Minions

Software Design Document

CMPT 276 Assignment 3

February 16, 2015

Team Yog-Sothoth
by N Sumner, W Gilman, L Whateley

wsumner@sfu.ca	std# xxxx-xxxx
wgilman@lovecraft.sfu.ca	std# yyyy-yyyy
lwhateley@lovecraft.sfu.ca	std# zzzz-zzzz

Contents

1 Revision History Table

Rev	Date	Summary	Author
1	1 February 2015	Initial revision	N Sumner
2	10 February 2015	Added UI mockups and use cases	N Sumner
3	16 February 2015	Updated end of game notification. Allow the user to press the current location for a “don’t move” turn. Added a concrete example game.	N Sumner

2 Gameplay

Minions is a game in which the player seeks to evade a number of minions while those minions follow the player around cells on the game board. The player taps a cell adjacent to their current location in order to move there. All minions then move one cell closer to the player's new location. If any minion is able to occupy the same cell as the player, then the player is captured. If two minions occupy the same cell, they crash into each other and are immobilized. When no mobile minions remain on the board, the player has escaped capture and won the game. The player tries to escape capture with the fewest number of moves.

2.1 Example Game

1. The game starts with minions placed randomly in unique locations on the board and the player placed randomly in another unique location.

2. The player moves to an adjacent location and all of the minions move one adjacent cell closer to the player.

3. When the player maneuvers two or more minions to occupy the same location, the minions are immobilized, and they are depicted differently on the board.

4. Careful movement allows the player to use already immobilized minions as traps for other minions.

5. Tapping the player's current location moves the minions without moving the player.

3 Software Requirements Specification

3.1 Functional Requirements

1. The application's first screen must be a nice looking, animated welcome screen.
    1) Program must start up showing the welcome screen.
    2) The welcome screen must include at least the following elements:
        - Name of application
        - Name of application's author(s)
        - One or more images. For instance, it could include a picture or cartoon of the authors, an icon for the application, or related images.
    3) The welcome screen must include at least two different animations, such as fade, spin, or move.
    4) One of the welcome screen animations must be a more complicated animation such as rotating or moving a block of elements at once.
    5) The welcome screen must transition to Main Menu once all animations are finished.
    6) The total time the Welcome screen is displayed should be between 3 and 10 seconds.
    7) The welcome screen must have a button or similar interface that allows the user to skip the animations and go directly to the Main Menu.
2. The Main Menu must allow the user to navigate to the game, options, and help screens.
    1) Display a button to navigate to the Game screen.
    2) Navigating to the Game screen creates a new game with the correct configuration specified on the Options screen.
    3) Display a button to navigate to the Options screen.
    4) Display a button to navigate to the Help screen.
3. The Game screen must allow the user to play the Minions game.
    1) Display text stating how many minions total there are on the game board, mobile or otherwise.
    2) Display text stating how many mobile minions remain.
    3) Display text stating how many moves the player has used so far this game.
    4) Display text stating the total number of games started, starting at 1. This should persist across application launches.
    5) Display a grid of buttons or UI elements which have button-like functionality. The grid dimensions are set by the options screen.
    6) The starting number of minions on the game board is set by the options screen.
    7) The starting number of minions are randomly placed on different starting locations on the board.
    8) The player is randomly placed on an unoccupied location of the board.
    9) Tapping a grid button adjacent to the player moves the player to that cell. Upon moving:
        1) All mobile minions are moved one cell closer to the player both vertically and horizontally.
        2) If any two or more minions occupy the same cell, they crash into each other and become immobile.
        3) If any minion and the player occupy the same cell, the minion has caught the player and the game is over.
        4) The displayed number of mobile minions is updated.
        5) The number of rounds is updated.
    10) The locations of all minions and the player must be indicated on the game board via an appropriate image.
    11) Mobile and immobile minions must be shown using different images.
    12) Player movement may be animated to show the cell to which the player moves and the fact that it lures the minions to that location.

13) The game screen may contain buttons for Teleport and Shield options.
14) Teleport and shield may each be used once per game.
15) Upon clicking Teleport, if teleport has not been used, the player's next move may be anywhere on the board.
16) Upon clicking Shield, if the shield has not been used, all minions immediately adjacent to the player are immobilized.

