Nathan Chapman

S-kuru

Part II: Implementing, Testing & Evaluation

Table of Contents

Implementation....2
Installation Guide....2
Step 1: Is Your Computer Ready?....2
Step 2: Get your Copy of S-kuru....2
Step 3: Play!....3
User Manual....3
Controls....3
Section I: Play....3
Part II: Editor....6
Testing....7
Game Controls....7
Responsiveness & Intuitiveness....7
Errors & Bugs....7
Evaluation....8
Tester Evaluation....8
Game Design & Concept....8
Ease-of-Use and Longevity....8
Bugs & Errors....8
Software Objectives....8
Maintenance....10
Appendix A: User Testing Survey....11
General Questions....11
Game Controls....11
Responsiveness....11
Errors and Bugs....11
Appendix B: Game Evaluation Survey....12
Game Design & Concept....12
Ease of Use & Longevity....12
Bugs & Errors....12

Implementation

Installation Guide

This guide will help you install and set up your computer ready for playing S-kuru.

Step 1: Is Your Computer Ready?

Your Computer must meet the minimum requirements to be able to successfully run S-kuru. These requirements to run S-kuru are very low due to its use of simple 2D graphics and an optimized physics engine for collisions.

As such, the requirements below are not to be considered the absolute minimum that the game will run on; instead, they are a broad guide and a recommendation for a guaranteed successful run of S-kuru.

Type	Requirements	Justification
Operating System	Windows, Linux/Unix and Mac OSX.	Any operating system that supports Python.
Processor	1.6 GHz Single Core	Minimum required for Windows 7[1], plus some extra processing for game itself.
RAM	1GB	Can run with 20 colliding balls with no loss of frame rate from 60 fps.
HDD	25MB	This is the total size of the downloaded file (once extracted) including all level files etc.
Video Card	Not required	Game has no 3D capabilities and hardware acceleration has not been implemented as it is unnecessary for this solution
Screen	1024 x 768; 32-bit colour	Game runs at 800 x 600, 32-bit colour
Other	Full keyboard, mouse	Used to control game

Please consult your user manual for more detailed instructions on checking your system specifications.

Step 2: Get your Copy of S-kuru

On this disk is a folder called “Installation”. Alternatively, you can download this from the S-kuru project homepage (https://github.com/Crashdown/Skuru) for the most updated version of the game.

Copy this folder to your desktop (or anywhere else on your computer). You will need to ‘extract’ the folder that S-kuru is in. To do this, right-click the file on your desktop and select “Extract All…” (See

[1] http://windows.microsoft.com/en-AU/windows7/products/system-requirements

Figure 1). Follow the wizard that appears. In less than a minute, you'll have a fully working version of S-kuru on your computer!

Step 3: Play!

S-kuru is now ready to play. Launch it play double-clicking the file called "S-kuru.exe" (see Figure 2).

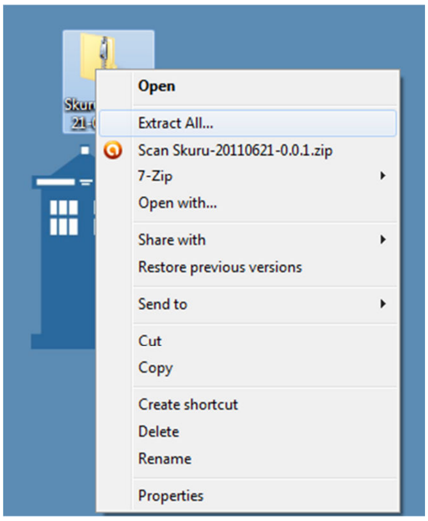

Figure 1: Extracting S-kuru

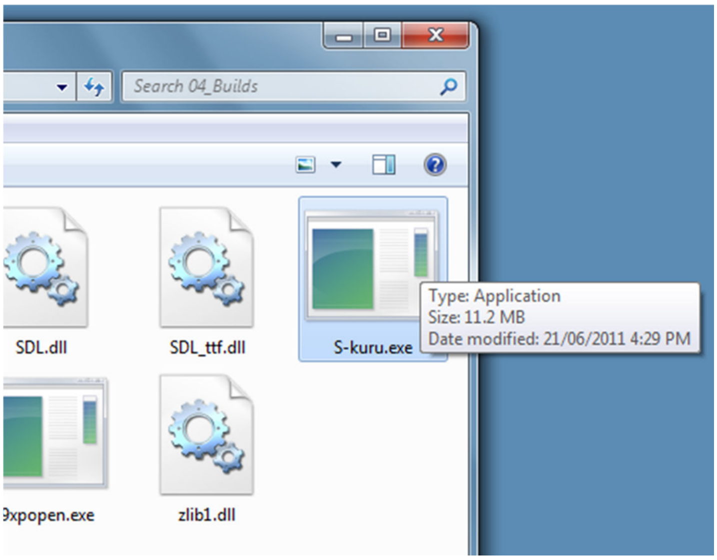

Figure 2: Launching the Game

User Manual

For ease of use, the user manual is divided into two distinct sections. Each section describes the general use of each of the two main modes within S-kuru: *Play* and *Editor*.

