|  |
| --- |
| Nathan Chapman |
| S-kuru |
| Part II: Implementing, Testing & Evaluation |

|  |
| --- |
|  |

Table of Contents

[Implementation 2](#_Toc296967727)

[Installation Guide 2](#_Toc296967728)

[Step 1: Is Your Computer Ready? 2](#_Toc296967729)

[Step 2: Get your Copy of S-kuru 2](#_Toc296967730)

[Step 3: Play! 3](#_Toc296967731)

[User Manual 3](#_Toc296967732)

[Controls 3](#_Toc296967733)

[Section I: Play 3](#_Toc296967734)

[Testing 5](#_Toc296967735)

[Game Controls 5](#_Toc296967736)

[Responsiveness & Intuitiveness 5](#_Toc296967737)

[Errors & Bugs 5](#_Toc296967738)

[Evaluation 6](#_Toc296967739)

[Game Design & Concept 6](#_Toc296967740)

[Appendix A: User Testing Survey 7](#_Toc296967741)

[General Questions 7](#_Toc296967742)

[Game Controls 7](#_Toc296967743)

[Responsiveness 7](#_Toc296967744)

[Errors and Bugs 7](#_Toc296967745)

[Appendix B: Game Evaluation Survey 8](#_Toc296967746)

# 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]](#footnote-1), plus some extra processing for game itself. |
| RAM | 1GB | Can run with 20 colliding balls with no loss of framerate 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 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).

