

Chess clock

Homework assignment - System modeling (FA048)

2023. március 28.

1. Overview

The student's task is to design the controller of a chess clock based on the specification below using YAKINDU.





1.1. The chess clock

The task of the chess clock is to specify the time to think for the participants. Both players (White and Black) have a given amount of time to think, which the chess clock displays. The time of the current player constantly decreases. If the time runs out, the player loses and the clock beeps to indicate the termination of the game due to timeout. If the current player moves in time, the player should press the appropriate button. As a result the decreasing stops and (depending on the game's setup) the time increases by a certain amount of reward time and the other player comes instantly. The player can use the remaining time (increased with reward time) to think in the next turn.

The expected behavior of the chess clock is discussed in section 2.





1.2. Control panel

The control panel of our chess clock consists of the following elements.

- Main text display (**Display text**), where the chess clock provides general information.
- Clock displaying the time of the White player (**White display**).
- Clock displaying the time of the Black player (**Black display**).
- Beeper that may beep when timeout or other events occur (**Beeper**).
- Buttons  and  for setting up and starting/stopping the game (see section 2).
- The button  pressed after the White player's steps.
- The button  pressed after the Black player's steps.

1.3. Yakindu binding


We start building the chess clock's model in YAKINDU state modeling tool by defining the system's interfaces to the outside world. To represent them in YAKINDU we use interface events and variables (these are also used by the GUI, that will be provided later for support). Thus, our model can react to a keypress, and we can display the values of the variables on the screen. The interfaces of the system are listed below. The homework must be prepared accordingly.

- **Display.text**: interface variable representing the main display. Its text value can be read on the screen.
- **Display.whiteDisplay**: interface variable representing the White player's clock. Its non-negative integer value determines the number of seconds, which will be displayed in minutes:seconds format. If we assign a negative integer value to the variable (eg. -1) then the clock will be turned off (inactive).
- **Display.blackDisplay**: interface variable representing the Black player's clock. Similarly to the above, its non-negative integer value determines the number of seconds, which will be displayed in minutes:seconds format. If we assign a negative integer value to the variable (eg. -1) then the clock will be turned off (inactive).
- **Beeper.beep**: output event, that beeps the beeper when triggered.
- **Buttons.modeButton**: input event representing when  is pressed.
- **Buttons.startButton**: input event representing when  is pressed.
- **Buttons.whiteButton**: input event representing when  is pressed.
- **Buttons.blackButton**: input event representing when  is pressed.

Of course, further state variables can be defined for inner usage (that is, separated from the interfaces). There is a placeholder for this purpose in the YAKINDU model-stub issued 5th week.

2. Behavior to be implemented

2.1. Switching between setup modes

Initially `White begins` is displayed on the screen. Before the game begins certain setup options are available; pressing  cyclically switches between the following contents.

- `White initial time`,
- `Black initial time`,

- White increment time,
- Black increment time
- White begins again, and so on.

Section 2.4 discusses the detailed operation of these setup menus. When switching between setup menus by pressing **[MODE]** the chess clock beeps. The **[MODE]** button only has a function here, at the setup part. During the game there will be no use for it.

The button **[START/RESET]** will start the game in any of the listed cases (setup menus).

2.2. The course of the game

During the game the clock of the White player will always show the White player's remaining time to think, and the clock of the Black player always shows the Black player's remaining time to think. The displayed text on the main display is either **White moves** or **Black moves**, depending on which player's turn it is.

When the game begins both clocks are set to the initial time to think. The initial time to think is 120 seconds for both players by default, but this value can be changed for each player individually (see section 2.4).

The first turn is the White player's turn by default but the chess clock can be set otherwise.



The time of the current player is constantly decreasing (one second per second, as obvious) while the other player's time remains unchanged. If the current player presses the appropriate button (which is **[♔/+]** for the White player and **[♚/-]** for the Black player; the other player's button is ineffective) before the time runs out, then the current player's time increases and the other player's turn begins (his time starts decreasing). The amount of reward time given is 0 seconds for each player by default, but this value can be changed for each player individually (see section 2.4).

However, the increased time can't exceed the maximum time enabled. If the sum of the remaining time and the reward time exceeds the maximum time to think, then the player's time will be set to the maximum time. The value of the maximum time is 1200 seconds for both players.



2.3. Finishing and aborting the game

In case the players finish the game (by checkmate, stalemate, surrender or draw by agreement) before the time runs out, they indicate it, by pressing the **[START/RESET]** button. As a result **White begins** is displayed on the screen again, and the players can press **[START/RESET]** to start a new game or **[MODE]** to access the settings.



In case the current player's remaining time is dangerously low, that is, 5 or less, then the chess clock beeps to warn the player to move quickly.

