This document is intended for modders who wish to use some or all of the casino games in their own modules without getting into the game scripts.

Table of Contents

Basic Integration Instructions [NWLMOD1]

The full Neverwinter Lights casino games package includes the art resources in BlackRain_NWL_CasinoGames.hak plus the scripts and conversations found in BlackRain_NeverwinterLights_MP_ConvoScriptExport.erf. In order to incorporate the games into your own module, take the following steps:

1 - Add BlackRain_NWL_CasinoGames.hak to your module's hak list, or unpack the file's contents and add it into your module's existing hak.

2 - Import the contents of BlackRain_NeverwinterLights_MP_ConvoScriptExport.erf into your module.

3 - Create one or more dealers for each game, or useable objects for slots. *IMPORTANT* Each dealer or slot objects must have a unique object tag, even if you have multiple dealers for the same game.

4 - Attach the corresponding conversation to each dealer based on what game you wish them to manage:

Blackjack: z_convo_blackjack

Chinchirorin: z_convo_chinchirorin

Craps: z_convo_craps

Poker(Five Card Draw): z_convo_poker_5cdraw

Poker(Seven Card Stud): z_convo_poker_7cstud

Poker(Texas Hold'em): z_convo_poker_txshldm

Roulette: z_convo_roulette

5 - For slot machines, attach the script z_slots_trigger to the On Used hook of the object you're using as a slot machine.

6 - If you wish to use the rules pane functionality, attach the game-specific rules pane trigger scripts to useable objects:

Blackjack: z_blackjack_rules_trigger

Chinchirorin: z_chinchirorin_rules_trigger

Craps: z_craps_rules_trigger

Poker(Five Card Draw): z_poker_5cdraw_rules_trigger

Poker(Seven Card Stud): z_poker_7cstud_rules_trigger

Poker(Texas Hold'em): z_poker_txshldm_rules_trigger

Roulette: z_roulette_trigger

NOTE The rules pane readout will reflect the default rules for each game, as defined in their respective includes. If you wish to override rules on certain dealers and then have rules panes reflecting that (for example, a Blackjack room where dealers operate with different bet values to create high and low stakes tables), add in a string variable sHandlerTag to the rules pane object, and give it the value of the associated dealer object's tag.

At this point the games should be integrated into your module. For further game customization options, see the next section.

Detailed Game Options [NWLMOD2]

This section is intended for scripters who wish to modify game settings and logic for their own use, and is further broken down by sections for each individual game.

Script System [NWLMOD2.1]

The scripts used for the casino games follow these naming conventions:

z_[GAMENAME]_trigger - script used to initiate the game

gui_z_[GAMENAME]_inputhandler - script to handle GUI interaction from players

gui_z_[GAMENAME]_bethandler - betting GUI interaction for games with complex betting in its own script for readability

z_[GAMENAME]_rules_trigger - script used to create a GUI describing the game's rules

z_[GAMENAME]_include - game logic, included in the header of the game's trigger/inputhandler/bethandler/rule scripts

These scripts are not tied to any specific game script system, but are used by multiple games:

y_multiplayer_casino_lib - Multiplayer GUI functions used by all games

y_poker_include - common game logic used by all Poker games

z_casino_rulespane_include - rules pane functions used by all games

gui_z_casino_rulespane - script used to handle rules pane GUI interaction from players

All games except for slots have tutorials, which follow these naming conventions:

z_casino_tutorial_trigger - script used to initiate a tutorial for any game

z_casino_tut_inputhandler - script used to handle tutorial GUI interaction from players

z_casino_tutorial_include - general tutorial functions, included in the header of the tutorial trigger and inputhandler

z_casino_tutorial_[GAMENAME] - game-specific tutorial functions

In addition to the games, the casino package also includes a leaderboard system:

z_casino_stats_trigger - script used to initiate the leaderboard GUI for players

gui_z_casino_stats_inputhandler - script to handle leaderboard GUI interaction from players

z_casino_stats_include - leaderboard functions, used by stat trigger and inputhandler as well as all game includes

