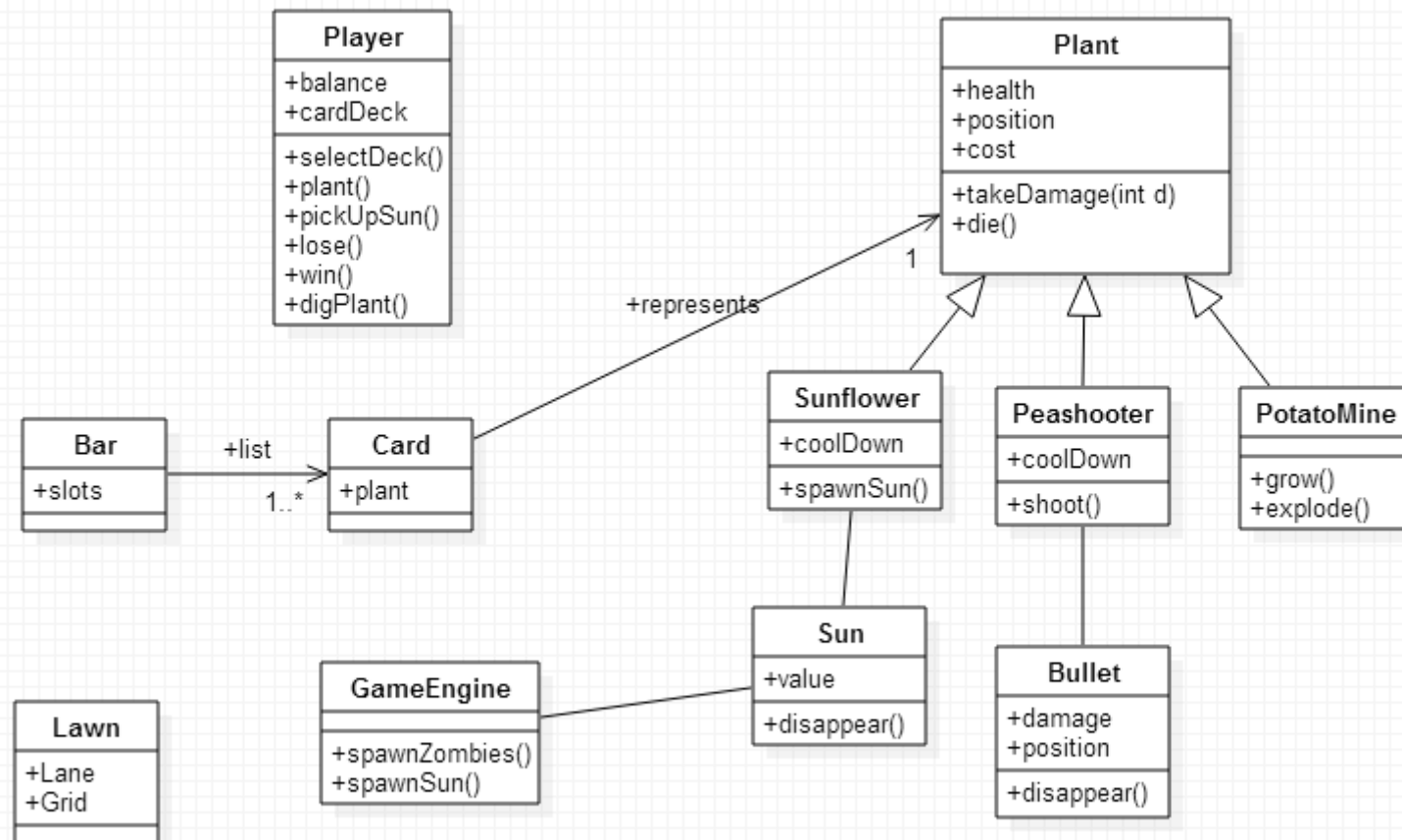
Lesson 11

Analysis Case Study Pt. 2

Outline of this Class

- Use the PvZ example to go over requirements elicitation and analysis stages.
 - Usage of Statechart diagram and Activity diagram
 - Consider how to address new features while maintain flexibility.