4. When the player wins or loses, notify the player and return to the Main Menu.
    1) When the either the last mobile minion or the player is immobilized, redraw the game board showing the updated minion images and updated minion counts.
    2) Display a dialog to congratulate/console the player for their respective win/loss.
    3) The dialog must have at least one image and some text notifying the player.
    4) When the player dismisses the dialog, e.g. taps OK, return to the Main Menu.
    5) From the Main Menu, pressing the Android back button must then quit the application.
5. The options screen must allow the user select the board size and number of minions.
    1) User can select the board size, from options including at least:
        - 3 x 4
        - 4 x 6
        - 8 x 12
    2) The board should always be organized so that the longer dimension, e.g. 4 in 3x4, stretches along the longer edge of the phone.
    3) User can select number of minions, from options including at least:
        - 6 minions
        - 10 minions
        - 15 minions
        - 20 minions
    4) The game size and number of minions are saved between application runs.
    5) Do not allow the user to select any invalid configurations, such as a 3x4 grid with 20 minions.
    6) Allow the user to reset the total number of games started to 0.
6. The Help screen displays some information about who wrote the application and some text explaining the game.
    1) The about-the-author text must include a hyperlink to the CMPT 276 home-page.
    2) The game information text must explain some of the basics about the game. must use your own wording, not copying the text from the assignment document. Your text should reflect the theme of your game.
    3) The Help screen must provide the correct citation for any images, icons, or other resources (such as music) used in the game. Include a hyperlink if applicable.
    4) If the user taps anywhere on the help screen, return to the Main Menu.
    5) Pressing the Android back button on the Help screen returns to the Main Menu.

3.2 Nonfunctional Requirements

1. The application must have the game play described in this document, but you may choose a different theme.
    1) The theme must affect the images chosen for backgrounds, button images, and the text of the game and help screen.
    2) The theme may affect the name given to your application; it need not be Minions.
2. Application source code must be maintainable.

1) Code must be well organized into methods and classes.
2) Game logic must be in a separate class from game UI class. (Hint, watch this video or read about Model-View-Controllers)
3) Game logic may be in a separate Java package from UI code.
4) Code must use good naming conventions and have good layout (indentation).

3. The application runs on Android smart phones and tablets.

1) The application must run under at least Android OS version 4.1 (Jelly Bean) or newer.
2) The application may support running on Android OS version 2.3.3.
3) The application must display well at a screen of size size of 4 inch 480x800 pixels. ("Nexus S" configuration).
4) The application may support large screens, such as 7" and 10.1" tablet screens.
5) The application must support horizontal (landscape) layouts.
6) The application may also support vertical (portrait) layouts. If this is supported (allowed), the game must look good and be playable in this layout.

4. Pressing the "back" button on the Android phone must always take the user to the previous screen in a reasonable way.

1) From the Welcome screen, back must exit the program.
2) From the Main Menu, the game must exit without returning to the Welcome screen.
3) From other screens, back must return to the previous activity on the activity stack.

5. The application should appear complete and well built.

1) The application must have an appropriate (non-default) icon.
2) Each screen must have a background image.
3) Each screen may use the same background image.
4) All text must be clearly readable over the background.
5) All text that appears on the UI must be read from the strings.xml file to support internationalization.

6. Quick to learn for a new user.

1) An average grade 10 student must take no more than 2 minutes to learn to play the game after a 60 second demonstration on how to use the application.

7. Must be responsive to the user.

1) Each user interaction (such as a button press) must start to generate its response within 0.5s when run on a real device. (Looser performance criteria applied for running in the emulator).
2) Application may play a feedback sound when the user performs an interaction.
3) Application may generate a short vibration when the user performs an interaction.

4 Use Cases

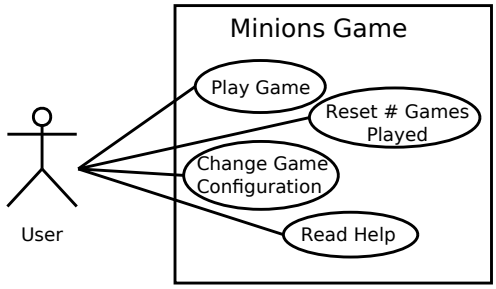

Figure 1: Use case diagram for the Minions game

5 Scenarios

5.1 Scenario: Play a Game

5.1.1 Initial Assumptions

- User is logged into the phone and has just launched the application.

5.1.2 Normal Workflow

- User sees the welcome screen and animations. After the animations complete, it automatically advances to the Menu screen.
- User selects button to play game.
- User taps on game cells to move around the board and flee the minions.
- Game displays information about number of immobile minions in the row and column of each cell.
- Once all minions are immobilized, user sees congratulations message and is returned to the Main Menu.

5.1.3 Variation: Change Game Size

- Before starting the game from the Main Menu, user selects to go to Options screen.
- User changes game board size and number of minions in game.
- Use presses the Android back button to return to Main Menu.
- Resumes normal workflow to begin playing game.

5.1.4 Concurrent Activities

- User may navigate away from application and return to the application without losing its state so long as the application does not close. If the application is closed, the game state may be lost.

5.1.5 State on Completion

- User is viewing the Main Menu.

