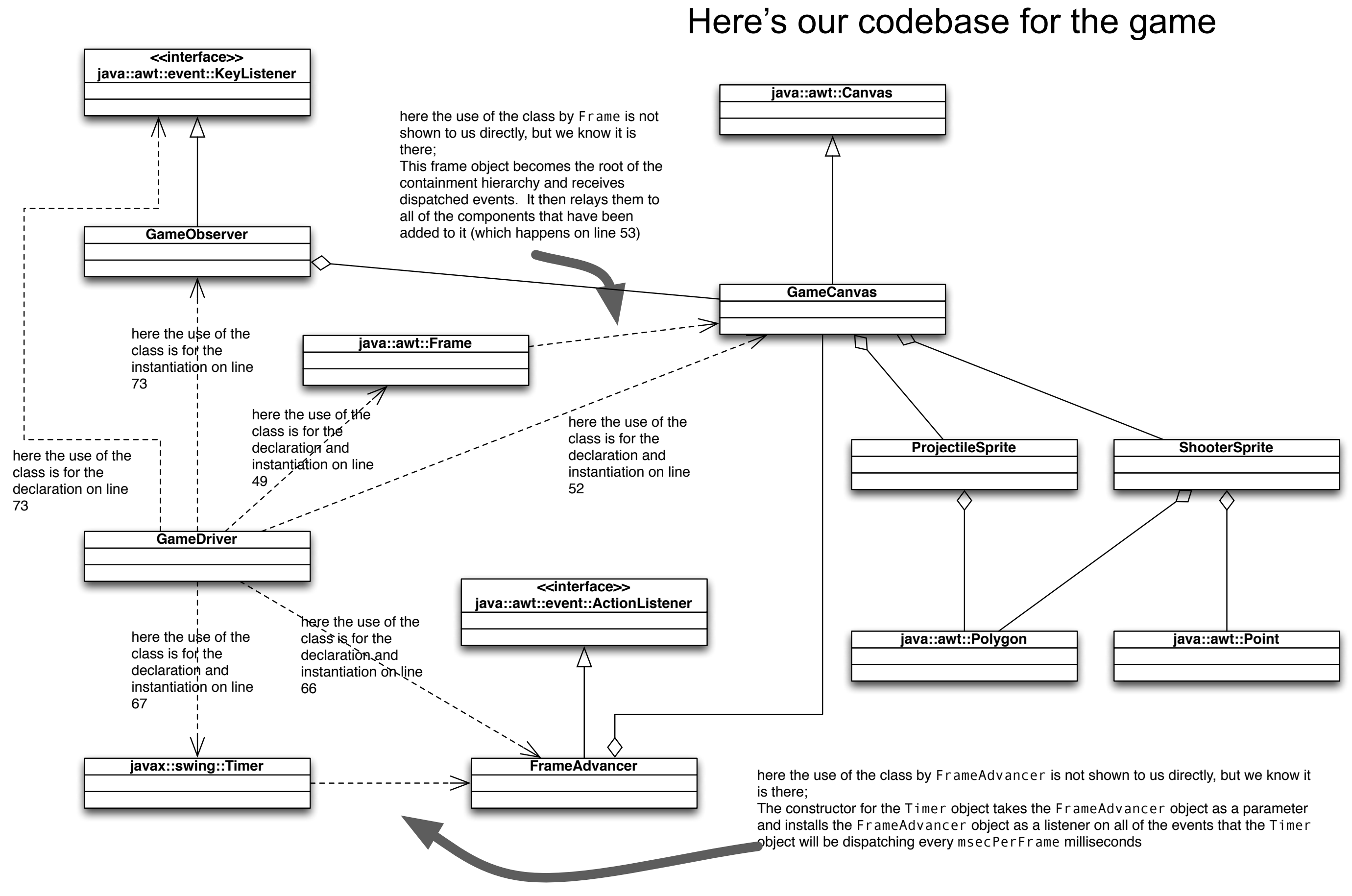
CSE1720

Week 09, Class Meeting 22 (Lecture 18)

Winter 2013 ◆ Tuesday, March 5, 2013

Here's our codebase for the game



<<interface>>
java::awt::event::KeyListener
java::awt::Canvas
GameObserver
GameCanvas
java::awt::Frame
here the use of the class is for the instantiation on line 73
here the use of the class is for the declaration and instantiation on line 49
here the use of the class is for the declaration and instantiation on line 52
here the use of the class is for the declaration on line 73
ProjectileSprite
ShooterSprite
GameDriver
<<interface>>
java::awt::event::ActionListener
here the use of the class is for the declaration and instantiation on line 67
here the use of the class is for the declaration and instantiation on line 66
java::awt::Polygon
java::awt::Point
javax::swing::Timer
FrameAdvancer

here the use of the class by `Frame` is not shown to us directly, but we know it is there;
This frame object becomes the root of the containment hierarchy and receives dispatched events. It then relays them to all of the components that have been added to it (which happens on line 53)

here the use of the class by `FrameAdvancer` is not shown to us directly, but we know it is there;
The constructor for the `Timer` object takes the `FrameAdvancer` object as a parameter and installs the `FrameAdvancer` object as a listener on all of the events that the `Timer` object will be dispatching every `msecPerFrame` milliseconds

Model-View-Controller Architecture

This architecture is our goal

Model

this module implements:
- what is the state of the system?
such as
- what is the current score?
- whose move is next?
- how close to the end?
- what actions are allowed right now?

