

[Version number, author, save date, and task days total are fields and can be updated with F9]

## Player Input System Design Document, v5

Greg Sabatini

October 8, 1999

### **Introduction**

The player input system interacts between all input devices (mouse, keyboard, and any device to be added in the future.) Information such as keys/mouse buttons pressed get queried from this system by the scripting system.

### **Requirements**

This system should be polled once per frame to find the current status of the mouse and keys. After the poll, any information queried will be that gathered by the poll. It will be used by the script system to check screen position/hotspot location. It will also be used to draw the mouse cursor.

### **Structures/Classes**

```
struct framedata
{
    int X; // X Position
    int Y; // Y Position
    int button; // Bitfield variable for mouse button
    char* keys; // Bitfield array for keypresses
}
```

CInput class – Wraps all devices and stores current data and last frame data (cross-platform)

```
{
    CKeyboard *pKeyboard;
    CMouse *pMouse;
    struct framedata Last;
    struct framedata Curr;
}
```

CKeyboard class – Wraps keyboard input for CInput class (platform specific)

CMouse class – Wraps mouse input for CInput class (platform specific)

### ***Functions/Methods***

CInput::CInput – constructor creates CKeyboard & CMouse.

CInput::~CInput – destructor destroys CKeyboard & CMouse.

CInput::Poll – copies curr frame data to last frame and queries CKeyboard & CMouse for new data.

CInput::GetData – functions to query data from curr frame data

### ***Diagrams***

Frame start

CInput::Poll

CInput::GetData

Frame end

### Schedule Task List

System Tasks	Duration	Dependent
Design CInput class	0.5 Days	Design Document
Win32 Design CKeyboard & CMouse class	0.5 Days	CInput class design
Mac Design CKeyboard & CMouse class	0.5 Days	CInput class design
Code CInput class	0.5 Days	CKeyboard & CMouse design
Code Win32 CKeyboard class	0.5 Days	CInput class coded
Code Win32 CMouse class	0.5 Days	CInput class coded
Test & Revise Win32 CKeyboard & CMouse classes	1 Day	Win32 CKeyboard & CMouse classes coded
Code Mac CKeyboard class	1 Day	CInput class coded
Code Mac CMouse class	1 Day	CInput class coded
Test & Revise Mac CKeyboard & CMouse classes	1 Day	Mac CKeyboard & CMouse classes coded
Revise CInput (cross platform)	0.5 Days	Win32 & Mac Testing and Revision of CKeyboard and CMouse
Rework #1 Input system	1 Day	As Needed
Test & Revise Input System Rework #1	1 Day	Input System Reworked #1
Rework #2 Input system	1 Day	As Needed
Test & Revise Input System Rework #2	1 Day	Input System Reworked #2
Total	11.5 Days	

### Memory

Bitfields reduce the amount of memory needed by this system. Approximately 60 bytes is required: 17 bytes for key presses, 4 for X, 4 for Y, 4 for mouse \* 2 sets of data.

There may be memory required by platform specific API's.

No storage is used for this system.

### ***Risk Assessment***

Should be a straight forward system. All games must use keyboard and mouse. Mac remains uncharted territory, which is why the coding time is double that of Windows.

### ***QA & Test***

All keys should work, mouse should return values for button presses and location. Popup could be created that shows all input values to verify correct functionality. Once this system is verified, keypresses and mouse clicks not registering should not be the fault of this system. This system will be continuously logged, however, which should allow problems with it to be easily detected.