Classes First Pass

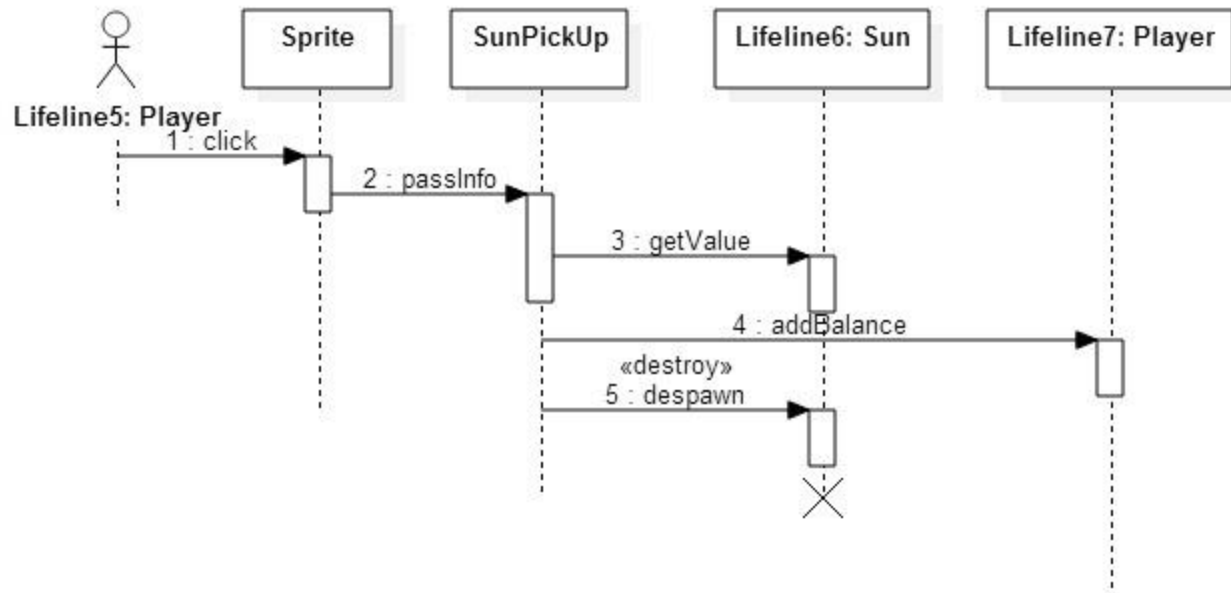


Further Analysis

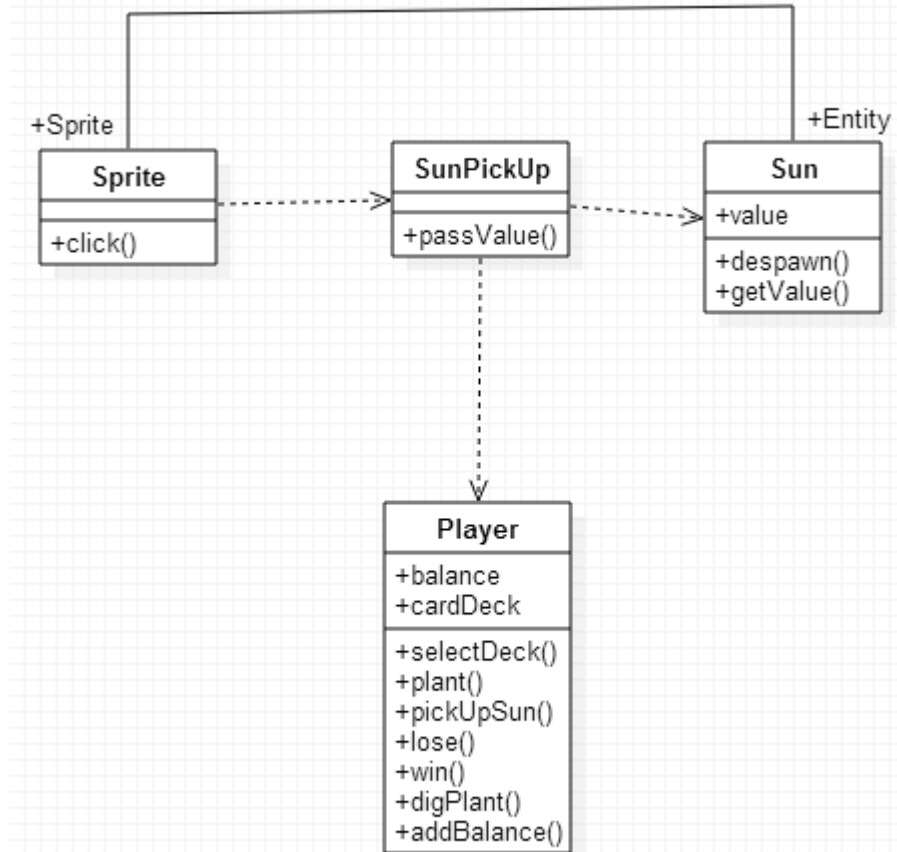
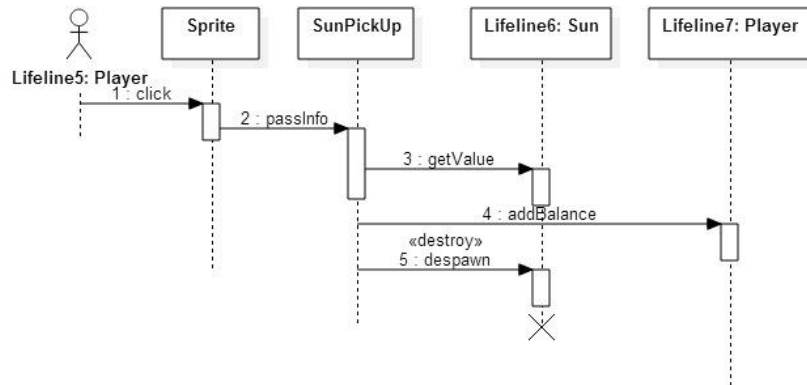
- Sequence diagram of use case(s)
 - 1. How does a player pick up a sun?
 - 2. How does a plant shoot and damage a zombie?

