Tutorial: Action Transplant by Cake

This is a tutorial about action transplant, that is, make some character able to do the action of another. We'll take transplanting **Raye**'s Thunder Strike (G↑J, where G for guard and J for jump) to **Lucas** as an example and describe the mechanism and steps of action transplanting. In the following we'll call Raye "R" and Lucas "L". This tutorial is divided into 2 parts:
1. Relationship between `"actionGroup"`, `"action"` and `"frame"`; 2. Steps of action transplant.

Following is the body of the tutorial.

1 Relationship between "actionGroup", "action" & "frame"

These 3 variables are parallel items of a character's Spt file. The key part that connects them is as following. Items in `{}` are sub-items of items before the same `{}`.

```
# contents: "actionGroup"
"0": {
    "actionIndex": {
        "0": {
            "0": 0.0,
            ...,
            "7": 3.0,
            ...
        },
        ...
    },
},
"1": {...},
"2": {...},
"3": {...}
```

```
# contents: "action"
"0": {
    "frameIndex": 1.0,
    "a_keyTgr": {
        "0": {
            "agi": 0.0,
            "ati": 3.0,
            "ai": 1.0,
            "k": "gra",
            "kr": "gla",
            "kl": 3.0,
            "atf": true...
        }, ...
    }
}
```

```
# contents: "frame"
"0": {
    "index": 0.0,
    "last": false,
    ...
},
"1": {
    "index": 1.0,
    "last": false,
    ...
},
"2": {
    "index": 2.0,
    "last": true, ...
}
```

In the above code chunks, `"agi"` refers to the number (like `"0"`, `"3"`) in `"actionGroup"`. `"0"` (`"1"`, `"2"`, `"3"` is not listed here), which is a direct sub-item of `"actionIndex"`, is referred to by `"ati"` in `"action"`'s code chunk. And `"ai"` refers to items like `"7"` in the first code chunk. And `"7"`'s value `3.0` refers to action's index, which are items like the first `"0"` in `"action"`'s code chunk.

Every action in HF starts from some "pre-actions". For example, Lucas's uppercut can start from stand, run and roll.

In the above code chunk of `"action"`, the first `"0"` stands for the first (`"1"` for 2nd, `"2"` for 3rd, …) action in action list. It's also the action's index, but different from `"ai"`, whose value refers to action's index.

(pre-action) `"action"` > `"a_keyTgr"` $\xrightarrow{\text{"agi", "ati", "ai"}}$ `"action"` $\xrightarrow{\text{"frameIndex"}}$ `"frame"`

"a_keyTgr" means allow-Key-Trigger, "kl" is key-length. In "k" (key) and "kr" (key-reverse, which is left, only active when "atf" -auto-turn-face is "true"), "r" is right, "l" is left, "u" is up, "d" is down, "a" is attack, "j" is jump, "g" is guard.

As the above sketch map shows, when the corresponding key is pressed, the character moves from the pre-action to the next action, which is determined by "agi" (action-group-index), "ati" (action-tree-index?) and "ai" (action-index). An action contains several frames, and each action is connected to the first of its frames by its sub-item "frameIndex". "last": true means current frame is the last frame of its action. Frames will automatically turn into next frame when "last"'s value is false.

2 Steps of action transplant

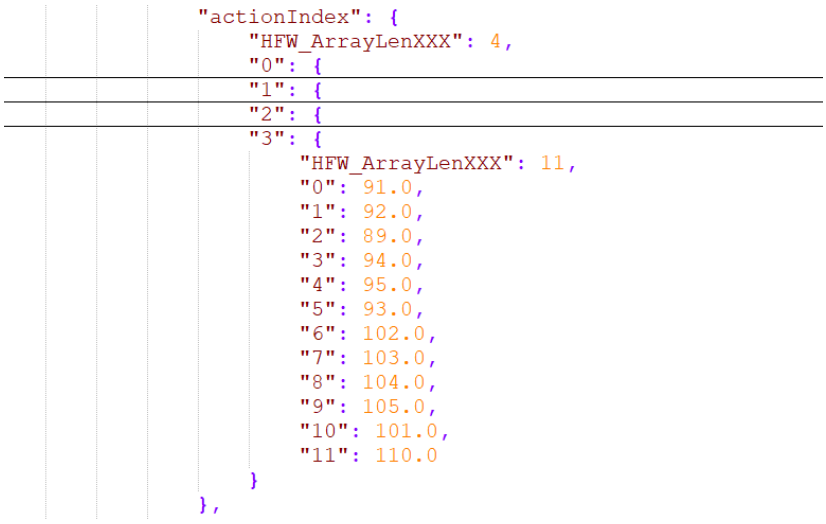
2.1 Add new Triggering key in L's "action"

Open L's Spt file's "action" > "1" > "a_keyTgr", edit the ending part as shown above. Remember to make "kl", "k" and "kr" compatible. Another thing you should remember is to set the value of "agi" according to the action's pre-action: if the pre-action is made on horse, set "agi"'s value to 4.0, which refers to "4" (whose "name" has value "rid" for "ride") in "actionGroup"; Otherwise, set "agi"'s value to 0.0, which refers to "0" (whose "name" has value "normal") in "actionGroup".


```
    },
    "1": {
        "HFW_classNameXXX": "Data.A_KeyTgr",
        "ai": 3.0,
        "atf": true,
        "hp": 0.0,
        "ati": 3.0,
        "pollhp": 0.0,
        "kl": 3.0,
        "k": "gra",
        "mp": 90.0,
        "agi": 0.0,
        "rkt": false,
        "kr": "gla"
    },
    "2": {
        "HFW_classNameXXX": "Data.A_KeyTgr",
        "ai": 11.0,
        "atf": false,
        "hp": 0.0,
        "ati": 3.0,
        "pollhp": 0.0,
        "kl": 3.0,
        "k": "guj",
        "mp": 335.0,
        "agi": 0.0,
        "rkt": false,
        "kr": "guj"
    }
},
```


2.2 Edit L's "actionGroup"

Open L's Spt file's "actionGroup" > "0" (its "name"'s value is "normal")> "actionIndex" > "3" (where L's special actions are, such as uppercut), do as shown on top of next page: add , after "10", and then add "11": 110.0 under "10" (110.0 refers to the index for L's new action, since L's last action's index before transplant is "109").



2.3 Add new action to L's "action"

Search for `"Big_COL"` in R's Spt file, and the target action is the whole `"95"`, which contains `"BIG_COL"`. Copy the whole `"95"` (that is, `"95": {...}`), open L's Spt file's `"action"` > `"109"` (the last original action), add the whole `"95"` after `"109"` as the following left code chunk, and make some changes as the following right code chunk.

```
    "109": {
    ...
    },
    "95": {
    ...
    }
}
```

```
    "109": {
    ...
    },
    "110": {
    ...
    "index": 110.0,
    "frameIndex": 421.0,
    ...
    }
}
```

the `421.0` in `"frameIndex"` is because L's last original frame's index is `"420.0"`.

2.4 Add new frames to L's "frame"

Copy R's frames from `"274"` to `"288"`, which are the frames of Thunder Strike, add them to the end of L's `"frame"`, set `"274"`, ..., `"288"` to `"421"`, ..., `"435"`, and set their `"index"` accordingly. And for their `"refIndex"`, set them accordingly if the value is NOT `0.0`.

Done. Now save the changes and use HFW to generated your game exe file.