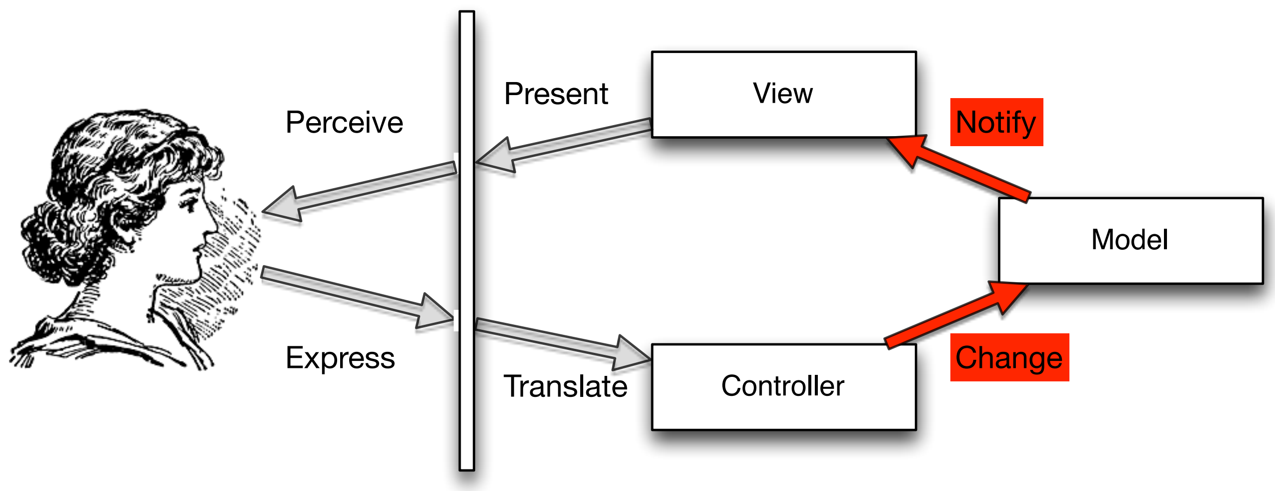
View

this module implements:
- what does the user see/hear
- what actions can the user perform?
aka the user interface

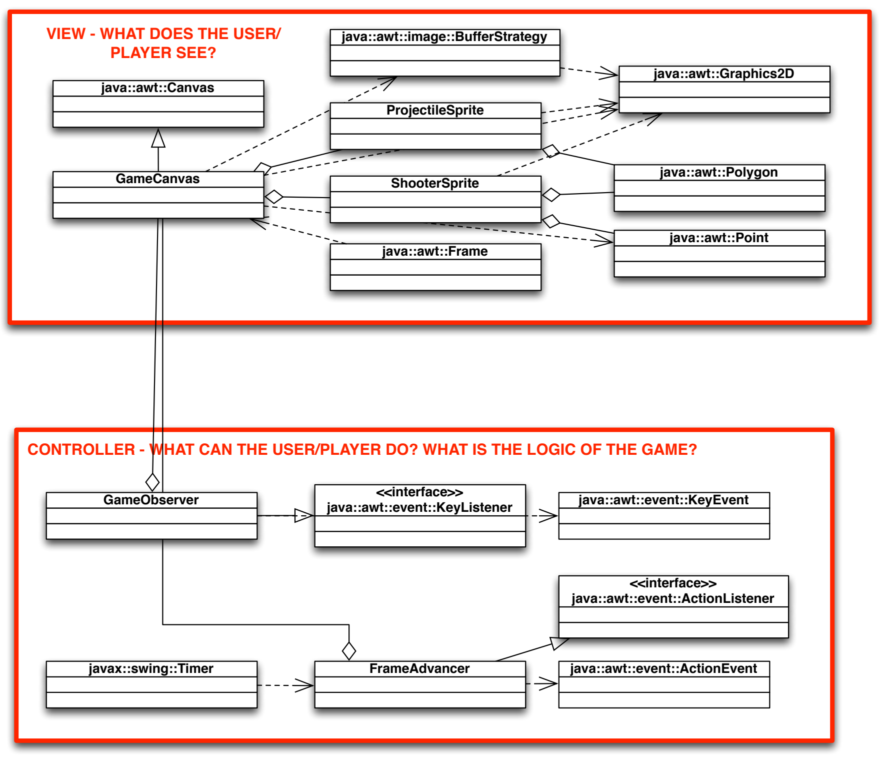
Controller

this module implements:
- the logic of the game
- given a user action, what is the impact
on the state of the system?
- given other events (clock ticks, countdown
timers), what are the impacts on the system's
state