Controls

S-kuru could be most broadly defined as a 'point-and-click puzzle game'. What this means is that the main control of the game is by using the mouse. Full controls for the game are as follows:

Control	Action
Left-Click	*Within Menu:* Select current item *During Game:* Click and hold to draw a new circle
Right-Click	*When Editing:* Place new ball of current colour
Number buttons 1 – 4	*During Game:* Change circle colour
Enter	*Within Menu:* Confirm textbox input
ESC	*Anytime:* Quit S-kuru

The purpose of each of these actions will be explained in the two later sections.

Section I: Play

This section of the user manual will focus on the gameplay of S-kuru, including the different level modes as well as general hints and tips towards enjoying S-kuru.

Gameplay Description

To complete each level of S-kuru, every ball must get to an exit. To do this, the player ‘draws’ new circles onto the surface. Once a circle is drawn, balls can bounce down into the new area, and hopefully get to the exit. Balls of different colours can only travel through areas either of their colour, or circles of a colour that they are made up of.

To explain more clearly:

	White Circle	Red Circle	Green Circle	Blue Circle
White Ball	O	X	X	X
Red Ball	O	O	X	X
Green Ball	O	X	O	X
Blue Ball	O	X	X	O

An *X* means that a ball cannot pass through a circle of that colour; instead, it will collide as if with a normal circle wall. *O* means it can.

Game Levels

The levels in S-kuru are divided into two categories: Campaign and Custom.

The Campaign levels came with the game when you first installed it. These are the pre-made, ready-to-go levels that are professionally made.

The Custom levels are those made by users using the *Editor* game mode. These can be played by any player, and similarly to the campaign levels each player’s progress through these levels is saved.

There are variations on this concept to create a total of three different level types.

Level Type 1: Limited Circles

The player must complete the level within a set number of circles. If the player uses more than the prescribed number of circles, they must restart the level and attempt to complete it with the required amount or fewer.

Level Type 2: Timed Game

The player must race against the clock to complete the level within a certain time. Similarly to the previous level type, if every ball has not exited within the given time the level must be reattempted.

Level Type 3: Specific Colour

In this level type, the gameplay is slightly altered. Only balls of a certain colour are allowed to exit the level (for example, blue). If another coloured ball (say, a red ball) exits the level, the level will have to be restarted.

Player Profiles

Each player’s progress through the various levels in S-kuru is recorded on their player profile. This way, multiple people can play S-kuru and not interfere with each other’s progress through the campaign levels.

Level Screen

The following image will explain what the player sees when attempting a level, and what each piece of data means to the player.

