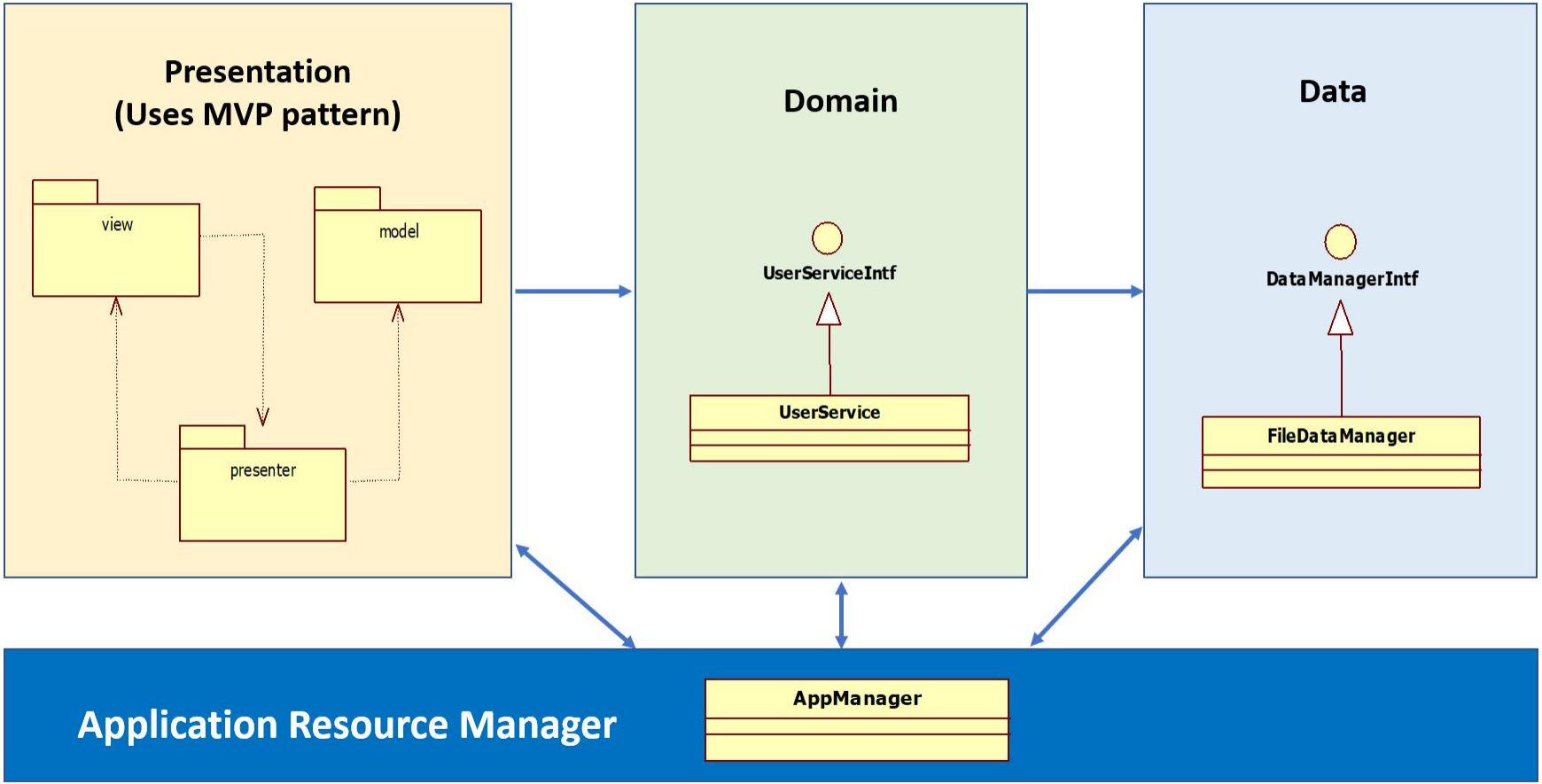
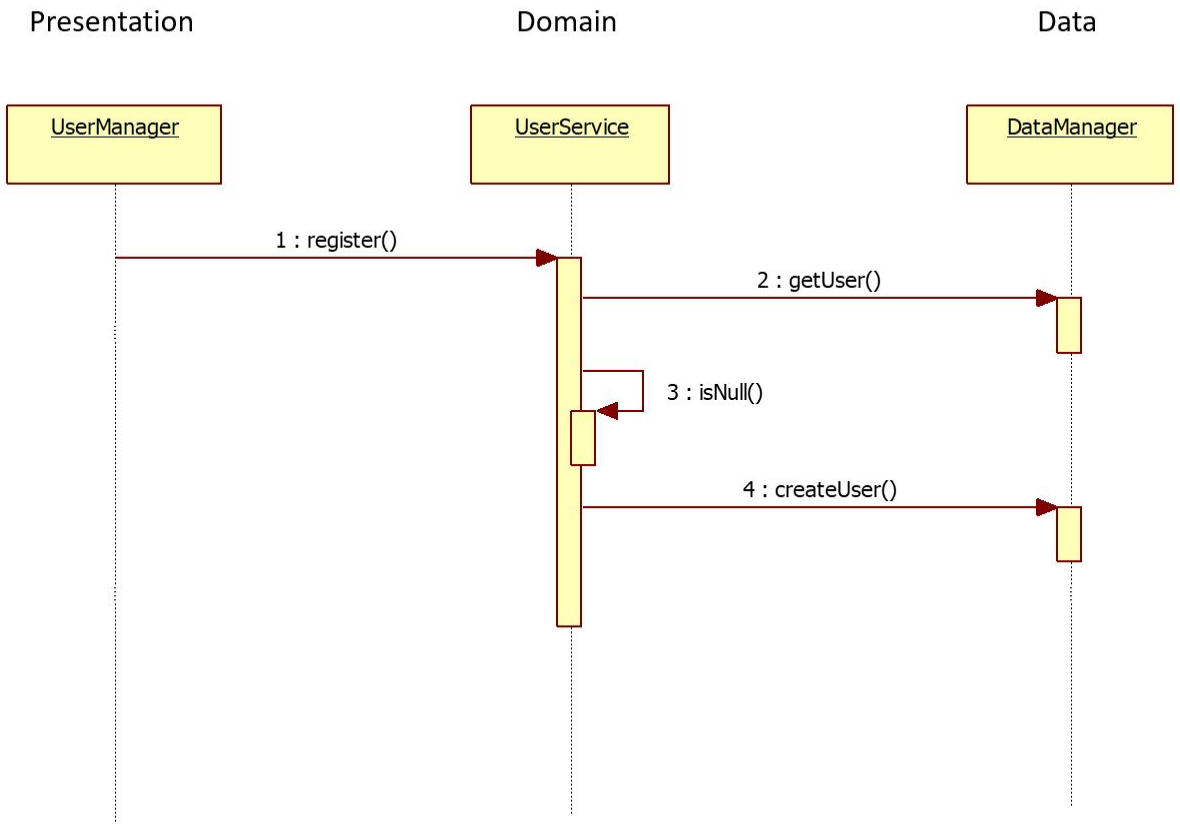
CSC207 Presentation