Sequence Diagram Model

- 1. How does a player pick up a sun?



Class Diagram Revision

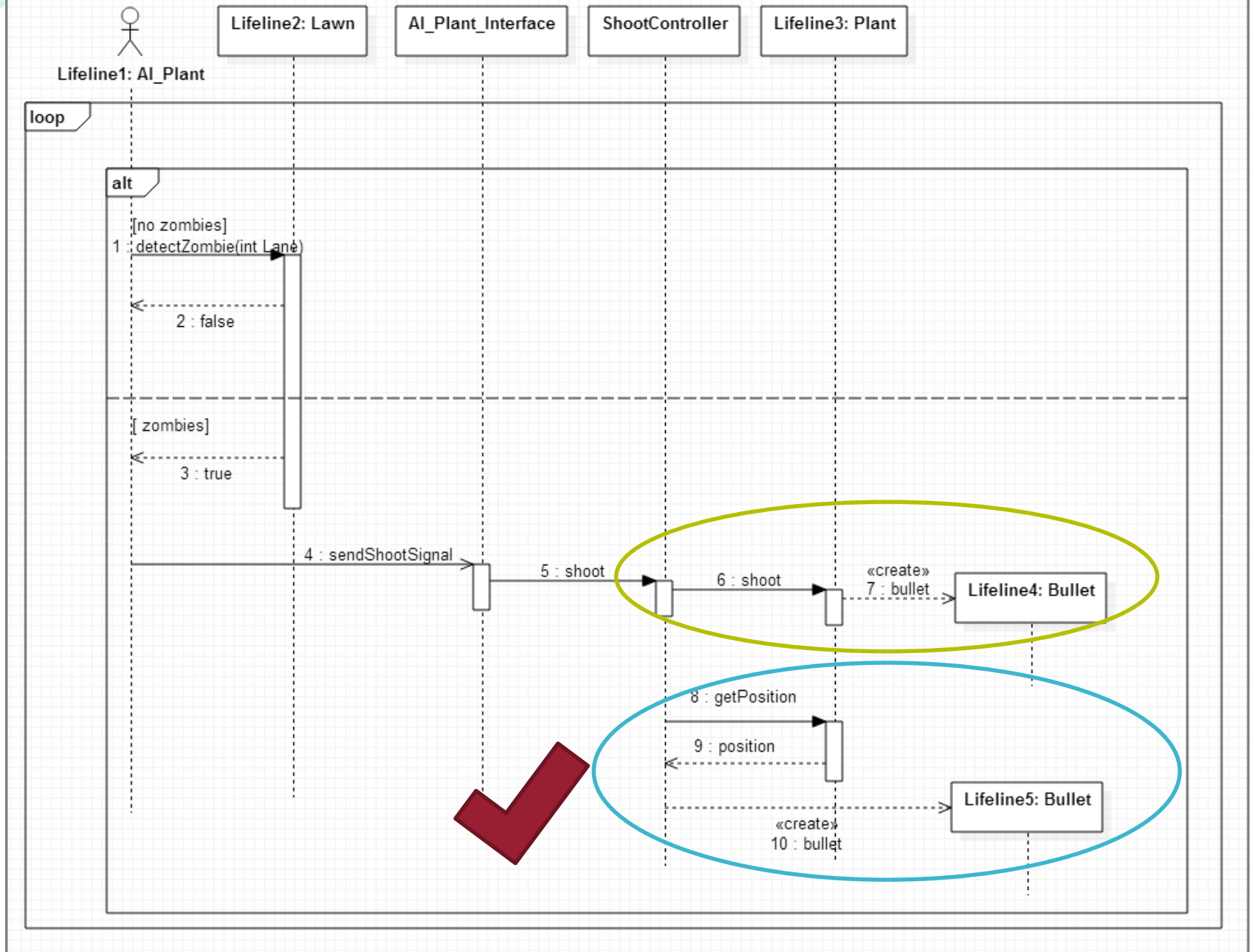


Pick up coins?

Sequence Diagram Model

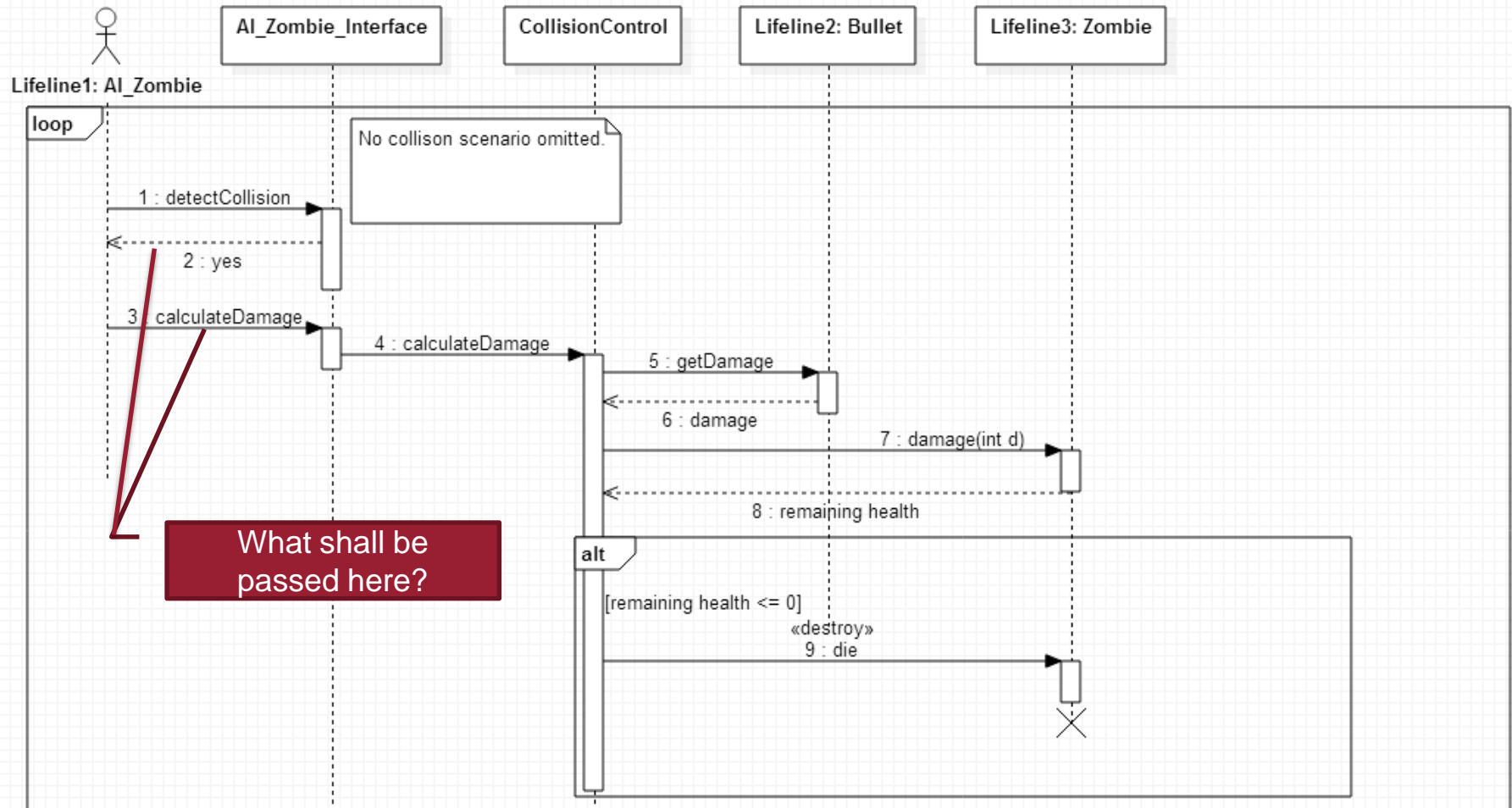
- 2. How does a plant shoot and damage a zombie?
 - Actor -> Boundary -> Control -> Entity
- AI_Plant – Shoot
 - 1. AI detects a zombie in the same lane
 - 2. AI sends a signal to shoot a bullet.
 - 3. System spawns a bullet traveling forward in the lane.
- AI_Zombie – Take Damage
 - 1. AI detects a bullet collides with a zombie.
 - 2. AI sends a signal to damage the zombie.
 - 3. System reduces zombie's health.