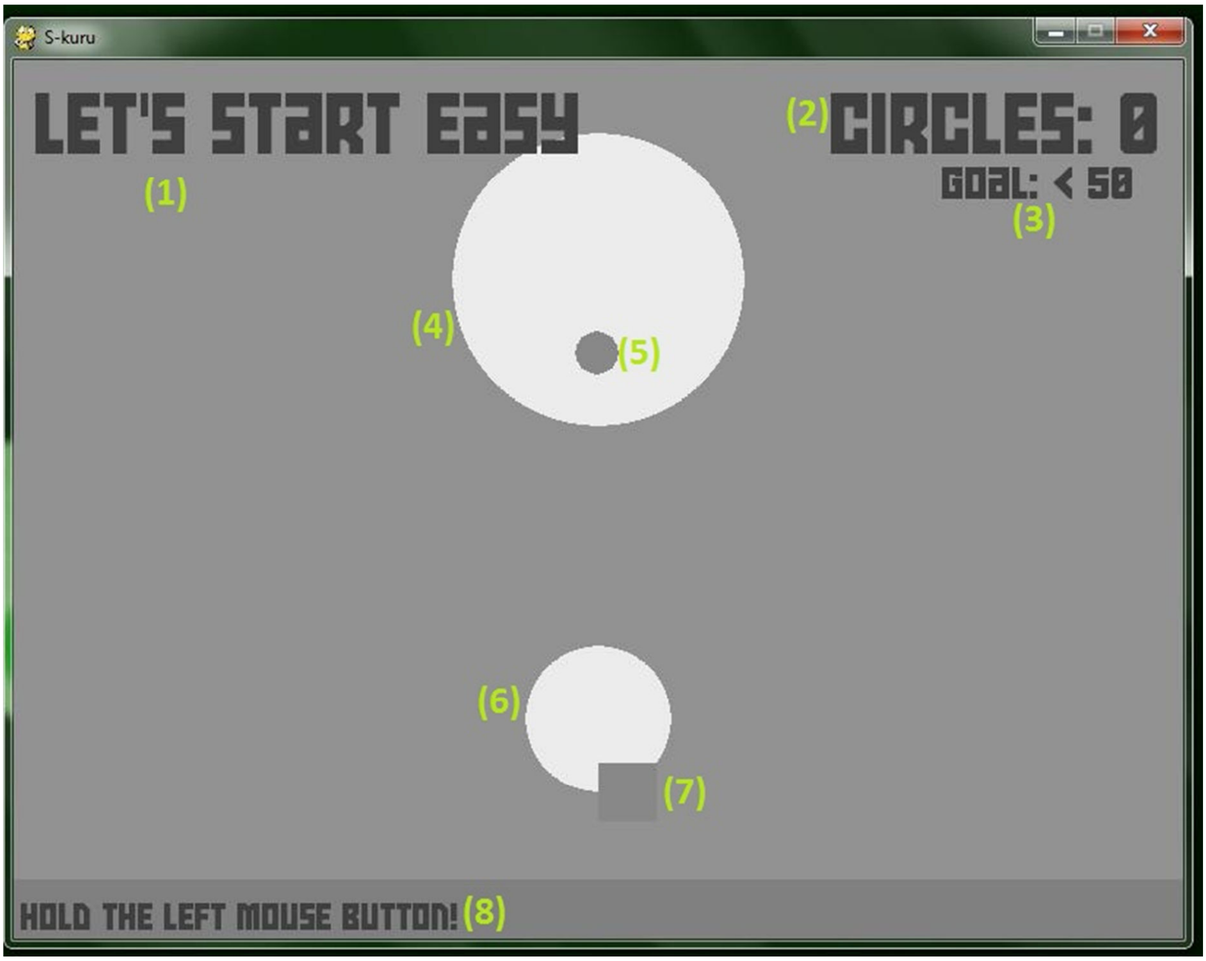

Figure 3: A Simple Level in S-kuru

The elements as numbered in Figure 3 are as follows:

1. **Level Title** – A Name for the level you are playing. This has no effect on gameplay.
2. **Number of Items Towards Goal** – the number of items you have already used towards the level limit. What these items are changes for each game type. In this example, it is counting the number of circles the user has placed.
3. **Level Goal** – Complete the level with this many items or fewer to successfully complete the level. What these items are changes for each game type. In this example, the player has to complete the level using less than 50 circles.
4. **Circle** – One of the main objects players can interact with. Players place new circles by holding the Left Mouse Button. Balls can bounce down and into these circles once they are placed.
5. **Ball** – One ball. There can be more than one ball in each level, and balls come in varying colours.
6. **Another Circle** – See #4 above.
7. **Exit** – Where balls must get to the complete a level. Exits are distinguishable because of their square shape. Different coloured balls will have an exit of the same colour.
8. **Hint Text** – A little bit of text. It has no effect on gameplay, but may provide some helpful hints (or a few jokes!).

Part II: Editor

Although this module is not available to users yet as it is not completed, a basic run-down of how to use the module follows. This is to help users understand the module when it is first available through a version update.

Creating a level in the Editor is much the same as playing a level normally:

- Click-holding the Left Mouse Button grows and places a circle.
- Using the number keys 1 – 4, players can change the colour of the circle
- Balls can be placed in the currently selected colour using the Right Mouse Button

Once the user has saved and completed their level, they will be asked to add a level name, goal and hint text. This data will then be saved to a file in the 03_userLevels directory, and can be played from the Play menu as any other level.

Testing

To completely test the game, I had several students play a release candidate of the game. Although unfinished, this version of the game provided me with information on a few key areas that needed improving for the final build of the program. One major issue with testing the game with this release candidate is that the testers were only able to experience the game through the campaign levels – the level editor was not completed at this stage.