In case the current player's time runs out, the clock beeps to inform the players that the game is over and the current player lost. Then, if the White player lost `White flag fallen` can be seen on the screen, otherwise `Black flag fallen` is displayed. The players' clocks continue to display their remaining time (as obvious, the losing player's time is zero). The chess clock doesn't react to the players' buttons (, ) or `MODE`, but `START/RESET` resets the clock (`White begins` appears again, etc. as discussed previously).

2.4. Setting the parameters

When the user chose `White initial time` menu, White player's time can be set. In this case the Black player's clock is inactive while the White player's initial time to think is displayed on the White player's clock.  decreases the time by 5 seconds while  increases it by 5 seconds. If the value is less than 60 seconds or more than 180 seconds, the change will be unsuccessful. In this case the chess clock beeps instead of modifying the value. Similarly whilst displaying `Black initial time` the White player's clock is inactive, while the Black player's clock can be used to set the Black player's time similarly to the above.

When `White increment time` menu is chosen the White player's reward time can be displayed and set, and when `Black increment time` menu is chosen the Black player's reward time can be displayed and set. Displaying and setting the time works similarly to the above. The minimum value of reward time is 0 seconds, the maximum is 60 seconds, and the value can be increased or decreased by 5 seconds per keypress.

When the user sees `White begins` (or `Black begins`) after setting the first player) the first player can be set.  sets the White player and  the black player as the first player. If the White player is chosen `White begins`, otherwise `Black begins` is displayed. The clocks are inactive both cases.

In each of these menus the modified value of a parameter is preserved until it is modified again, regardless of the number of games played and the other settings.

3. Released Test Cases

3.0.1. base1

Basic test: Pressing each button.

 →  → `MODE` → `START/RESET`

3.0.2. base2

Basic test: Waiting for 3 sec.

+3s

3.0.3. optionCycle

By pushing the MOD button in the menu each options are available, and are set to the default values.

Check #1> White begins on the main display



Check #2> *BEEP* is expected

Check #3> White initial time on the main display

Check #4> 02:00 on the WHITE clock



Check #5> *BEEP* is expected

Check #6> Black initial time on the main display

Check #7> 02:00 on the BLACK clock



Check #8> *BEEP* is expected

Check #9> White increment time on the main display

Check #10> 00:00 on the WHITE clock



Check #11> *BEEP* is expected

Check #12> Black increment time on the main display

Check #13> 00:00 on the BLACK clock



Check #14> *BEEP* is expected

Check #15> White begins on the main display

3.0.4. checkSetStartTimeForWhite

Checking the upper and lower bounds of the target option for the White player. First, the value is increased from default to maximal, plus one more time to check if it stops. Then, it decreased to minimal and checked again it it stops.











Check #1> 02:00 on the WHITE clock



Check #2> 02:05 on the WHITE clock



Check #3> 02:10 on the WHITE clock

 →  →  →  →  →  →  → 

Check #4> 02:50 on the WHITE clock



Check #5> 02:55 on the WHITE clock



Check #6> 03:00 on the WHITE clock



Check #7> *BEEP* is expected

Check #8> 03:00 on the WHITE clock

Check #9> 02:55 on the WHITE clock

Check #10> 02:50 on the WHITE clock

Check #11> 01:10 on the WHITE clock

Check #12> 01:05 on the WHITE clock

Check #13> 01:00 on the WHITE clock

Check #14> 01:00 on the WHITE clock

Check #15> *BEEP* is expected

Checking the upper and lower bounds of the target option for the Black player. First, the value is increased from default to maximal, plus one more time to check if it stops. Then, it decreased to minimal and checked again if it stops.

Check #1> 02:00 on the BLACK clock

Check #2> 02:05 on the BLACK clock

Check #3> 02:10 on the BLACK clock

Check #4> 02:50 on the BLACK clock

Check #5> 02:55 on the BLACK clock

Check #6> 03:00 on the BLACK clock

Check #7> *BEEP* is expected

Check #8> 03:00 on the BLACK clock

Check #9> 02:55 on the BLACK clock

Check #10> 02:50 on the BLACK clock

Check #11> 01:10 on the BLACK clock



Check #12> 01:05 on the BLACK clock



Check #13> 01:00 on the BLACK clock



Check #14> 01:00 on the BLACK clock

Check #15> *BEEP* is expected

3.0.6. checkSetBonusTimeForWhite

Checking the upper and lower bounds of the target option for the White player. First, the value is increased from default to maximal, plus one more time to check if it stops. Then, it decreased to minimal and checked again if it stops.

