

Racket GUI

Inhaltsverzeichnis

1 Racket GUI Komponenten	3
2 Code	5

1 Racket GUI Komponenten

Folgende Komponenten wurden erfolgreich eingebaut:

1. racket/gui
2. bitmap
3. frame
4. menu-bar
5. menu
6. menu-item
7. checkable-menu-item
8. choice
9. message
10. frame show
11. frame create-status-line
12. frame get-menu-bar
13. frame set-icon bitmap
14. frame set-status-text
15. frame has-status-line?
16. button
17. canvas
18. canvas get-dc
19. check-box
20. check-box set-value
21. mouse-event

22. timer
23. timer start
24. popup-menu
25. radio-box
26. scroll-event
27. text-fiel
28. combo-field

2 Code

```
#lang racket
(require racket/gui)
(define bitmap
  (read-bitmap "C:/Users/User/Desktop/door_1.gif"))

(define frame
  (new frame%
    [label "testframe"]
    [width 300]
    [height 300]
  )
)

(define menu-bar
  (new menu-bar%
    [parent frame]
  )
)

(define menu
  (new menu%
    [label "menu"]
    [parent menu-bar]
  )
)

(define menu-item
  (new menu-item%
    [label "menu-item"]
    [parent menu]
    [shortcut #\A]
    [shortcut-prefix (list 'ctl)]
    [callback
      (lambda(menu-item event)
        (send dc set-text-foreground "red")
        (send dc draw-text "test" 5 5)
        (send dc draw-line 3 7 9 22)
      )
    ]
  )
)

(define checkable-menu-item
  (new checkable-menu-item%
    [label "checkable"]
    [parent menu]
    [shortcut #\V]
    [shortcut-prefix (list 'ctl)]
    [checked #t]
    [help-string "helpstring"]
    [callback
      (lambda(checkable event)
        (send frame set-label "text")
      )
    ]
  )
)

(define choice
```

```

(new choice%
 [label "choice"]
 [parent frame]
 [choices (list "1" "1" "1")]
 [callback
  (lambda(choice event)
   (send button set-label
    (format "~a"
     (send choice get-selection)
    )
   )
  )
 ]
)

(define msg
 (new message%
  [parent frame]
  [label "text"]
  [auto-resize #t]
 )
)

(send msg set-label "text neu")
(send msg auto-resize)

(define test
 (lambda(x)
  (display (+ 1 x))
 )
)

(send frame show #t)
(send frame create-status-line)
(send frame get-menu-bar)
(send frame set-icon bitmap)
(send frame set-status-text "status-text")
(send frame has-status-line?)

(define button
 (new button%
  [label "button"]
  [parent frame]
  [callback
   (lambda(button event)
    (send button set-label "settext")
   )
  ]
 )
)

(define canvas
 (new canvas%
  [parent frame]
 )
)

(define dc(send canvas get-dc))

(define check1 #f)

```

```

(define check-box
  (new check-box%
    [label "check"]
    [parent frame]
    [callback
      (lambda(check event)
        (set! check1
          (send check-box get-value)
        )
        (display check1)
      )
    ]
  )
)

(send check-box set-value #t)

;(define mouse-event
;  (new mouse-event%
;    [event-type 'enter]
;    [x 5]
;    [y 0]
;  )
;)

;(define timer
;  (new timer%
;    [notify-callback
;      (lambda()
;        (display
;          (send mouse-event button-down? 'left)
;        )
;      )
;    ]
;  )
;)

;(send timer start 100 #f)
;(begin
;  (new popup-menu%
;    [title "poptxt"]
;    [popdown-callback
;      (lambda(pop event)
;        (test 5)
;      )
;    ]
;  )
;)

(define radio-box
  (new radio-box%
    [label "radio"]
    [choices '("1" "2" "3")]
    [callback
      (lambda(radio event)
        (display
          (send radio-box get-selection)
        )
      )
    ]
  )
)
[parent frame]

```

```
)  
)  
  
(define scroll-event  
  (new scroll-event%  
    [event-type 'top]  
    [position 200]  
    [direction 'vertical]  
  )  
)  
  
(define text-fiel  
  (new text-field%  
    [label "text-field"]  
    [parent frame]  
  )  
)  
  
(define combo-field  
  (new combo-field%  
    [label "radio"]  
    [choices '("1" "2" "3")]  
    [callback  
      (lambda(cfield event)  
        (display  
          (send combo-field get-value)  
        )  
      )  
    ]  
  [parent frame]  
  )  
)  
)
```