

Executive Summary

□ *Abstract of game story*

Each level is presented as a new mini-game that the player must solve in order to move onto the next level. The gimmick in this game, however, is that all mini-games are only solvable using a pair of on-screen hands, one left and one right.

□ *Game Play and Appearance*

As stated in the abstract, a pair of hands should be shown on the screen at all times. It is through these hands that the player will solve the current mini-game. Some levels may allow for the hands to move freely around a defined quadrant while other levels keep the hands in one place on the screen. In short, the control of the hands will depend on the current mini-game being played.

Also on the screen will be a status bar which shows the real-time restrictions for finishing a certain mini-game, such as time left to go, distance to go, and “misses” left. Also on the status bar will show any type of goal that must be met for that particular mini-game.

The rest of the on-screen appearance will be of a first person view of the world or mini-game. These scenes are described further in depth in the Scene Outlines section, found within the Game Specification.

Development Specification

- The estimated time to finish should be approximately three-and-a-half weeks with at least 20 hours per week going into the project.
- Code should be done in C/C++ to follow the skeleton code provided by LaMothe.
- Before every function, a comment block should be added to include the inputs, outputs, and summary of computation of that specific function.
- The code should allow for almost instantaneous response. In a game where accuracy in timing is one of the more important parts of the game, it could be the difference between winning and losing the mini-game.
- A code review should be conducted every week to make sure that the guidelines listed above are adhered to.
- Playtesting should occur after each module is complete. For example, once the hands are drawn on screen and the code is implemented, the code should be compiled and tested to ensure that indeed the intended result is achieved.

Product Specification

Production Team Description

| Division and Role | Member Name |
|--------------------------------|-----------------------------------|
| Management and Design | |
| <i>Software Planner</i> | Chris Lesnieski |
| <i>Lead Architect</i> | Chris Lesnieski |
| <i>Project Manager</i> | Chris Lesnieski |
| <i>Game Designer</i> | Chris Lesnieski |
| Programming | |
| <i>Lead Programmer</i> | Andrew LaMothe |
| <i>Programmer</i> | Chris Lesnieski |
| Art | |
| <i>Lead Artist</i> | Chris Lesnieski |
| <i>Artist</i> | Andrew LaMothe TBA |
| Music and Miscellaneous | |
| <i>Musician</i> | TBA |
| <i>Sound Effect Technician</i> | Andrew LaMothe Chris Lesnieski |
| Support and Quality Assurance | |
| <i>Technical Support</i> | Chris Lesnieski |
| <i>QA Lead</i> | Chris Lesnieski |
| <i>QA Technician</i> | Chris Lesnieski |
| <i>Playtester</i> | Chris Lesnieski TBA |

Target Audience

- This game is being developed for Windows machines so only machines with Windows operating systems will be able to run it.

- The main human interface is with a two-buttoned mouse, be it USB or PS/2. This may potentially alienate people who are trying to play on a laptop where the only pointing interface is a touchpad or eraser-head. Users will still be able to play the game, but those pointing devices are much more inaccurate than a traditional mouse.
- People who are interested by more of the games that feature mini-games within them. One example would be Incredible Crisis which came out for the original Sony Playstation. Also, because of the themed game play (two hands, use of only mouse) people that play similar games such as DDR, Donkey Konga, or Katamari Damacy, will be interested in what this title has to offer.
- Because there is no blood, violence, gore, sex, adult situations, or obscene words, this title is geared towards players anywhere from age 7 and up.

Game Play

Please see the *Pace of Game Play* section in the Game Specification.

Production Tools

- Microsoft Visual C++ 6.0 will be used as the IDE. Because of the level of familiarity with it by the programmer and the language being used, this IDE is the most obvious choice.
- Microsoft Paint will be used for light editing of graphics within the game. For quick pixel by pixel editing and putting all of the animation frames into place, this tool will give us the most bang for our buck.
- JASC Paint Shop Pro will be used for heavy editing of the graphics within the game. Things that make this tool useful is it's clone brush, cropping ability, resizing ability, and the ability to make changes to palette including the hue, brightness, and contrast.
- A to be announced sound editor will be used as well in the project. This tool should have the ability to convert from MP3 to WAV format and back, allow editing of WAV files (such as altering the pitch and adding effects), saving in multiple formats, and contain no spyware.