

## **Proposal for FKISS3**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> Proposal for FKISS3		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 19, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Proposal for FKISS3</b>	<b>1</b>
1.1	Proposal for FKISS3 . . . . .	1
1.2	Labels . . . . .	1
1.3	Exception handling . . . . .	1
1.4	Label commands . . . . .	1
1.5	Variables . . . . .	2
1.6	Conditional commands . . . . .	2
1.7	Notes . . . . .	3

## Chapter 1

# Proposal for FKISS3

### 1.1 Proposal for FKISS3

Proposal for FKISS3. Modified 3-21-98

Authors

Chad Randall -- Author of PlayFKISS 95/NT

Nick Roberts -- Author of PlayKiss Acorn

Labels

Exceptions

Label commands

Variables

Conditional commands

Notes

### 1.2 Labels

Labels, new label event. Similar to an Alarm. Possible 512 labels, numbered from 0 to 511.

```
;%label(1)
```

### 1.3 Exception handling

Exception handling event. Executed when numeric operations go bad. Such as divide by zero, or variables out of range.

```
;%overflow()
```

### 1.4 Label commands

---

Label commands. These access the label events. goto does not continue processing of the calling event once the label event is finished. gosub does. gotorandom and gosub random both take the first parameter as a percentage. The first label listed will be called this percent of the time. In the following examples, label 1 is called 25% of the time, label 2 is called 75%.

```
;@goto(1)
;@gosub(1)
;@gotorandom(25,1,2)
;@gosubrandom(25,1,2)
;@exitevent() ; Works any where to stop execution of the current event and
               return control to the calling one.
```

## 1.5 Variables

Variables. There are 26 variables, listed as A through Z. These are set to 0 during the initial load. New commands to access the variables have been added. Case is not important. All variables are 32-bit signed integers.

```
;@let(A, 1) ; Assigns the integer value of 1 to the variable A.
;@let(A, B) ; Assigns the value stored in "B" to "A".
;@add(A, B, C) ; A = B + C ; Adds "B" and "C" together and stores the result in "A ←
"
;@sub(A, B, C) ; A = B - C
;@mul(A, B, C)
;@div(A, B, C) ; Integer division. Use mod() to retrieve the remainder.
;@mod(A, B, C) ; Returns the modulus of "B" and "C" and stores the result in "A".
;@random(A, 1, 10); random number from 1 to 10
;@letobjectx(A, #1) ; A = x coordinate of object #1
;@letobjecty(A, #1)
;@letfix(A, #1) ; A = fix value of object
;@letmapped(A, "cel.cel") ; A = 1 if mapped, 0 if not.
;@letset(A) ; A = current set number
;@letpal(A)
;@letmousex(A) ; A = real-time mouse x position
;@letmousey(A) ; A = real-time mouse y position
;@letcatch(A) ; A = object number of current caught (caught) object
;@letcollide(A, "cel1.cel", "cel2.cel") ; A = 1 if cel1 touches cel2, 0 if not
;@letinside(A, #0, #1) ; A = 1 if objects touch, 0 if not
;@lettransparent(A, "cel1.cel")
```

## 1.6 Conditional commands

if, else, endif commands. This begins to build a real logic system. These commands actually modify how an event is processed. Right now, no nesting is allowed. Only a simple if, else, endif structure. Use gosubs for more complex logic (spaghetti).

```
;@ifequal(a, b)
;@ifequal(a, 1)
;@ifnotequal(a, b)
;@ifnotequal(a, 1) ;
```

```
;@ifgreaterthan(a, b) ; runs if A > B
;@ifgreaterthan(a, 1)
;@iflessthan(a, b) ; runs if A < B
;@iflessthan(a, 1)
;@else() ; Parenthesis are required.
;@endif() ; Parenthesis are required.
```

## 1.7 Notes

Some notes.

1. Variables can be used with any command where an integer is expected.  
The following code is legal:

```
;@movebyx(#2, A, B)
;@changeset(C)
;@timer(D, E)
```

The following is NOT legal:

```
;@alarm(D) ; WRONG! Events must use hard-coded values.
;@unmap(F) ; WRONG! Objects must be declared with static values and preceeded with ←
"#".
```

2. The additional coding will most likely BREAK non-FKiSS3 viewers. Authors are STRONGLY encouraged to included the following code:

```
#100.32760 "warning.cel" ;A message box saying "This set requires FKISS3!"
;@EventHandler()
;@alarm(63) ; Dummy wrapper for older viewers.
;@version(4)
;@unmap(#100)
;@release(#100) ; In case the user still wants to try.
;@unmap(#100)
```

There would be a thorough testing CNF available to all programmers. This CNF would test various fuzzy areas of these specs.

---