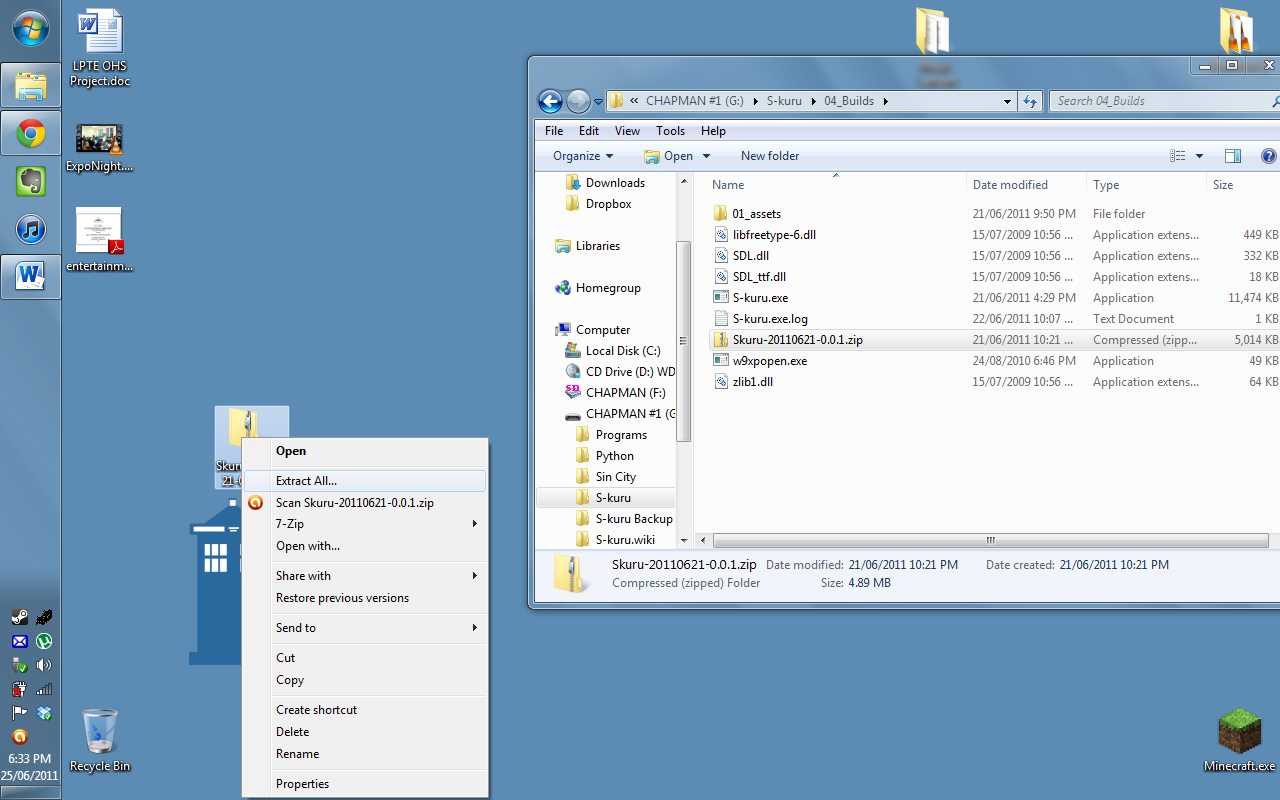


Figure 1: Extracting S-kuru

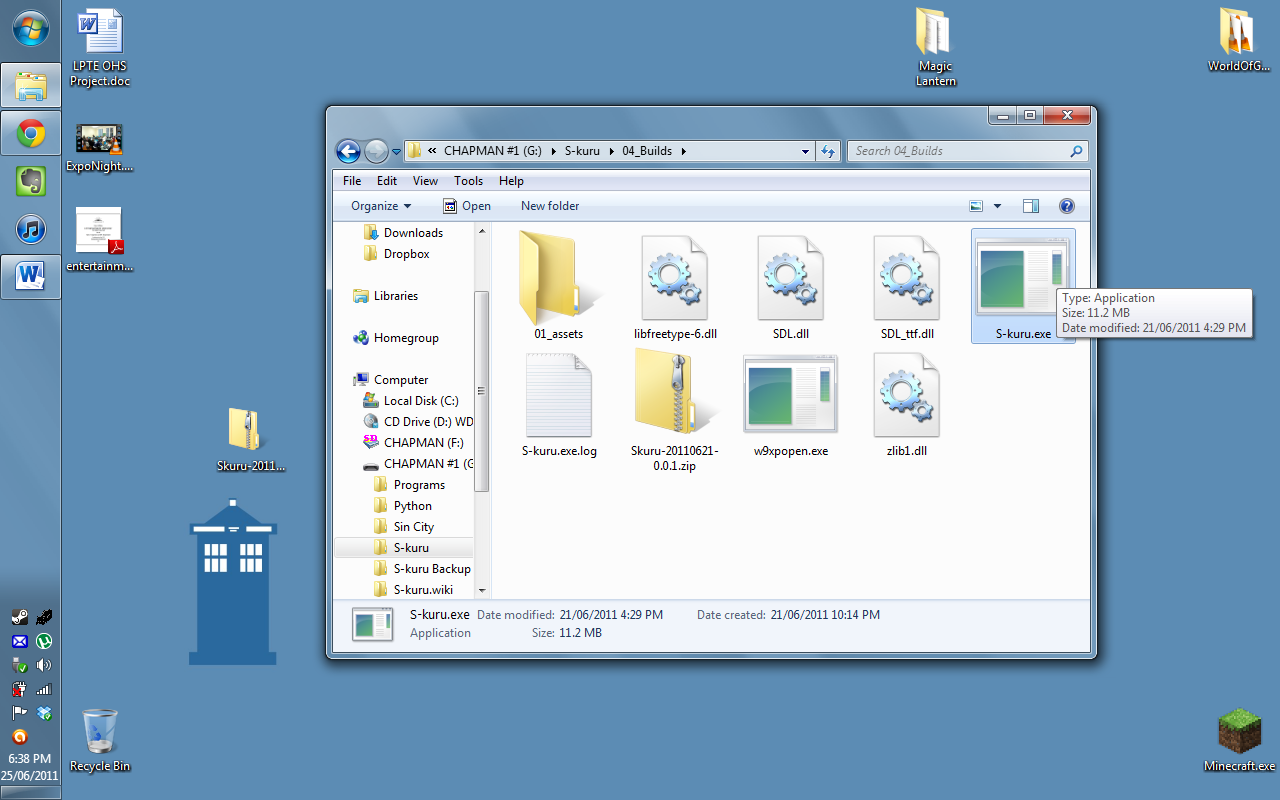


Figure 2: Launching the Game

## User Manual

For ease of use, the user manual is divided into two distinct sections. These each describe 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.

# 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

## 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 a main colour into the initial design of S-kuru could have helped, however I feel that this would have detracted from the primary colour scheme.

# 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 5, 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.

1. Today’s Date
2. 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?
   1. Left Mouse
   2. Right Mouse
   3. ESC
   4. 1
   5. 2
   6. 3
   7. 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

1. http://windows.microsoft.com/en-AU/windows7/products/system-requirements [↑](#footnote-ref-1)