



Model Document

Tim Fanelle, Christopher Hufham, Patrick Kenny, and Jaime Lujan

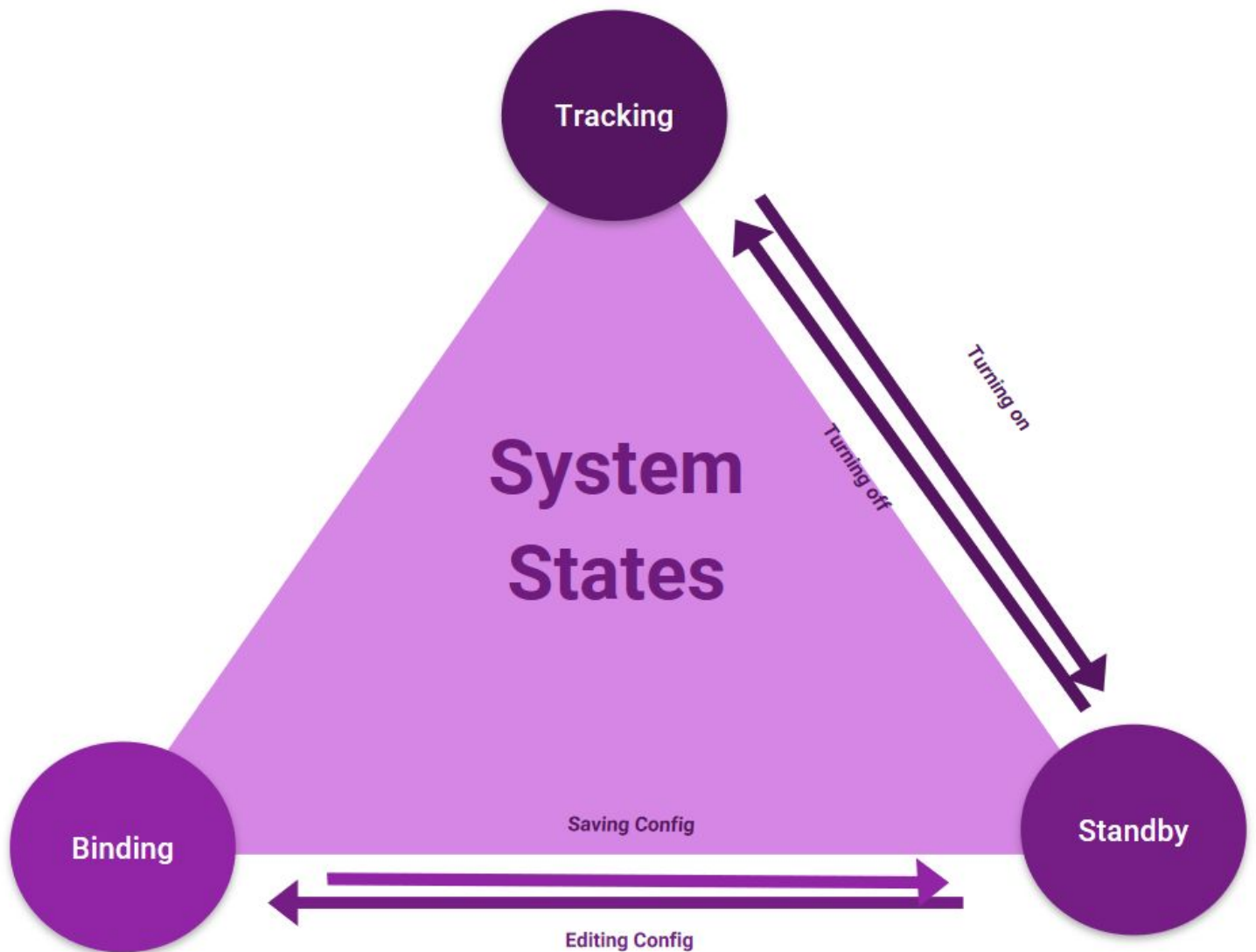
Table of Contents

| | |
|---|-----------|
| Table of Contents | 2 |
| Introduction | 3 |
| System State Diagram | 4 |
| Tracking Process Model | 6 |
| Binding Process Model | 8 |
| Saving and Loading Process Model | 9 |
| Saving Behavioral Model | 10 |
| Use Case Diagram | 11 |
| Validation and Criteria | 12 |