interaction SequenceDiagram1



Sequence Diagram Model

- 2. How does a plant shoot and damage a zombie?
 - Actor -> Boundary -> Control -> Entity
- AI_Plant – Shoot
 - 1. AI detects a zombie in the same lane
 - 2. AI sends a signal to shoot a bullet.
 - 3. System spawns a bullet traveling forward in the lane.
- AI_Zombie – Take Damage
 - 1. AI detects a bullet collides with a zombie.
 - 2. AI sends a signal to damage the zombie.
 - 3. System reduces zombie's health.



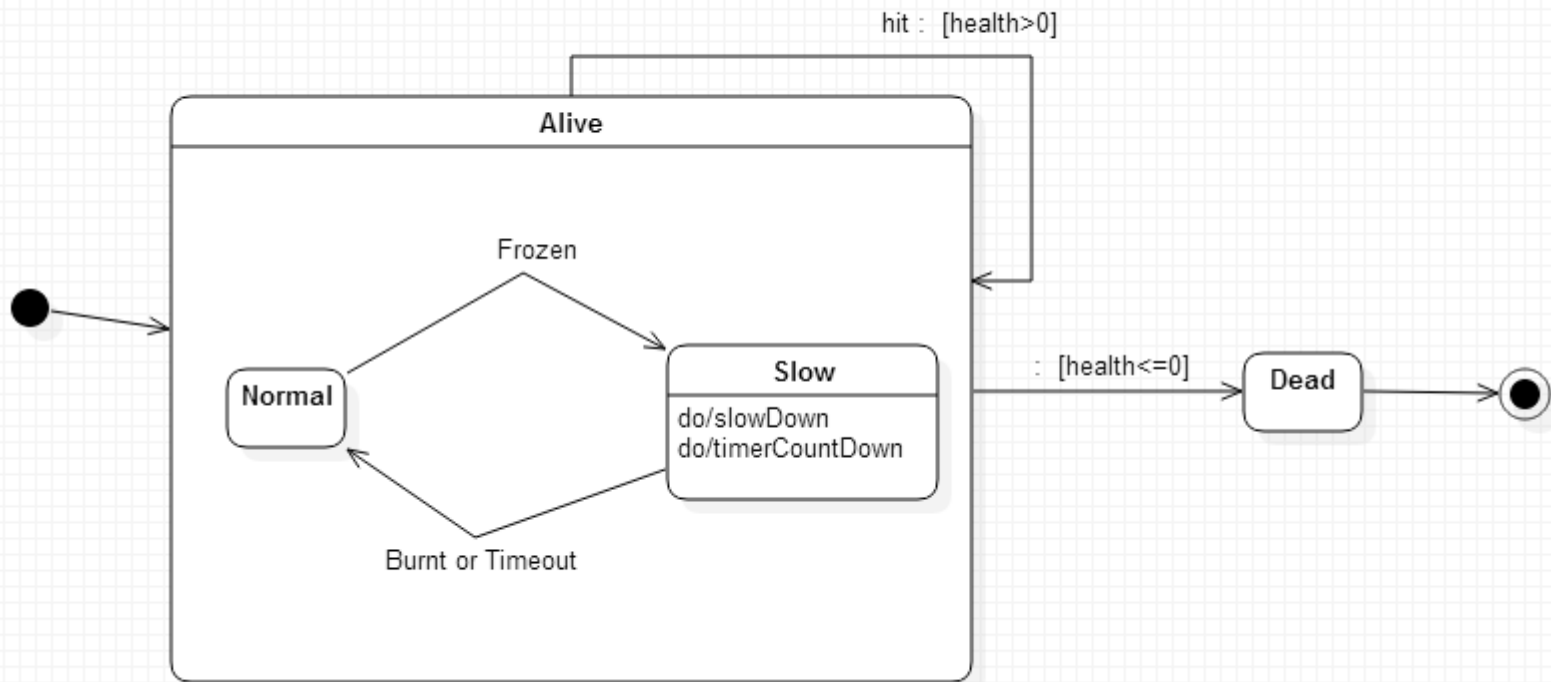
New Feature?