MODE → MODE → MODE

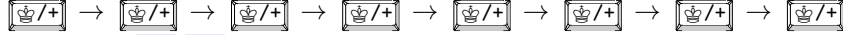
Check #1> 00:00 on the WHITE clock



Check #2> 00:05 on the WHITE clock



Check #3> 00:10 on the WHITE clock



Check #4> 00:50 on the WHITE clock



Check #5> 00:55 on the WHITE clock



Check #6> 01:00 on the WHITE clock



Check #7> *BEEP* is expected

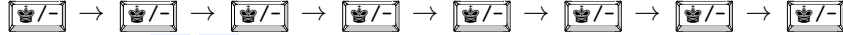
Check #8> 01:00 on the WHITE clock



Check #9> 00:55 on the WHITE clock



Check #10> 00:50 on the WHITE clock



Check #11> 00:10 on the WHITE clock



Check #12> 00:05 on the WHITE clock



Check #13> 00:00 on the WHITE clock



Check #14> 00:00 on the WHITE clock

Check #15> *BEEP* is expected

3.0.7. checkSetBonusTimeForBlack

Checking the upper and lower bounds of the target option for the Black player.
First, the value is increased from default to maximal, plus one more time to check if it stops. Then, it decreased to minimal and checked again it it stops.

→ → →

Check #1> 00:00 on the BLACK clock

Check #2> 00:05 on the BLACK clock

Check #3> 00:10 on the BLACK clock

→ → → → → → →

Check #4> 00:50 on the BLACK clock

Check #5> 00:55 on the BLACK clock

Check #6> 01:00 on the BLACK clock

Check #7> *BEEP* is expected

Check #8> 01:00 on the BLACK clock

Check #9> 00:55 on the BLACK clock

Check #10> 00:50 on the BLACK clock

→ → → → → → →

Check #11> 00:10 on the BLACK clock

Check #12> 00:05 on the BLACK clock

Check #13> 00:00 on the BLACK clock

Check #14> 00:00 on the BLACK clock

Check #15> *BEEP* is expected

3.0.8. checkSetStartPlayer

Checking whether the initial player can be set.

Check #1> White begins on the main display

Check #2> Black begins on the main display

Check #3> Black begins on the main display

→ → → →

Check #4> Black begins on the main display



Check #5> White begins on the main display



Check #6> White begins on the main display

3.0.9. checkEffectSetStartPlayer

Checks the result of changing the initial player.



→ [START/RESET]

Check #1> Black moves on the main display

3.0.10. checkEffectSetStartTimeForWhite

Checks the result of changing the initial time for player White.



Check #1> White initial time on the main display

Check #2> 02:00 on the WHITE clock



Check #3> White initial time on the main display

Check #4> 02:05 on the WHITE clock



Check #5> White moves on the main display

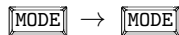
Check #6> 02:05 on the WHITE clock



Check #7> 02:02 on the WHITE clock

3.0.11. checkEffectSetStartTimeForBlack

Checks the result of changing the initial time for player Black.



→ [MODE]

Check #1> Black initial time on the main display

Check #2> 02:00 on the BLACK clock



Check #3> Black initial time on the main display

Check #4> 02:05 on the BLACK clock



→ [Crown icon with a plus sign]

Check #5> *BEEP* is NOT expected

Check #6> Black moves on the main display

Check #7> 02:05 on the BLACK clock



Check #8> 02:02 on the BLACK clock

3.0.12. checkEffectSetBonusTimeForWhite

Checks the result of changing the initial time for player White.



→ [MODE] → [MODE]

Check #1> White increment time on the main display
 Check #2> 00:00 on the WHITE clock
 [KING/+]
 Check #3> White increment time on the main display
 Check #4> 00:05 on the WHITE clock
 [START/RESET]
 Check #5> White moves on the main display
 Check #6> 02:00 on the WHITE clock
 [KING/+]
 Check #7> *BEEP* is NOT expected
 Check #8> 02:05 on the WHITE clock

3.0.13. checkEffectSetBonusTimeForBlack

Checks the result of changing the initial time for player Black.

[MODE] → [MODE] → [MODE] → [MODE]
 Check #1> Black increment time on the main display
 Check #2> 00:00 on the BLACK clock
 [KING/+]
 Check #3> Black increment time on the main display
 Check #4> 00:05 on the BLACK clock
 [START/RESET] → [KING/+]
 Check #5> *BEEP* is NOT expected
 Check #6> Black moves on the main display
 Check #7> 02:00 on the BLACK clock
 [KING/-]
 Check #8> *BEEP* is NOT expected
 Check #9> 02:05 on the BLACK clock

3.0.14. checkResetInOptions

Checking the effect of pushing the START/RESET button in the options.

[MODE]
 Check #1> White initial time on the main display
 Check #2> 02:00 on the WHITE clock
 [KING/+]
 Check #3> White initial time on the main display
 Check #4> 02:05 on the WHITE clock
 [START/RESET]
 Check #5> White moves on the main display

3.0.15. checkStartInGame

Checking the START/RESET button in the game

Check #1> White begins on the main display

START/RESET

Check #2> White moves on the main display

START/RESET

Check #3> White begins on the main display