Introduction

This document outlines the behavior of the PyFocals facial tracking controller system. The “PyFocals” program will allow users to pick from a list of facial movements recognized by the software and bind the keys of a keyboard to them. When the user’s face (as captured by a webcam) matches a movement recognized by the software, it will send the key associated by the user to the operating system as a keypress signal. For more general information on the software, please refer to the previous software project plan document.

System State Diagram

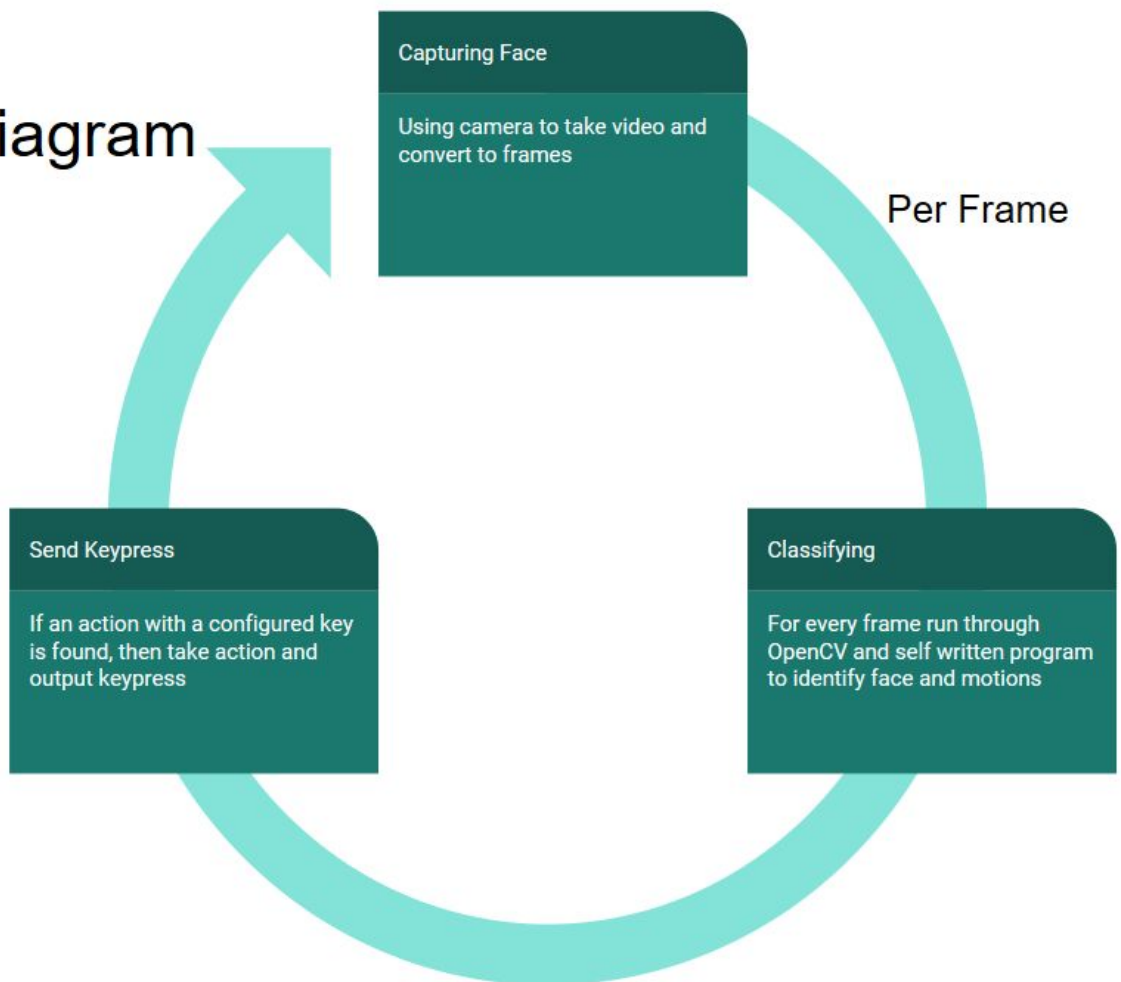


There are three states the program can be in, and it will be in exactly one at any given time. The states are tracking, binding, and standby; which are described below. The functionality in the tracking and binding states are also described in more detail later in the document.