Game Controls

The control scheme and button assignments were generally well received. Assigning the circle colours to the number keys 1 – 4 was familiar to many testers, as this is commonly used to change weapons within a First-Person Shooter game. One player, a "non-gamer" found that this control scheme was cumbersome in the timed levels (where speed was a priority for completing the level). This was (as I noticed during their testing) they would only use one hand, alternating between mouse and keyboard when required. Unfortunately, this slowed down their speed at completing levels, which in some levels caused them to have to re-attempt the level.

I have determined however that this is not necessary to 'fix', as it is an isolated problem, only occurring on one or two levels. In addition to this, with practice the player can easily train themselves to use both hands (as gamers do). Indeed, even within the single testing session the tester that was having this problem began to assimilate to the 'traditional' placement of hands on the keyboard.

Responsiveness & Intuitiveness

The game is very responsive to player input, mainly due to the clever coding of the Button() class which allowed for automatic rollover effects. This allowed for visual feedback to the user about the state of their mouse, and to differentiate text from buttons (as they both the same typeface).

The choice of using PyGame, a purpose-built set of modules for creating games in Python also paid off. The testers experienced no problems in lag or slow responsiveness during the game, except during the initial loading screen. The lag in this section is due to my use of time.sleep(), which effectively pauses executing of the program for a set number of seconds. During this 'sleep' time, the GUI window could not be moved, minimised or closed. Although this should not be an issue, it could prove to be cumbersome for some users in the (very unlikely) event that S-kuru was to crash part way through this stage of execution.

Errors & Bugs

Various minor bugs were found during testing & fixed almost immediately. These included things like:

- Problems parsing data between profiles
- Inputting profile name as "Name" vs. "name" created two separate profiles
- Impossible levels
- GUI / Rendering Bugs
- Ball bouncing or physics errors (cosmetic only)

Evaluation

Tester Evaluation

Game Design & Concept

The concept of the game, as explained in the User Manual and from the play tester's own experiences in playing the game was received in a very positive manner. The 'core mechanic' (this term was described in Part I of the report) of the game was "remarkably clever", one tester said in their response to the survey questions.

The actual design of the game was regarded as stylised, however there was some disagreement amongst testers as to whether the colour scheme was either stylised or dull. Once later levels were introduced with coloured circles and balls, the testers found this much more visually appealing. Perhaps the integration of an accent colour into the initial design of S-kuru could have helped, however I feel that this would have detracted from the primary colour scheme.

Ease-of-Use and Longevity

Without being given the user manual, there were some issues interacting with the game. For example, although during the levels the hint system was in place providing a tutorial through the first few levels, the menu system itself lacked any explanation. Whilst this was not a problem for simple button-clicking choices, for the screen where the user has to enter their player profile name this became almost impossible to pass – it not occurring to players to hit [Enter] to submit and continue.

The hints system providing a tutorial for the unwitting player helped make the game very self-explanatory, and many users found themselves easily finishing most of the basic levels.

As for the longevity of the game, it can be infinitely expanded (well, almost: up to 1,000 levels in each of the campaign and user/custom modes) with new and exciting levels. There could be a minor problem in that it could become difficult to continue to create new and exciting levels; this however could easily be fixed with a new patch or version introducing some new elements into the game whilst keeping the same familiar *core mechanic*.

Bugs & Errors

Similarly to the previous tester survey, I asked this time if they could find any bugs and if any previous bugs had been fixed. There were no major issues to report in this area.

Software Objectives

The objectives for S-kuru were defined as the following in Part I of this report:

1. ***Simplicity of Use*** – the game must flow and play smoothly, and the controls must be intuitive. Not only in which button does what, but in (for example) how fast a circle grows as the user holds their mouse down. More measureable aspects of this include:
    a. Simple and feedback about progress of a level. Clear feedback about progress through the game.
    b. Anything that is 'clickable' throughout the game should be labelled or designed in such a way as to make it visually so.

2. ***Complexity of Thought*** – the game should provoke thought within the program's use and force the user to consider each move and its implications within the game.
    a. Puzzles (i.e. individual levels) should provide a differing experience as they progress. They should not be repetitive.
    b. There should be multiple solutions to puzzles, even if some of these were not originally considered by the level designer.
3. ***Consistency whilst constantly evolving*** – the game should shift and change constantly so as to be become stagnant and dull. Throughout this change, however, the gameplay *core mechanic* should remain the same, as will level design. This allows for a consistent user experience throughout a varying game.

