

Building a Finite State Machine

Matt Goff, Anton Morozov, Dan Wolf

Design Goals 1. Simulate FSM

- a. Deterministic
- b. Non-deterministic
- c. Can input an input list (or key stroke)
- d. If input is invalid, enter a reject state
- e. Simulating from a list (has a pause/play button and adjust speed slider)
- f. If, when simulating (non-deterministic) and there are two current states, and the next input is only valid for one, then you are in a real-state and a rejecting-state, then disregard the rejecting state!

 (or leave the rejecting state as a current state and all inputs move it to itself)

2. Make at least 2 export formats

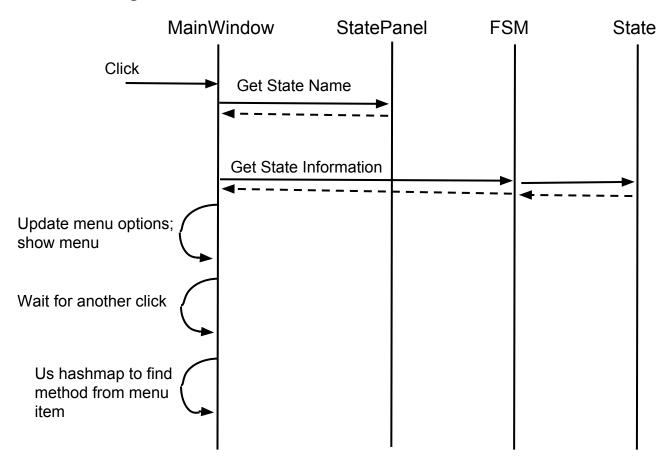
- a. Exporting file; loading optional
- b. One file format that could copy and paste.
- c. One file format that you could export as a PNG file

3. Other

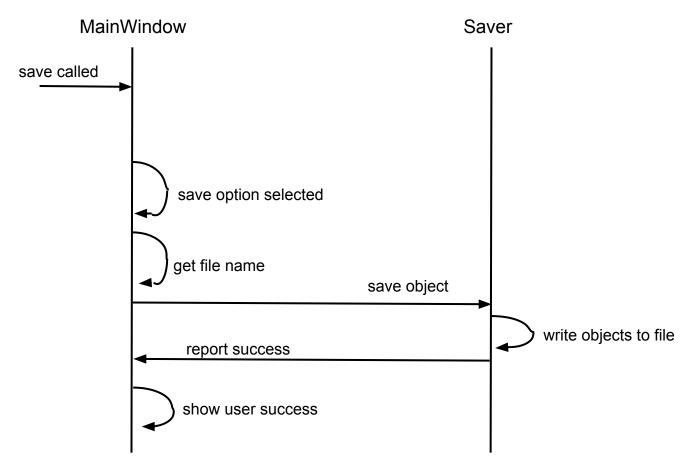
- d. Create separate command classes
- e. More tests
- f. Draw transitions

Class Interaction Flow Chart (top-level design) observes package "mode **FSMRunner** package "view instantiates theMachine: FSM Observer running: boolean package "util Loader notifyObserver () FSMRunner (theMachineParam: FSM) Commands load (filename: string): Object getAvailableFileNames (): String[] Saver MainWindow **FSM** save (objects : objects, fileName : String) states: LinkedList MainWindow () validFileName (fileName: String): boolean FSM () initWindow () getStateByName (stateName : String) : State initMenuBar () Command addObserver (newObserver : Observer) notifyObservers () runCommand () **PixelCoords** StatePanel serialVersionID: ID StateWidth: int xcoord: int StateHieght: int vcoord: int PixelCoords (xcoordParam: int) paint (newName: String) State StartingState: boolean CharacterTransition acceptingState: boolean myLabel: String mvChar: Character transitions: LinkedList nextState: String StateLocations **Transitions** serialVersionUID: ID number changeName (newName : String) locations: pixel coordinates removeState (stateLabel: String) CharacterTransition (myCharParam: Char, getLabel () nextStateParam : String) addTransition (t:transition) StateLocations () nextState () removeTransition (to : String , transitionChar putStateLocation (state : string, location : equals (other: Object): boolean PixelCoords) getLabel (): Char : char) getStateLocations (state : String) : nextState (): String Coordiantes

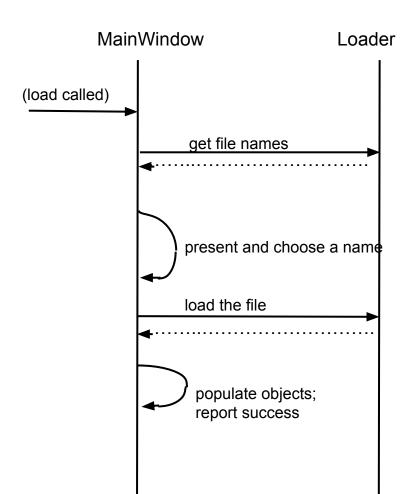
Clicking in the window



Save



Loading



Thank you