1. **Tracking:** The program is actively reading visual input from webcam and is processing changes that should trigger keypresses. This state contains the facial recognition algorithm, which is the heart of the software. This state be the most resource-intensive, both for our development and for the user's hardware. We expect this state to take the longest time to develop during construction. When the user requests to end tracking, the system moves to the standby state.
2. **Standby:** The software is doing nothing and is waiting for input. This is used as both the default state, and as buffering state between the other two. If the user turns on tracking, the system moves to the tracking state. If the user wants to edit the current keybindings, the system moves to the binding state.
3. **Binding:** This state means the user is currently making changes to the keybindings they have set. The program will not read the user's facial movements at this time. The user may also decide to save or load binding sets to/from their physical storage. When the user is satisfied with their bindings, the system moves to the standby state.

Tracking Process Model

Tracking Diagram

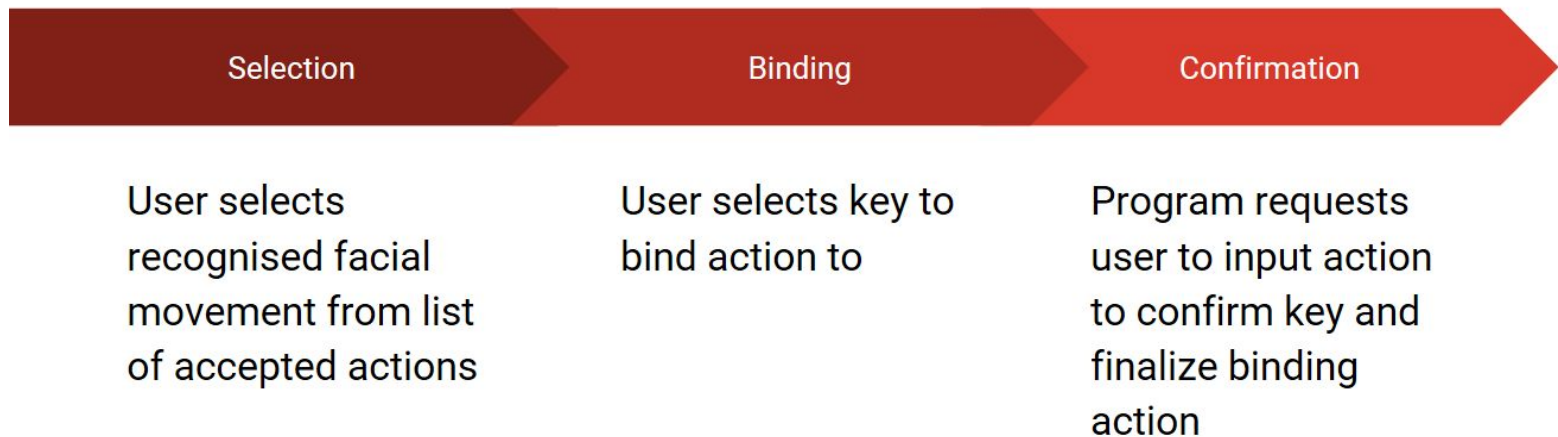


While the system is in the tracking state, the facial tracking algorithm has three steps performed for each frame given by the webcam. First, the software gets the image data of the user's face from the webcam. Then, it classifies the image and checks for recognized motions or facial states. Lastly, it sends the corresponding keypress if a motion is recognized, and the user has a key bound to that motion. The steps are described in greater detail below.