0635: Abdul Arif, Jean Li, Ashley Lu, Luqiong Xie

Our high level architecture:



Sequence diagram for Registration (happy path):



Addressing Phase 1 Comments

- Unification of naming conventions and coordinate systems
- Unification of inheritance hierarchies
  - While adhering to Liskov substitution principle

Extension #2: Adding a new game

- Added a fourth game (brick game)
- Required minimal modifications; mainly added new code (Open-Closed Principle)
- Unique property: bricks have a damaged and undamaged appearance
    - Only the key needed to be changed; View classes unaffected

Adding Animations

- Changed GameViews from having single bitmap to a list of bitmaps
- Implementation of frame management hidden by using a common getAppearance method

Game Design (Apple, Tapping, Jumping):

- Easy for Extension (E.g. Add new game items, new input variables, new statistics logic)
- Good Encapsulation (E.g. Change update logic, statistics logic)

GameManager
- update() update GameItems and statistics by applying dependency inversion

```
Update () {

    JumpingMovementInfo jumpingMovementInfo = new JumpingMovementInfo(....);
     for (GameItem item : getGameItems()) {
            Result result = item.update(jumpingMovementInfo);
            // process result
            updateStatistics( result);…
            // If (result.isStarCollected){numStars +=1;}
            // ...
     }
}
```