- What about these?

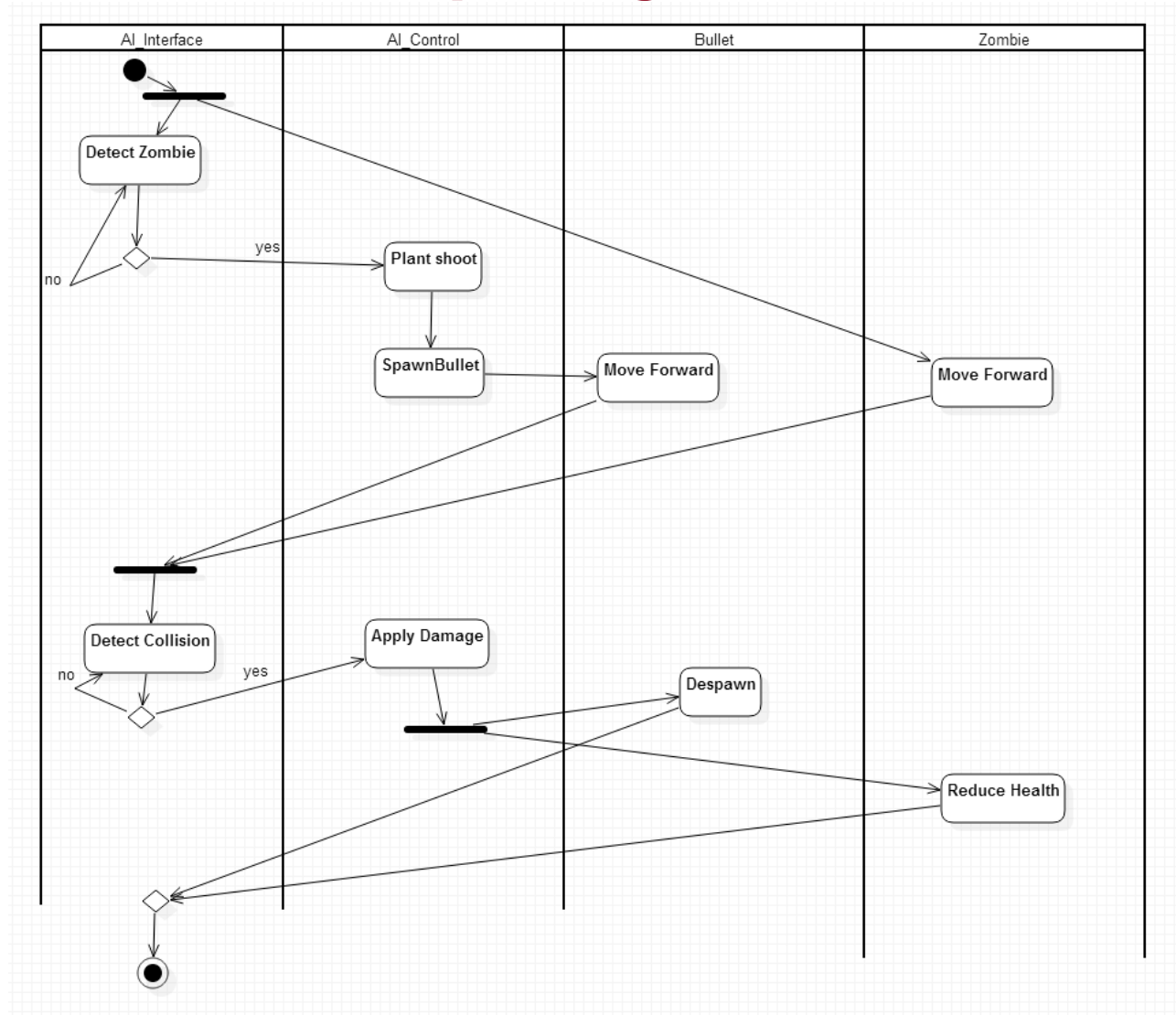


- How many changes?
- Where do we make the changes?

