Super Slider

Technical Design Document

Daniel Schenker

Daniel Schenker

D.A.Schenker@gmail.com

7/19/2012

Super Slider

# Introduction

## Purpose

The purpose of this document is to outline the technical design of the game Super Slider.

The main purpose of this document is to explain the general method that will be used to program this game.

Please refer to the Game Design Document for an explanation on how the game works, as well as the overall concept.

# Design

## Stages

1. Pre-Game
   1. Initialize memory manager
      1. The memory manager will be used to hold the entire game, allowing for controlled allocation and deallocation control.
   2. Initialize the game manager
      1. The game manager controls all aspects of the game by calling functions whenever necessary. The game manager controls everything from the game loop to scoring, etc.
2. Menu
   1. Start - When the player is ready, they select “Start” to play the game with the desired settings.
   2. Options - The menu is where the player can set the options for the game.
      1. A checklist of information that can be shown in game or hidden.
      2. Whether a timer that counts up, or countdown that counts down to a time limit, or neither.
      3. Whether the game should start with showing the solved puzzle, and then choose “scramble”, or start off immediately in a scrambled state.
      4. Whether the move count affects the score.
   3. Quit
3. Game – Start
   1. Show the puzzle in its initial state.
   2. Start the timer or countdown if needed.
4. Game – Main Loop
   1. Check if the puzzle is solved. If so skip to step 5.
   2. Check for player move.
   3. Check validity of move.
   4. Perform move.
   5. Add to move counter. (Note: Moving multiple tiles at once still count towards multiple moves)
5. Game – End
   1. Display the player’s results as well as a high score table.
   2. Give the player the option to retry or return to the main menu.
6. Post-Game
   1. Close the game
   2. Deallocate all memory used for the game.
   3. Deallocate the memory manager.

## Functions/Tasks Involved

* Rendering
* User Interface
  + Draws both the menus as well as the in game UI.
* Image loader
  + Loads both preset images as well as checks whether user uploaded images are valid.
* Tile puzzle scrambler
  + Used to scramble the puzzle in a way that is guaranteed to be solvable.
* Check if the puzzle is solved
* Timer control
* Score calculator
* State Saver
  + Used to remember the state of a puzzle for both save files as well as for if the user would like to retry a puzzle.

# Method of Programming

## Game Engine

* Peter Chan’s engine, Dx9 SGE (Simple Game Engine).

## Programming Language

* C++