---

Blackjack [NWLMOD2.2]

These variables in the Blackjack include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iHard17: Whether dealer stands on hard 17s only or all 17s. -1 to disable, 1 to enable.

iFiveCardCharlie: Whether the five card charlie rule is in effect. -1 to disable, 1 to enable.

iDealerWinsTies: whether the dealer wins tied hands. -1 to disable, 1 to enable.

iDoubleExposure: Whether both dealer cards are dealt face up. -1 to disable, 1 to enable.

fNaturalPayoff: The payoff rate of a player natural blackjack. Default is 1.5.

iMaxBet: The maximum players may bet on a single round. Default is 5000.

iMinBet: The minimum bet per hand. Default is 100.

iBetIncrement: The amount a bet increments or decrements per button press. Default is 100.

These variables control game settings, and cannot be overridden in-game:

iNumDecks: The number of decks to use in Blackjack. Default is 4.

fDealDelay: Delay between each card dealt. Default is 0.5.

fPlayDelay: Delay between AI play decisions. Default is 0.75.

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

sDefaultName: AI player position name. Default is "Casino Patron".

Chinchirorin [NWLMOD2.3]

These variables in the Chinchirorin include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iMinBet: The minimum bet per hand. Default is 100.

iMaxBet: The maximum players may bet on a single round. Default is 5000.

iBetIncrement: The amount a bet increments or decrements per button press. Default is 100.

iTripleRatio: The ratio won or lost by a Triple roll. Default is 3.

iStraightRatio: The ratio won or lost by a Straight roll. Default is 2.

These variables control game settings, and cannot be overridden in-game:

iTotalFrames: Number of frames in the dice animation track. Default is 500, only modify if the GUI is changed.

iStartFrameVariance: The height tolerance for the beginning of dice rolls. Default is 20.

iBaseFrameSkip: The base number of pixels to skip per frame of die animation. Default is 40.

iFrameSkipVariance: Variance applied to iBaseFrameSkip on a per-die basis. Default is 20.

fBaseDelay: Base time delay between each frame of die animation. Default is 0.04.

iDelayVariance: Variance applied to fBaseDelay. Default is 4, each int value represents a time value of 0.01.

fInitialDelay: Initial delay between showing dice and starting the roll. Default is 0.8.

fFinalDelay: Delay between end of roll and determining roll result. Default is 0.9.

iBaseFrameSkipLoss: The base number of pixels to subtrack from iBaseFrameSkip after a bounce. Default is 10.

iFrameSkipLossVariance: The variance applied to iBaseFrameSkipLoss on a per-die basis. Default is 10.

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

Craps [NWLMOD2.4]

These variables in the Craps include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iMaxBet: The maximum players may bet on a single round. Default is 1000.

iMinBet: The minimum bet per hand. Default is 100.

iBetIncrement: The amount a bet increments or decrements per button press. Default is 100.

These variables control game settings, and cannot be overridden in-game:

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

fInitialDelay: Initial delay between showing dice and starting the roll. Default is 0.8.

fFinalDelay: Delay between end of roll and determining roll result. Default is 0.9.

iTotalFrames: Number of frames in the dice animation track. Default is 679, only modify if the GUI is changed.

iStartFrameVariance: The starting position tolerance for the beginning of dice rolls. Default is 20.

iBaseFrameSkip: The base number of pixels to skip per frame of die animation. Default is 30.

iFrameSkipVariance: Variance applied to iBaseFrameSkip on a per-die basis. Default is 15.

fBaseDelay: Base time delay between each frame of die animation. Default is 0.04.

iDelayVariance: Variance applied to fBaseDelay. Default is 4, each int value represents a time value of 0.01.

iBaseBounce: Base number of animation frames after the dice bounce. Default is 4.

iBounceVariance: Variance on iBaseBounce on a per-die basis. Default is 6.

iBaseFrameSkipLoss: The base number of pixels to subtrack from iBaseFrameSkip after a bounce. Default is 10.

iFrameSkipLossVariance: The variance applied to iBaseFrameSkipLoss on a per-die basis. Default is 10.

sDefaultName: String to put in an empty player position slot. Default is "   -".