Statechart Diagram Model



Activity Diagram Model



Analysis Conclusion

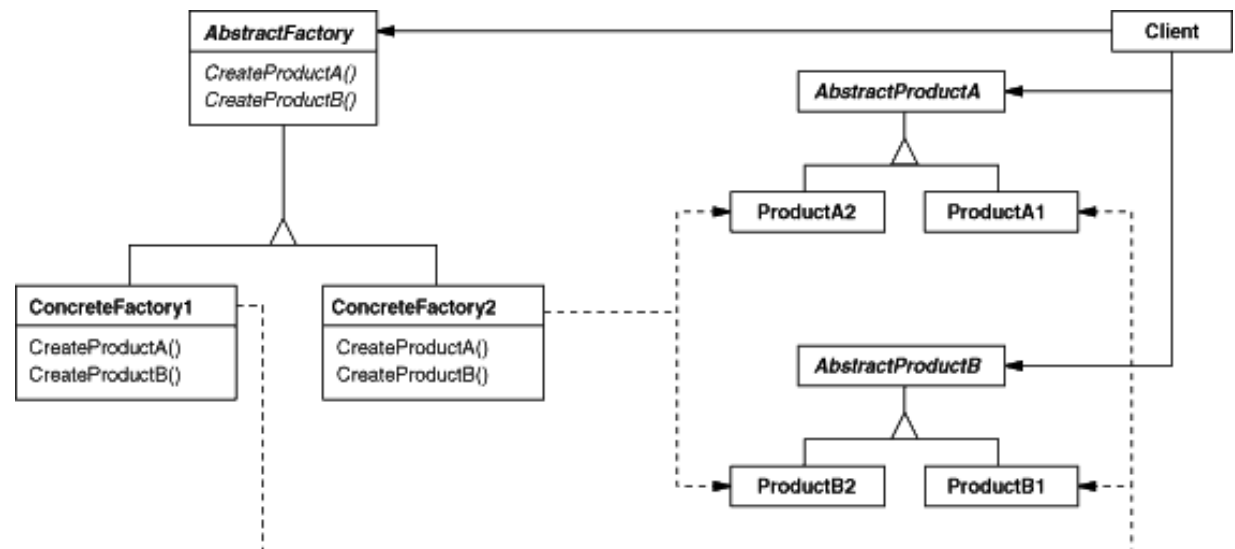
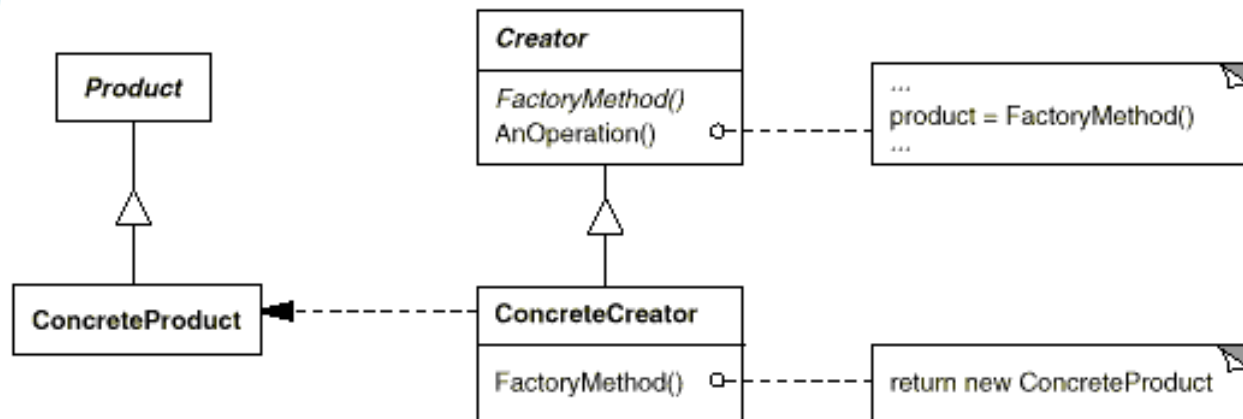
- Use the various UML models to clarify our requirements via analysis.
- Main components is the object model: class diagrams
 - Entity, boundary classes come first; then control classes. (Refer to the previous slides for definition of these 3)
- Sequence/Activity/Statechart diagrams to help refine the details, and identify potential changes needed.

