

PokemOz

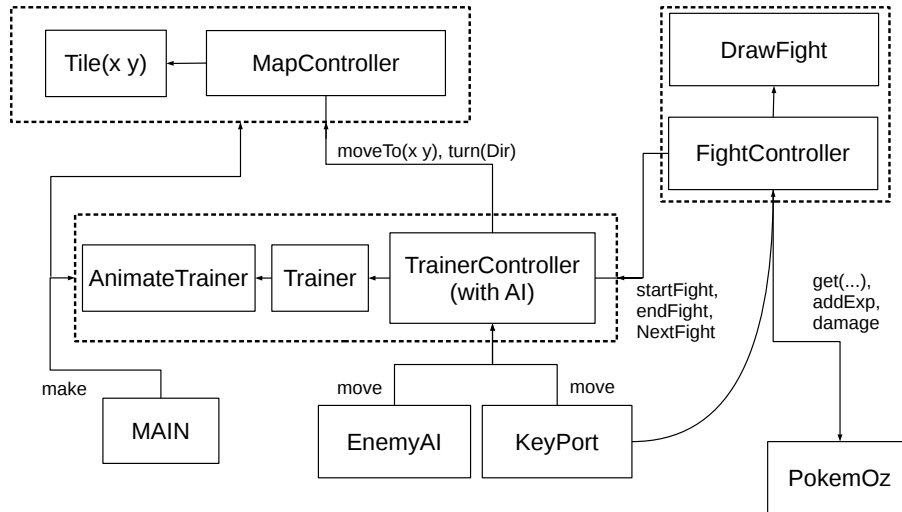
Antoine Vanderschueren and Victor Joos

May 4, 2015

`\documentclass[article]{memoir}`

1 Component Diagram

Our component diagram is based in a large part on the lift example in section 5.4 of CTMCP ¹.



Every one of these components are modeled using **NewPortObject** or an alternative **NewPortObjectKillable** which allows the game to stop the thread when it is no longer needed, to save on resources.

¹VAN ROY, P., HARIDI, S., *Concepts, Techniques, and Models of Computer Programming*, The MIT Press, Cambridge.

2 State Diagrams

We programmed every part of our program using port-objects. Using these port-object allowed us to make an easily testable system using states. In the following section, we will show the state diagrams for the stateful port-objects.

Tile

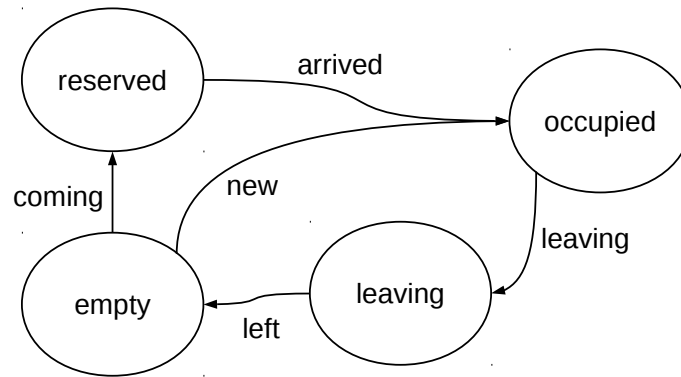


Figure 1: Tile State Diagram

A Tile on the map has an easy state diagram. Each tile has a set of fixed coordinates that can be used by other port-objects to send a tile some messages, through the MapController. The **reserved** and **leaving** intermediate states allow a tile to refuse new Trainers wanting to go on a tile while another trainer is not yet on the tile, but is animating to it at the moment.

PlayerController

This state diagram shows the states of both the PlayerController and the Trainer port-objects. The TrainerController receives “keys” from the keyboard or the Artificial Intelligence, and will then move the Trainer on the map or start a fight, and relinquish control to the FightController.

FightController

The last important port-object is the FightController

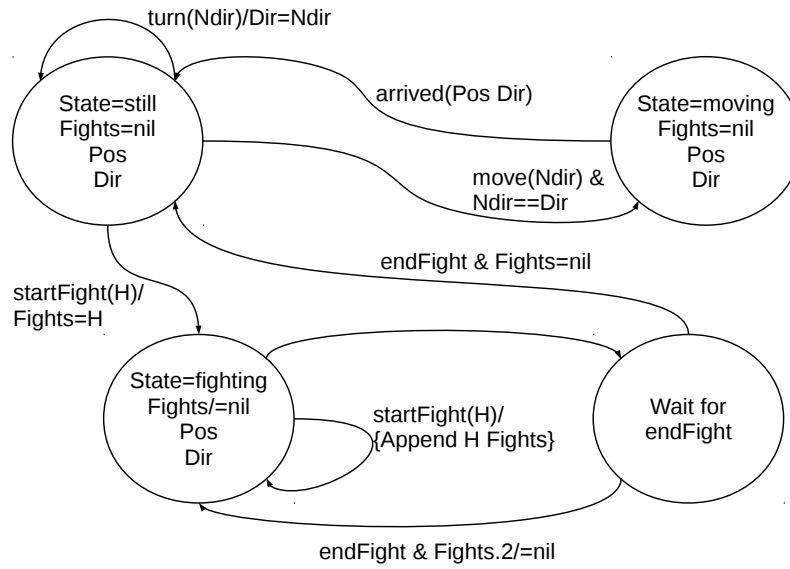


Figure 2: PlayerController State Diagram

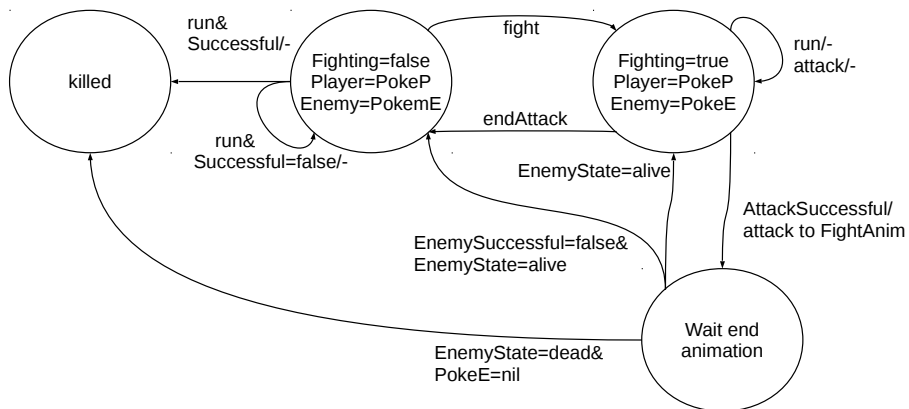


Figure 3: FightController State Diagram