Poker (Five Card Draw) [NWLMOD2.5]

These variables in the Five Card Draw include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iAnte: The amount each player antes up before a deal. Default is 100.

iInitialBet: The initial amount in a fixed-limit betting scheme. Default is 100.

iMaxRaises: The maximum amount of raises per betting round. Default is 3.

These variables control game settings, and cannot be overridden in-game:

fDealDelay: Delay between each card dealt. Default is 0.5.

fPlayDelay: Delay between AI play decisions. Default is 1.0.

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

sDefaultName: AI player position name. Default is "Casino Patron".

sDefaultGameName: Game name displayed on player GUI. Default is "Five Card Draw."

Poker (Seven Card Stud) [NWLMOD2.6]

These variables in the Seven Card Stud include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iAnte: The amount each player antes up before a deal. Default is 100.

iInitialBet: The initial amount in a fixed-limit betting scheme. Default is 100.

iMaxRaises: The maximum amount of raises per betting round. Default is 3.

These variables control game settings, and cannot be overridden in-game:

fDealDelay: Delay between each card dealt. Default is 0.5.

fPlayDelay: Delay between AI play decisions. Default is 1.0.

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

sDefaultName: AI player position name. Default is "Casino Patron".

sDefaultGameName: Game name displayed on player GUI. Default is "Seven Card Stud."

Poker (Texas Hold'em) [NWLMOD2.7]

These variables in the Texas Hold'em include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iAnte: The amount each player antes up before a deal. Default is 100.

iInitialBet: The initial amount in a fixed-limit betting scheme. Default is 100.

iMaxRaises: The maximum amount of raises per betting round. Default is 3.

These variables control game settings, and cannot be overridden in-game:

fDealDelay: Delay between each card dealt. Default is 0.5.

fPlayDelay: Delay between AI play decisions. Default is 1.0.

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

sDefaultName: AI player position name. Default is "Casino Patron".

sDefaultGameName: Game name displayed on player GUI. Default is "Texas Hold'em."

Roulette [NWLMOD2.8]

These variables in the Roulette include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific dealer:

iMinBet: The minimum bet per hand. Default is 100.

iBetIncrement: The amount a bet increments or decrements per button press. Default is 100.

NOTE Roulette has no maximum bet, in order to allow players the thrill of betting their entire purse on one spin.

These variables control game settings, and cannot be overridden in-game:

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

iMinIterations: Number of spin animation frames before the roulette wheel begins to stop. Default is 51.

fSpinDelay: Delay between each frame of wheel animation. Default is 0.03.

sDefaultName: String to put in an empty player position slot. Default is "   -".

These variables control game payout rates:

iStraightPayout: Payout for a single number bet. Default is 35.

iSplitPayout: Payout for a split bet. Default is 17.

iStreetPayout: Payout for a three number bet. Default is 11.

iCornerPayout: Payout for a corner bet. Default is 8.

iFivePayout: Payout for a 0-00-1-2-3 bet. Default is 6.

iSixLinePayout: Payout for a double line bet. Default is 5.

iThirdPayout: Payout for a third board bet. Default is 2.

iHalfPayout: Payout for a half board bet. Default is 1.

Slots [NWLMOD2.9]

These variables in the Slots include control game rule settings, and can be overridden in-game on a per-dealer basis by writing the variable on the specific slot machine:

iMaxBet: The maximum players may bet on a single round. Default is 500.

iMinBet: The minimum bet per hand. Default is 100.

iBetIncrement: The amount a bet increments or decrements per button press. Default is 100.