- 1. Capturing Face:** This is the initial capturing of the user's face by the webcam or other video capturing device. The software pulls in image data from the user's webcam if it is available (If not, the software returns to the standby state.) For each image frame retrieved from the webcam, this cycle of three steps executes.
- 2. Classifying:** This step is the analysis and processing of the frame data. The facial tracking algorithm (which is not yet determined) compares the image from the previous step to its list of recognized motions or facial state. An example of a motion would be a nod of the head. An example of a facial state would be having the left eye closed. If it determines the user has input a motion or facial state, then the process continues to the next step. If the classifier determines nothing has happened, the software waits for the next frame to be retrieved from the webcam.
- 3. Send Keypress:** When a motion or facial state is detected, the software checks the list during this step. If the user, while in the binding state, had input a key to be bound to the particular motion detected, then the software sends a signal to the operating system to simulate a keypress of that key.

Binding Process Model

Binding

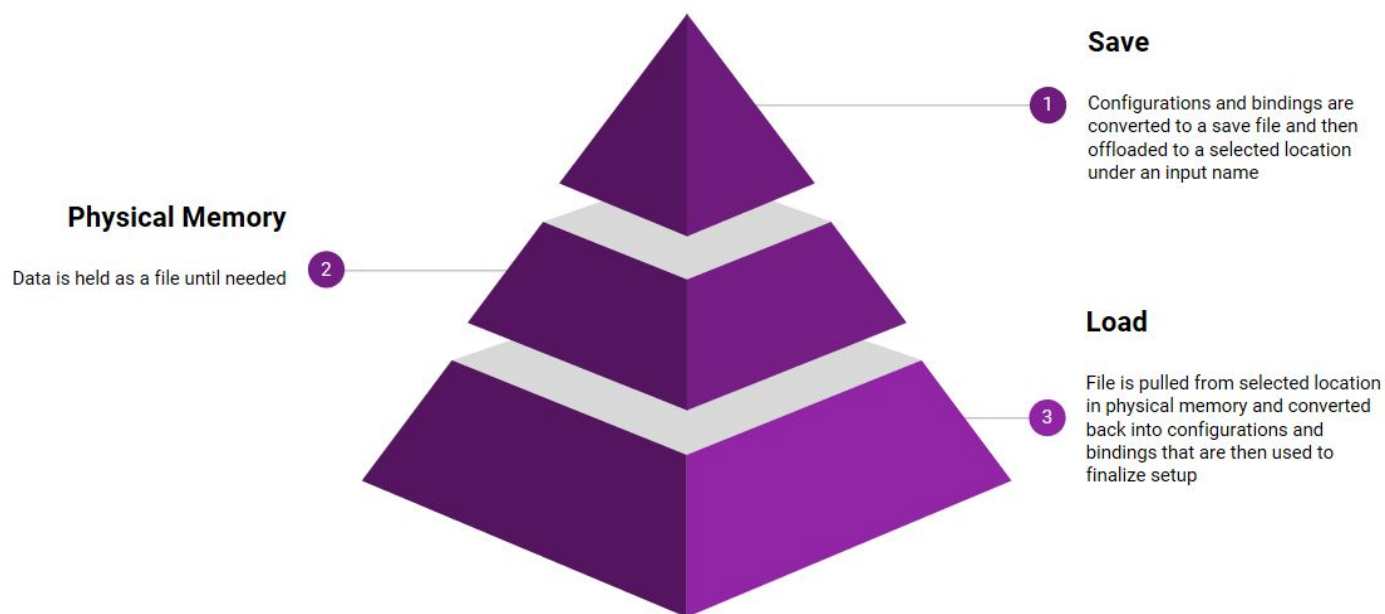


While the system is in the binding state, the user may set up the keybindings to control other software as they see fit. The process of binding a facial movement to a keystroke has three steps:

1. First, the user selects a gesture from the provided table of gestures recognized by the algorithm. As of now, we are uncertain of how many gestures there will be, or what exactly they will be. This will be determined by what the libraries can feasibly allow the program to recognize.
2. After selecting the gesture that the user wishes to bind, they are prompted to hit the key on their keyboard that they wish the program to emulate when the selected gesture is performed.
3. Lastly, the user is prompted to make the selected gesture to confirm the keybinding. When the user gives the gesture, the binding is confirmed and can be used during tracking.