6 Requirements Validation

6.1 UI Mockup

1. The user launches the game. The welcome animation plays, and the program advances to the main menu.

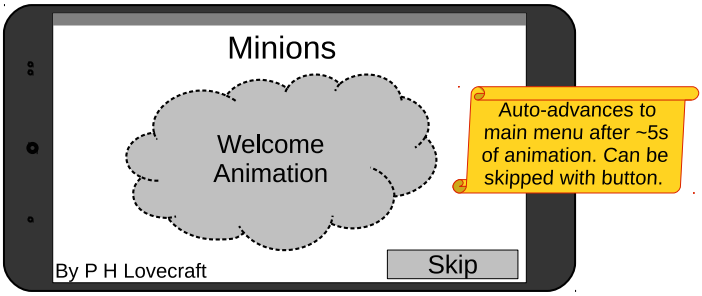

Figure 2: The welcome splash screen automatically advances after the animation.

2. The user taps a button to play and changes to the game screen.

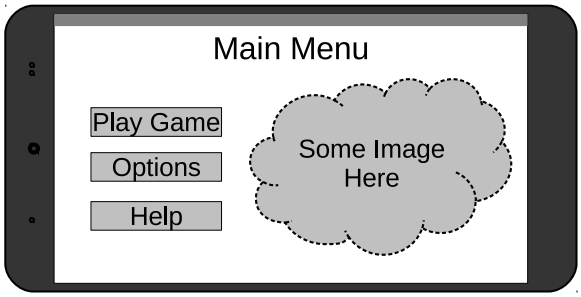

Figure 3: Menu allowing access to the game, options, and help

3. The user sees the game screen with a grid of buttons. The user taps buttons to flee and immobilize minions.

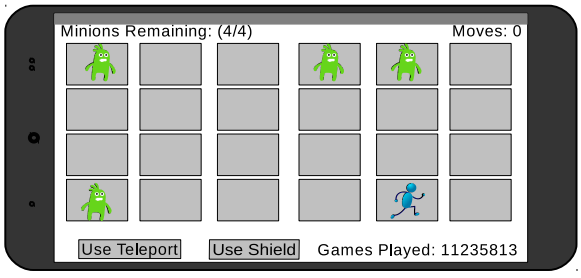

Figure 4: Game screen before any moves by the user

4. As the user immobilizes minions, the minion count updates.

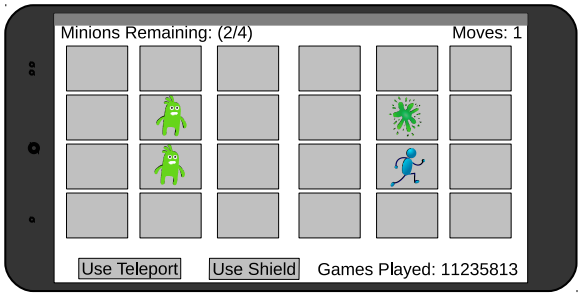

Figure 5: Game screen after a move by the user

5. Once the user has immobilized all minions, they see a congratulations message and automatically return to the main menu.

Figure 6: Congratulatory message for winning

6. From the main menu, the user selects the options screen.

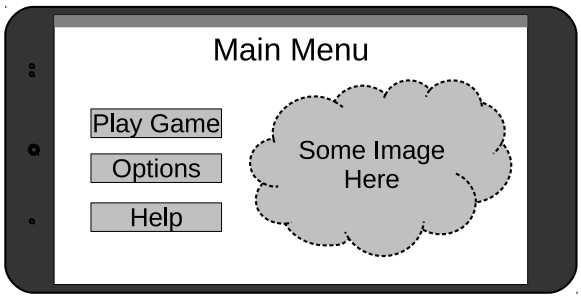

Figure 7: Menu allowing access to the game, options, and help

7. On the options screen, the user can change game settings.

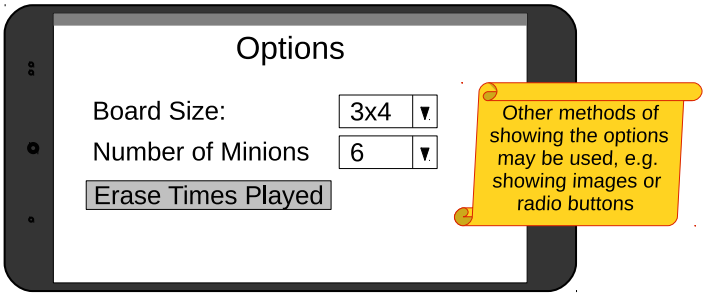

Figure 8: Options screen for changing game settings

8. The user returns to the main menu and navigates to the help screen.

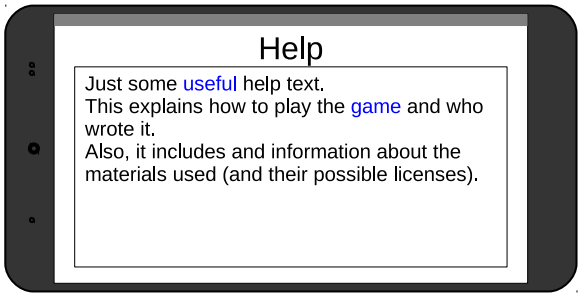

Figure 9: Help screen with directions and information about the game's creators.