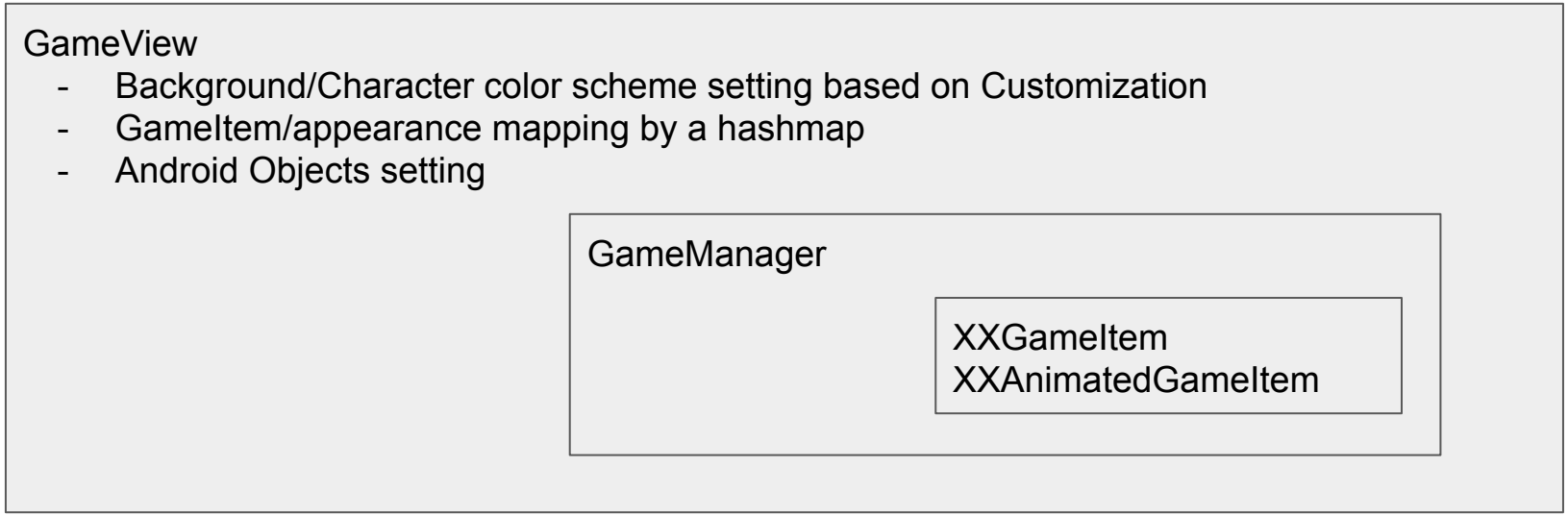
XXGameItem / XXAnimatedGameItem
- height, width, coordinates
- AnimatedGameItem: velocity, acceleration
- Result update(XXMovementInfo)
- Result update(MovementInfo movementInfo) {}

Result
- Variables needed by XXGameManager to update game items and statistics

MovementInfo
- numOfSeconds
- Variables needed by GameItem.update()

Game Design (Apple, Tapping, Jumping):

GameView only accesses GameManager for data extraction and updates, and doesn’t access GameItems directly.



SOLID Principles & Design Patterns

Single Responsibility and Open-Closed: xxGameManager

Dependency Inversion:

UserServiceIntf and DataManagerIntf (discussed earlier), xxGameManager.update()

Builder: AppleItemsBuilder, BrickItemsBuilder

Observer: GameStateObserver (discussed earlier)

Factory: GameManagerFactory (discussed earlier)

Singleton: (AppManager) (discussed earlier)

Improvements That Could Have Been Made

1. BrickMovementInfo’s implementation
2. Move musicPlayer: MediaPlayer variable to GameView class since it is an Android Object
3. Applying stricter access modifiers for variables and methods to protected, package-private, or private.
4. Make TappingGameView.surfaceCreated() shorter by creating a builder
5. Make XXGameView.extractBmpFiles() shorter by creating helper methods
6. Consolidate multiple background extraction methods to a single method in GameView since they are used by each XXGameView.
7. Change GameItem constructor signatures to width, height (instead of height, width)
8. Remove unused GameItem constructors
9. Remove unused methods