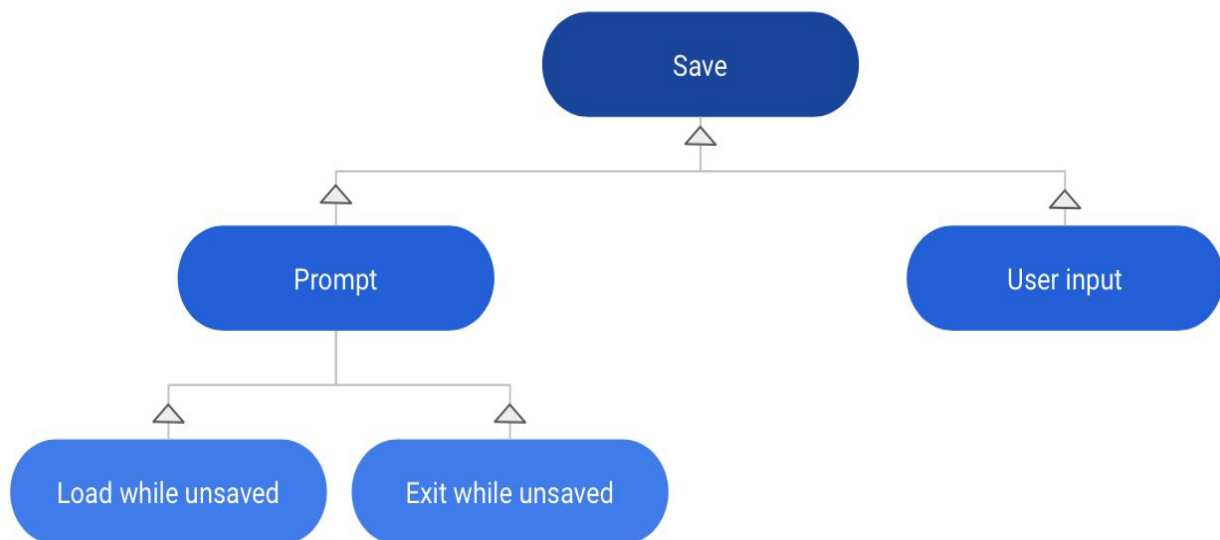
Saving and Loading Process Model

Save/Load



While in the binding state, the user may choose to save or load their current list of bindings. When the user saves, a json file is created on the user's machine in a directory within the program files containing one simple dictionary. The dictionary's keys are the names of the gestures, and the values are the integers representing the keys bound by the user. The file created file can later be loaded back into the software to immediately change the current bindings to whatever the bindings the file specifies.