These variables control game settings, and cannot be overridden in-game:

iReel1Skip: Reel 1 pixels skipped per frame of animation. Default is 35.

iReel2Skip: Reel 2 pixels skipped per frame of animation. Default is 45.

iReel3Skip: Reel 3 pixels skipped per frame of animation. Default is 55.

iMinIterations: Minimum number of reel animation frames before the machine can begin stopping. Default is 50.

iMaxIterations: Maximum amount of reel animation frames before a stop is forced. Default is 450.

iIdleTimer: Idle time allowed players in seconds before action is taken. Default is 30.

NOTE The slots reel spin animation is graphically intensive, and its performance is dependent on the computer's specs and the level of graphical detail of the game area. Adjust the reel skip variables accordingly to achieve a good level of animation based on your specific situation. The slot machine graphics can also take significant time to load up when a player first accesses the macine. One solution to this problem is to display and close the Slots GUI during the On Client Enter event for the area, removing the Slots load time during a level in exchange for a longer level preload time.

These variables control game payout rates:

iThreeBarsPayoff: Payoff for three bars. Default is 5000.

iThreeCherriesPayoff: Payoff for three cherries. Default is 1000.

iThreePlumsPayoff: Payoff for three plums. Default is 200.

iThreeWatermelonsPayoff: Payoff for three watermelons. Default is 100.

iThreeOrangesPayoff: Payoff for three oranges. Default is 50.

iThreeLemonsPayoff: Payoff for three lemons. Default is 25.

iTwoCherriesPayoff: Payoff for two cherries. Default is 10.

iOneCherryPayoff: Payoff for one cherry. Default is 2.

The include also defines weighins for each reel of the slot machine. These settings control the odds of a reel landing on any specific window during a spin. Each reel's combined weighings add up to 64, with windows that have higher weighings on any given slot having a higher percentage chance to land on that window in a spin.

Leaderboard [NWLMOD2.10]

These settings can be modified in the leaderboard include (z_casino_stats_include).

iMaxLeaderboard: Maximum number of entries displayed on the leaderboard for any given category. While all player data is tracked and stored, only the top iMaxLeaderboard results in each category will be displayed. Increasing this value will also increase the amount of processing required each time a player leaves a game and their data is updated and allocated to the correct position within the leaderboard stats.

iMinutesPerGameHour: This must be set to the same value as Minutes Per Game Hour in module properties. This is done because the leaderboard tracks playing time in real minutes rather than game minutes so players have a better idea of how long they're spending in games.

sLeaderBoardObjTag: Tag of the leaderboard object. Default is "", which will caues the code to use the module object to store leaderboard data.

NOTE The leaderboard include is included in each individual game include, and thus is integrated with each game. If you do not wish to use leaderboard functionality in your own module, the easiest way is to delete the leaderboard trigger and inputhandler scripts, then modify the include script and render all existing functions as empty placeholder functions. This will allow games to compile without needing to remove the leaderboard hooks from game scripts individually.

Additional Notes [NWLMOD2.11]

All three poker games use a fixed-limit betting scheme. This allows players with varying amounts of gold to compete on a level playing field without having to disclose their amount of gold held to other players.

Two settings can be changed in y_multiplayer_casino_lib, which when all scripts are compiled will affect all games:

const int iNumPlayersAllowed determines how many player positions there are across all games. By changing this setting and creating new game GUIs reflecting the change, it is possible to allow for different numbers of player positions. *NOTE* This package was developed and tested for four player positions. While the code has been designed to accomodate more or fewer players based on this variable, this functionality has not been extensively tested.

const int iUsePCName determines whether game and leaderboard GUIs display only character names, or both character and Player names. Set to 0 for character names only and 1 for both names.

All casino games include an iIdleTimer variable in their include which determines how long a player can idle during their turn in games. This was included to prevent player griefing by idling in multiplayer games or occupying dealers indefinitely. Specific action upon the end of the timer depends on the specific game, and can include removing the idle player from the table or forcing a player's decision during a round. Idle timer functionality for each game can be completely disabled by setting iIdleTimer to 0 and recompiling.

In order to effect changes in game logic in a game's include script, it's necessary to first compile it, then compile all scripts that are dependent on it (trigger and inputhandler scripts).
