```
// bool InitLightService(uint8_t Priority)
// {
  // set pwm for red and green led strips
// }

// bool PostLightService(ES_Event_t ThisEvent)
// {
  // post to light service
// }

// ES_Event_t RunLightService(ES_Event_t ThisEvent)
// {
      // if CurrentState is LightInactivity
            // if the EventType is LIGHT_START
                  // turn on weather LED
                  // turn on sun (sun is an RGB LED strip but only RG is used)
                  // init timers
                  // change state

    // else if CurrentState is WaitingForPot
            // if EventType is POT_CHANGED
                // update sun brightness

            // else if EventType is CORRECT_POT_VAL_DETECTED
                  // udpate sun brightness
                  // init pot debounce timer
                  // change state

            // else if EventType is ES_TIMEOUT and EventParam is POT_TIMER
                  // light sun red (turn off green LEDs on the LED strip)
                  // SetSunBrightness(PulseWidth, PulseWidth);
                  // init light feedback timer
                  // subtract from health bar
                  // change state

            // else if EventType is ES_TIMEOUT and EventParam is LIGHT_TIMER
                  // turn off sun (turn off LED strip)
                  // turn off weather LED
                  // Post to Game Service
                  // change state

      // else if CurrentState is Debounce
            // get current pot val
            // if EventType is ES_TIMEOUT and EventParam is LIGHT_TIMER
                  // turn off sun (turn off LED strip)
                  // turn off weather LED
                  // Post to Game Service
                  // change state
            // else if EventType is ES_TIMEOUT and EventParam is POT_TIMER
                  // light sun red (turn off green LEDs on the LED strip)
                  // SetSunBrightness(PulseWidth, PulseWidth);
                  // init light feedback timer
                  // subtract from health bar
                  // change state

            // else if EventType is ES_TIMEOUT and EventParam is
POT_DEBOUNCE_TIMER
                  // if pot val is correct
                        // light sun green (turn off red LEDs on the LED strip)
                        // init light feedback timer
                        // add to health bar
                        // change state

            // else
```

```
                        // update sun brightness
                        // change state

        // else if EventType is ES_TIMEOUT and EventParam is POT_CHANGED
            // update sun brightness

    // else if CurrentState is Feedback

            // if EventType is ES_TIMEOUT and EventParam is LIGHT_TIMER
                  // turn off sun (turn off LED strip)
                  // turn off weather LED
                  // Post to Game Service
                  // change state

            // else if EventType is ES_TIMEOUT and EventParam is LIGHT_FB_TIMER
                  // turn on weather LED
                  // turn on sun (turn on LED strip)
                  // Init POT_TIMER
                  // Change state

  // return ReturnEvent;
// }

/*****************************************************************************
 public functions
 *****************************************************************************/

// void TurnOffSun(void){
  // set brightness of red and green led strips to zero
// }

// bool Check4PotChange(void) {
      // create static variable for previous pot val
      // get current pot val

      // if pot value has changed
            // create an event to post

            // if pot value is the desired pot val
                  // set event type to CORRECT_POT_VAL_DETECTED

            // else
                  // set event type to CORRECT_POT_VAL_DETECTED

      // post event to light service
      // update previous pot val

  // return true if event detected
// }

/*****************************************************************************
 private functions
 *****************************************************************************/
// static bool IsPotCorrect(uint32_t CurrentPotVal) {
      // calculate whether or not the current pot val is within POT_TOLERANCE
away from the desired pot value
      // return true if so, and return false if not
// }

// static uint32_t NewWeatherLED(void) {
  // generate a new weather LED index (0, 1, or 2), cannot be the same as the
previous weather LED index
  // post LIGHT_CMD to LEDWriteService (parameter is a 8-bit number with the bit
high corresponding to the weather LED index)
```

```
  // return desired pot val (analog value corresponding to the weather LED
index)
// }

// static void SetSunBrightness(uint16_t RedPulseWidth, uint16_t
GreenPulseWidth){
    // set the pulsewidth for the red LED strip
    // set the pulsewidth for the green LED strip
// }

// static uint32_t GetPotVal(void){
      // read from the analog inputs
      // select the element corresponding to the potentiometer value
      // return the potentiometer value
// }

// static uint16_t CalcPulseWidth(uint32_t PotVal){
      // based on range of possible pot values and period of the LED strips,
calculate a pulse width from the pot value
// }

/*------------------------------ Footnotes -------------------------------*/
/*------------------------------ End of file -------------------------------*/
```