

Hangman Game Design

Hangman Game Display Objects

Scaffold drawn on left side of stage

letterbox: Input TextField

- Initially blank
- User types here to guess a letter

msgBox: Dynamic TextField

- Tells user to guess next letter and when the game is over

wordBox: Dynamic TextField

- Initially one blank for each letter in word
- Letters are added in their positions as they're correctly guessed

guessesBox: Dynamic TextField

- Display of letters guessed incorrectly

replayBtn: Button that appears at end of game to allow user to play again

Option 1: Drag instance of button to side of stage, where it's not visible. At end of game, change its position with ActionScript to be on the stage.

Option 2: Add replay button at end of game via addChild method. Also add listener at that time. Remove listener when user presses replay button.

MovieClips: One for each body part

Hangman Game Timeline Design

Option 1: Seven frames, with one scene per frame: the start scene with only the scaffold plus one for each additional body part, labeled Start, Head, Body, LeftArm, RightArm, LeftLeg, RightLeg. Use gotoAndStop at each subsequent scene each time the score is decremented.

Option 2: Use only one frame. Create a MovieClip class, exported to ActionScript, for each body part. Add each part dynamically to the display list via the addChild method each time the score is decremented.

Hangman Game Logic (Pseudocode)

Initialize Game:

Set score to 6 (the number of wrong guesses player is allowed)

Choose a random word for the user to guess from array of words

Display one dash for each letter in the word to be guessed

Repeat until the game is over via the Flash event-handling loop:

Wait for user to type letter in letter box, via an event listener

If (the letter the user typed is in the word)

 Display the letter instead of a dash in each position it occurs

 If the word is now complete, the game is over and the player wins

Else

 Decrement the score

 Add the letter to the guessed letters list displayed on the stage

 If (score > 0)

 Display the next body part

 Else

 The game is over and the player loses

Hangman Game: Required Variables, Functions and Events

Variables

score: an integer

- Initially set to 6, which is the number of incorrect letter guesses player is allowed before losing the game
- Decrementing when player guesses incorrect letter
- Game is over and user loses when score == 0

word: a String

- Word the user is trying to guess
- Set randomly to be one of the words in the wordlist array

wordSoFar: a String

- Dashes and guessed letters, showing how much of the word the player has guessed so far
- Initially set to all dashes: one for each letter in the word to be guessed
- When a correct letter is guessed, the letter is substituted for the dash at each position it occurs in the word to be guessed
- Displayed in wordBox text field

guessedLetters: a String

- Letters incorrectly guessed by the player
- Initially an empty String
- When an incorrect letter is guessed, the letter is concatenated onto the end
- Displayed in guessedLettersBox text field

wordList: an Array

- List of possible words for the user to guess
- One is randomly chosen at the start of the game and is placed in word

Functions

function **chooseWord**(wordList: Array) : String

- Chooses initial word for user to guess; called at start of game
- Calls random number generator to get random number between 0 and length of wordList array
- Returns word at that position in array

function **createDashedWord**(word: String): String

- Creates String with same number of dashes as letters in word
- Returns that String

function **addLettersToMatchWord**(wordSoFar: String, word: String, letter: String): String

- Finds occurrence(s) of letter in word and places letter at same position(s) in wordSoFar
- Returns updated wordSoFar

Function **processNextLetter**(event: ?)

- Main function that checks if next letter guessed is in word and then acts accordingly

Events

Player Choosing to Play Again: CLICK MouseEvent for replayBtn

Player Choosing Next Letter: We need an event that detects when the player has guessed their next letter, which they do by typing it in the letterBox text field.

Option 1: Use a FOCUS_OUT FocusEvent to check when the user has deselected letterBox.

Option 2: Use a KEY_DOWN KeyboardEvent to check when the user has pressed the enter key.