Objective	**Implementation Score – out of 5**		**Evidence & Explanation**
Simplicity of Use – clear feedback during level	Achieved	5	Overall level design, with level information clearly displayed at the top of the screen
Simplicity of Use – visually intuitive interactivity	Partially Achieved	4	Buttons have rollover effect, however restricted colour scheme can make clickable items unclear at times.
Complexity of Thought – Puzzle design & concept	Achieved	5	Survey of testers responded very positively to game concept and variety of puzzles
Complexity of Thought – Multiple solutions to puzzles	Unclear	--	Current campaign levels may or may not have multiple solutions; designed as tutorial levels, they are simplistic and testers only found 'intended' solutions
Consistency whilst constantly Evolving	Partially Achieved	4.5	All three level types are based off the same concept; however the levels may become somewhat repetitive eventually due to a lack of other mechanics in the game.

Maintenance

Although not directly connected to user feedback about a prototype of S-kuru, at around the half-way stage of development it was decided to spend what would turn out being a significant chunk of development time restructuring the source code of S-kuru and partially rewriting most modules. The main reason for this was poor planning of module design during the Pre-Implementation stages of the Software Development Cycle, and was likely a result of using the Prototyping Method for developing software.

This rewriting involved simplifying much of the logic used in the game (particularly in the physicsEngine module) as well as the removal of superfluous functions and code that was no longer required for the successful execution of the program.

Despite having some users regularly testing prototypes throughout the development of S-kuru, the two official and structured testing phases occurred (as recommended in the assessment guidelines sheet) during the two tester surveys. Following these, some major (and some minor) changes were made to the game to improve the overall product.

Some basic fixes to the game's GUI and the physicsEngine bounce and collision logic helped to stabilise the game and make it more user-friendly. I didn't change any of the controls as I felt that they were already intuitive and clear for users that were already familiar with gaming conventions (and indeed, the one tester that was having problems with the controls was beginning to get the hang of changing colours with the number keys anyway).

The mechanics of the game were rethought following the testers' advice, and despite some thoughts that it was over-simplified I believe that with proper level designs it could be made sufficiently complex and sophisticated.

Many bug-fixes were also undertaken after testers' finding the various crashes etc., and several small bugs to exist in the software solution due to the form of development environment created by the Prototyping approach.

Please Note: due to the large amount of paper that would be required (over 20 sheets), the Logbook for S-kuru has only had the first few pages printed. The rest of the logbook can be accessed from the CD, which also contains digital copies of this report.

Appendix A: User Testing Survey

The following questions appeared in a survey that accompanied the various test builds of S-kuru. Testers were asked to play the game thoroughly before responding. Similarly to how the results of the focus group are presented above on page 4, the questionnaire itself was divided into sections. The same survey was given during both the first and second rounds of testing (in the assignment sheet, these rounds are separated between the 'Testing' and the 'Evaluating' sections).

General Questions

These are used only as a baseline and to help explain later answers. They are not used to personally identify you.

1. Age
2. Competency with Computers? (Rated from 1 – 5)
3. How often would you play computer games per week?

The following questions are used to ensure that you have testing using the latest and most updated version of S-kuru.

4. Today's Date
5. Version Number of S-kuru (Folder Name)

Game Controls

1. Rate the game's ease-of-use (1 – 5)
2. What is each button/control designed to do?
    a. Left Mouse
    b. Right Mouse
    c. ESC
    d. 1
    e. 2
    f. 3
    g. 4
3. Describe how you played the game
4. Would you change these controls? If so, how?

Responsiveness

1. When you clicked the left mouse button, what happened?
2. How quickly did this happen?
3. Was it clear what each control would do?
4. Rate the responsiveness of S-kuru (1 – 5)
5. Does the responsiveness of the game need to be improved at any point?
6. Is S-kuru intuitive to play? How or how not?

Errors and Bugs

1. Please list any errors, game crashes, freezes or issues you had with the game whilst playing.

Appendix B: Game Evaluation Survey

This survey was given to a small group of testers following revisions to S-kuru based on their feedback from the Testers' Survey (see Appendix A). It was a short survey divided into three main sections for legibility.

Game Design & Concept

1. What are your first impressions when you launched the game?
2. Rate the design of the main menu (From 1 – 5)
3. Would you like to see the menu design changed or improved? If so, how?
4. Rate the design of the levels (From 1 – 5)
5. What, if anything, would you change about the level design?
6. Did you find the levels clear and know what you were expected to do?

Ease of Use & Longevity

1. How did you find the gameplay on the first level? Did you understand what was expected to complete the level?
2. Did you notice the hint text at the bottom of the level? Did it help you at all?
3. How could S-kuru be improved to be more intuitive?
4. How long did the levels take you to complete?

Bugs & Errors

1. If you encountered any bugs last time, have they been fixed?
2. Please list any errors, crashes or bugs that you encounter whilst playing S-kuru.