Schematic of MVC



Discussion of Codebase

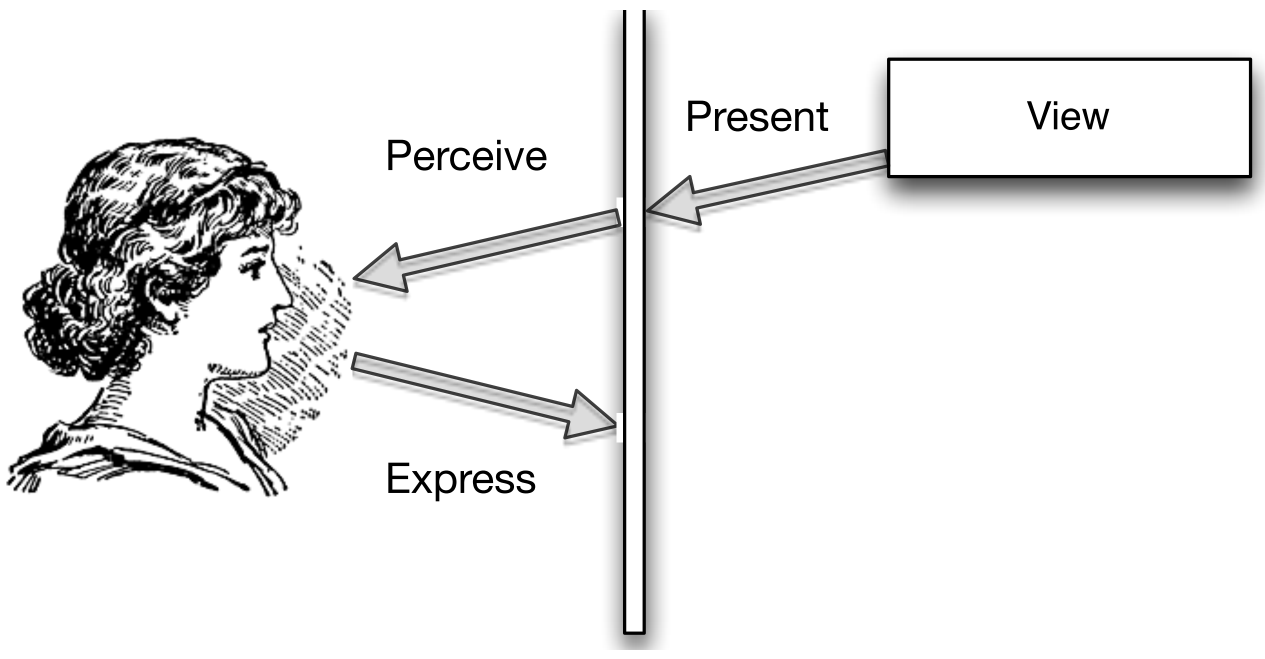

VIEW - WHAT DOES THE USER/ PLAYER SEE?
java::awt::image::BufferStrategy
java::awt::Graphics2D
java::awt::Canvas
ProjectileSprite
GameCanvas
ShooterSprite
java::awt::Polygon
java::awt::Point
java::awt::Frame
CONTROLLER - WHAT CAN THE USER/PLAYER DO? WHAT IS THE LOGIC OF THE GAME?
GameObserver
<<interface>> java::awt::event::KeyListener
java::awt::event::KeyEvent
<<interface>> java::awt::event::ActionListener
javax::swing::Timer
FrameAdvancer
java::awt::event::ActionEvent

MODEL - WHAT IS THE STATE OF THE GAME?

what should be appearing on screen?
what is the current score?
whose move is next?
which actions are allowed next?
....

5

Schematic – consider the View



6

Improvements Needed!

- Components that are associated with the **view** should only concern themselves with the "how" of the game's appearance, not the "what" of the game
  - the "what" of the game concerns which sprites are on-screen, the current score, the targets, the type of projectile, etc
  - these sorts of things are encapsulated within the `GameCanvas` class
  - we need to `GameCanvas` object to make use of a data model

Updated codebase

- Examine the codebase distributed with this class meeting (class meeting #22).
  - See how the `GameCanvas` object has delegated the task of representing *which* sprites to be drawn to the data model.
  - See how the `Controller` mutates the state of the data model when the user performs the various types of actions.