Move on to design...

- Class structures
 - Draw a class diagram with these three types of zombies.
 - If we were to choose from Factory and Abstract Factory pattern, which one seems better?
 - Make your class diagram follow the pattern you chose.

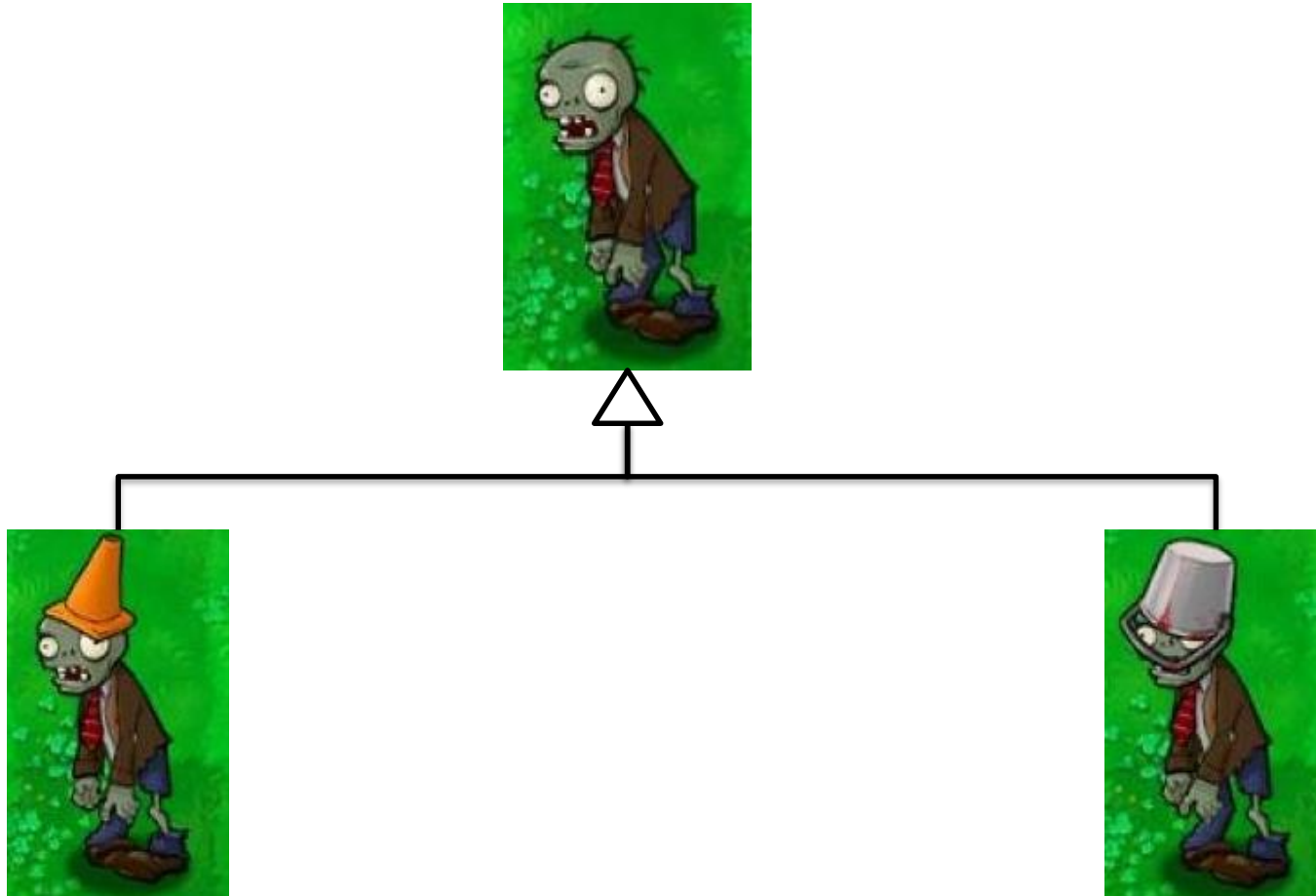


Factory/Abstract Factory reminder



Class Structure

- Direct Inheritance



Take a second look

- What are the differences among these zombies?
 - Health, obviously
 - Anything else?
 - Consider how they take damage

