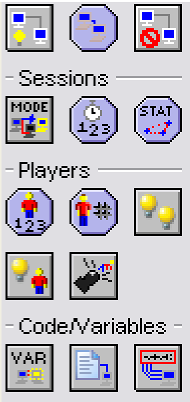

MPLAY.LIB

The Most Simplistic MPlay library you'll find.

By Josh @ Dreamland

This is the guide to my MPlay library.

READ: This library uses Gamemaker's built-in MPlay functions, however, it follows a completely different system. YOU MUST STICK TO THIS LIB! If you plan to use any of the functions in this lib, you must use nothing much but this lib. There is a global variable named player_number that uniquely identifies every player who joins. You need this variable.

In several instances I refer to a 'common event.' This is an event that is executed on every computer playing. This may be the step event of a controller, or something along those lines.

Action 1: Connect Automatically

This is a very simple action that does all the connecting for you. Provide:

- Player Name (optional)
- Session Name (anything)
- A code for if it joins to a session (EG, for transport since you don't need to wait for players)
- TCP/IP address (An IP or internet address to connect to [Use mplay_ipaddress( ) - the default - for single computer testing])

Action 2: If Connection is a Value

This simply checks the connection status. With my library, it can be **NONE, TCP/IP,** or **IPX**.

Action 3: Disconnect/End the Session

Self explanatory. Ending the session stops communication with the players (to the player that called it only,) Disconnect ends all communication and, disconnects.

Action 4: Session Mode

Set wheather or not to move to a new host (session creator) when the old one quits.

Action 5: If the number of Sessions is a Value

Check how many sessions there are to join (you won't need this)

Action 6: If the Sessions Status is a Value

Check if there is no session, if you created the session, or if you joined the session. (you likely won't need this)

Action 7: If the Number of Players is a Value

Check how many players there are. (eg, to start the game if you have enough players)

Action 8: If you are a Certain Player

See if you are a certain player (my library has actions that do this for you)

Action 9: Initialize Creation of New Players

This should be put in afther the first player that you don't need automatically created joins, but before the first player that you ***do*** need automatically created joins. The best spot, if as soon as you have two players the game starts, would be the game room's room start event. This must be in a common event.

Action 10: Create Players as They Join

This function will automatically create a player at the given coordinates as they join. The new player will also execute the code specified, which gives a possibility to set his/her player number, change the position, etc. Argument3 is specifying wheather argument0 is a *mask object*, meaning that it should always be created; or if it is a root name, meaning that you would say "player" and this function would create player1, player2, player3, etc. As they joined. This must be in a common event or in one of every player's events.

Action 11: Give a Player the Boot

Kick out the specified player and tell him/her the text in argument1. This will kick him/her out and end his/her game. This must be put in a common event or at least one executed by the player to be kicked off. To get it to work, you may want to use a global variable that decides wheather or not to kick off a player, then set it to true when he/she breaks a rule.

Action 12: Update a Variable on all Machines

Self explanitory. Place in a common event.

Action 13: Execute a Script on a Certain Machine

Self explanitory. Place in a common event if you can; if not put a reciever in the common event, and a forcer in the event that it happens in. Recievers and forcers must have the same "frequency." The reciever must have all the details in it, not the forcer.

Action 14: Execute a Code on a Certain Machine

Same as above, but with a smaller code that you type in the display window its self.