Saving Behavioral Model

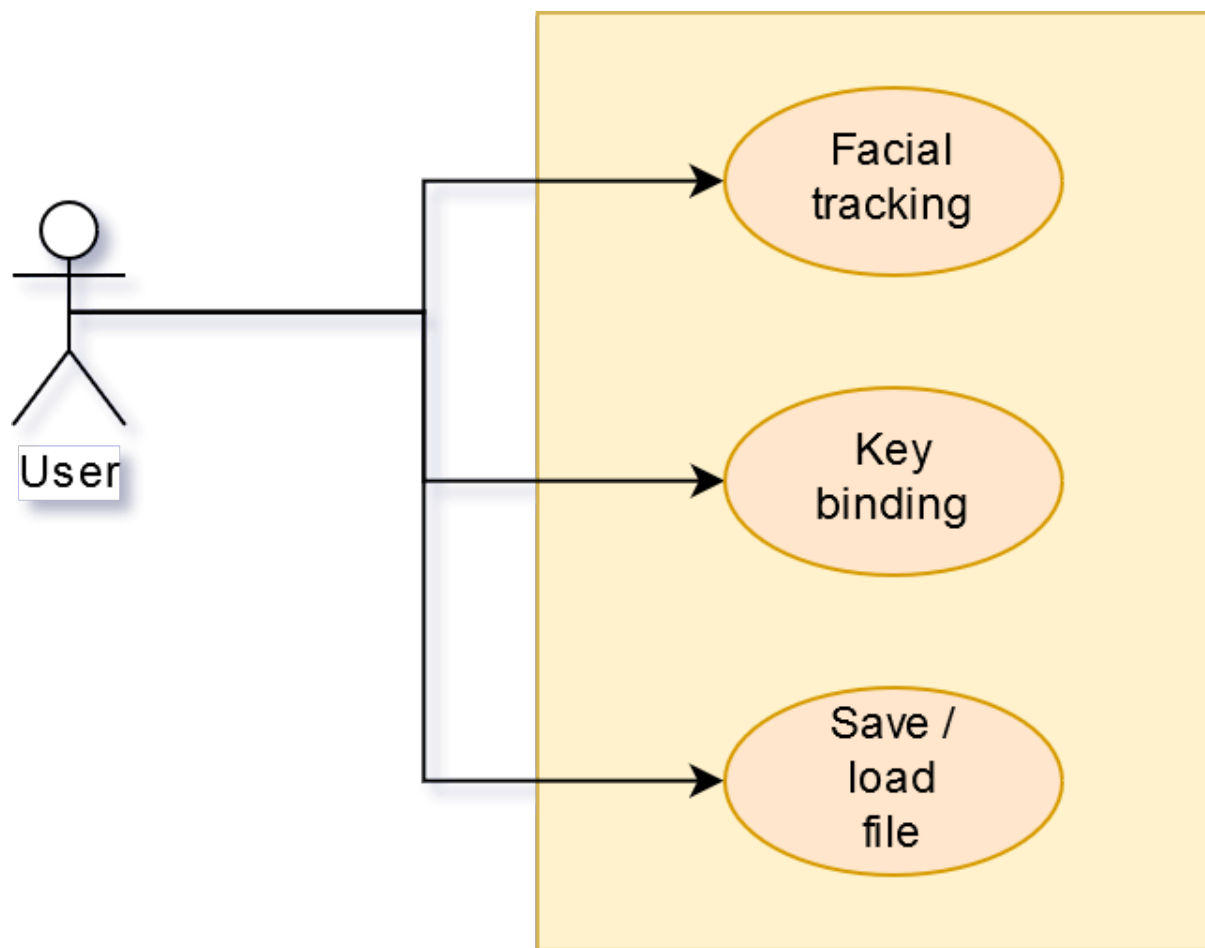


Saving the user's bindings as a file can happen at various points in the software. The most obvious time is when the user specifically requests to save the bindings. When this happens, no confirmation prompt is given because the software knows the user desires the bindings to be saved. There are two other times the software can save bindings, both of which give simple yes/no confirmation prompts.

The first is when the user attempts to load a binding file while they currently have bindings set. The prompt will allow the user a chance to save their current bindings to a file before loading the new ones over it in the program.

The second is when the user attempts to exit the software while they currently have bindings set. The prompt will allow the user to save their bindings before the software closes.

Use Case Diagram



The user has three different sets of functionality they may use while the software is running, all of which described in more detail above. Firstly, the user may use the facial tracking feature, which is the functionality that will be used the majority of the time. Second, the user may set up and adjust key bindings in order to better utilize the facial tracking functionality. Finally, the user may save or load their binding files so that they can be used later. These sets of functionality comprise the entirety of the software from the user's point of view.

Validation and Criteria

- ***Performance bounds:***
 - The performance bounds are that the algorithm will likely only have a maximum of 6-8 possible elements of the face to track while the system is in the tracking state.
 - The user will have limited amount of gestures to bind keys to while in the binding state.
- ***Classes of Tests:***
 - Testing image retrieval from webcam
 - Testing recognition of individual gestures
 - Testing keybinding
 - Testing keypress signals
- ***Expected Software Response:***
 - The software will ideally track 6-8 gestures (or states) of the face.
 - When the user performs gestures that are recognized while in the tracking state, the software will send a keypress signal to the operating system.
 - User can change keybinds while the system is in the binding state.
- ***Special consideration:***
 - Differing qualities and framerates of webcams is an issue that will need to be tested for.