TDT4240 - Excerise 2

Linn Vikre
Jim Frode Hoff

2014-02-07

1 Introduction

3.a) For the patterns listing in Step3, which are architectural patterns and which are design patterns? What are the relationships and differences of architectural patterns and design patterns?

Observer	2 Design Pattern
State	Design Pattern
Template Method	Design Pattern
MVC	Architectural pattern
Abstract Factory	Design Pattern
Pipe and Filter	Deisgn Pattern

3.b) How is the pattern you chose realized in your code? (Which class(es) works as the pattern you chose?)
We chose to implement the MVC-pattern in our assignment. In our implementation, the ¡VÅR KLASSE!¿AnimateSprite-class works as the model, and the ¡VÅR KLASSE¿GameScreen works as the View- Controller. Whenever the controller (the ¡VÅR KLASSE¿TouchListener) recieves a ¡VÅR EVENT¿TouchEvent and needs to move the sprite, it's call the setPosition-method of the ¡VÅR SPRITE¿AnimatedSprite. The ¡VÅR SPRITE¿AnimatedSprite class then sends a ¡VÅR SPRITE¿ChangeEvent to notify any classes listening to the model. This event is recieved in the ¡VÅR SPRITE¿GameScreen class, which then updates the textbox showing the position of the helicopter.

3.c) Is there any advantages in using this pattern in this program? (What are the advan- tages/disadvantages?)
An advantage of using the MVC-pattern is that the setLabel-method is only run at changes in the X- and Y-coordinates.