

# RPG Kit

## User and Technical Documentation

Philip Regan  
<http://www.oatmealandcoffee.com>

May 31, 2012

### Contents

|          |                                    |           |
|----------|------------------------------------|-----------|
| <b>1</b> | <b>About</b>                       | <b>3</b>  |
| <b>2</b> | <b>Playing a Game</b>              | <b>3</b>  |
| 2.1      | Accomplishments . . . . .          | 3         |
| <b>3</b> | <b>Editing a Game</b>              | <b>4</b>  |
| 3.1      | Game Selection . . . . .           | 4         |
| 3.2      | Game Mode . . . . .                | 5         |
| 3.3      | Game Component Selection . . . . . | 6         |
| 3.4      | Game Information . . . . .         | 6         |
| 3.5      | Player information . . . . .       | 7         |
| 3.6      | Tasks . . . . .                    | 7         |
| 3.6.1    | Editing tasks . . . . .            | 8         |
| 3.6.2    | Locations . . . . .                | 8         |
| 3.6.3    | Accomplishments . . . . .          | 8         |
| 3.6.4    | Linking Tasks . . . . .            | 8         |
| <b>4</b> | <b>Technical Manual</b>            | <b>12</b> |
| 4.1      | Model . . . . .                    | 12        |
| 4.2      | View Controllers . . . . .         | 12        |
| 4.3      | Data Controllers . . . . .         | 12        |
| 4.3.1    | OCGameDataController . . . . .     | 12        |
| 4.3.2    | OCGameEngine . . . . .             | 12        |

## List of Figures

|    |   |    |
|----|---|----|
| 1  | Home Location during game play. There are no Accomplishments here. . . . .  | 3  |
| 2  | Location with an Accomplishment. . . . .  | 3  |
| 3  | Pressing an Accomplishment will show a detail view. . . . .   | 4  |
| 4  | Pressing the “Attempt Accomplishments” button will pit you against the game itself. . . .   | 4  |
| 5  | Losing an Accomplishment can have its consequences. Playing chess is but one of them. . . .   | 4  |
| 6  | The Game Selection screen is the “home” screen for the app. All navigation ends up here. . .  | 5  |
| 7  | A sample is created by pressing the “New” button. . . . .   | 5  |
| 8  | Games, and everything in them, can be deleted with the “Edit” button. Simply press the<br>“Edit” button a second time to cancel without deleting. . . . . | 5  |
| 9  | The Game Mode screen is the “home” screen for a game. . . . .   | 6  |
| 10 | Game Components that can be edited. . . . .   | 6  |
| 11 | The Game information screen . . . . .   | 7  |
| 12 | The Player information screen . . . . .   | 7  |
| 13 | The Task information screen . . . . .   | 8  |
| 14 | Tasks can be edited and deleted, but they must also be linked according to their use later. . .   | 8  |
| 15 | Locations Linked to another Location. . . . .   | 9  |
| 16 | Delinking Locations is a simple matter of deleting from the list, but this does not delete the<br>Location from the game. . . . .                         | 9  |
| 17 | Multiple Locations can be linked at once. . . . .   | 9  |
| 18 | Managing Accomplishments is at the bottom of the Task Information Menu . . . . .  | 9  |
| 19 | All of the Accomplishments linked to a Location . . . . .   | 10 |
| 20 | Accomplishments are managed the same way as Locations . . . . .   | 10 |
| 21 | Simple map layout for the Auto Racing and Office games . . . . .  | 10 |
| 22 | Wumpus map to show how complex linking Locations can become with a bit of effort. . . . .   | 11 |

# 1 About

RPG Kit is an application that allows you to create and play simple role-playing games on the iPhone. This is a port of the original Android version and an extension of that work to include functionality that did not get completed.

## 2 Playing a Game

Games are essentially composed of these objects:

- *A player*—your character in the game
- *Locations*—which can be your home, a store, a race, or just about anything. Tasks linked to other tasks as siblings are Locations.
- *Accomplishments*—Tasks that are linked to locations as things to do are Accomplishments.

The application starts off with these sample games to play and edit:

- *A Day at the Races*—Like Gran Turismo, but without the hassle of actually driving the car.
- *Back At The Office*—Meanwhile, in the real world, this week is your turn in the Office Donut Club.
- *Hunt the Wumpus*—Only problem is...there's no Wumpus.

Game play is based on going to Locations doing any Accomplishments listed, and then going on to another location listed. Outcome of the Accomplishments is largely random with modifications based on the player's abilities, the Accomplishment's difficulty, and associated bonuses and penalties.

If you are at Home, there are no Accomplishments, but you may pass time to either gain income and/or repair any damage, heal, and the like should those be enabled. Home should always be a safe haven from everywhere else in the game.

### 2.1 Accomplishments

Accomplishments are types of Tasks that the Player is pitted against at a given Location. In order to perform the Accomplishments, simply press the “Do Accomplishment” button to have the game see how well you do. Each Accomplishment has a set of rewards and penalties that are applied to the Player

based on how well they do. There are no ties, and if a Player “loses” before all of the Accomplishments are completed, then game will stop and return the Player to the Location.

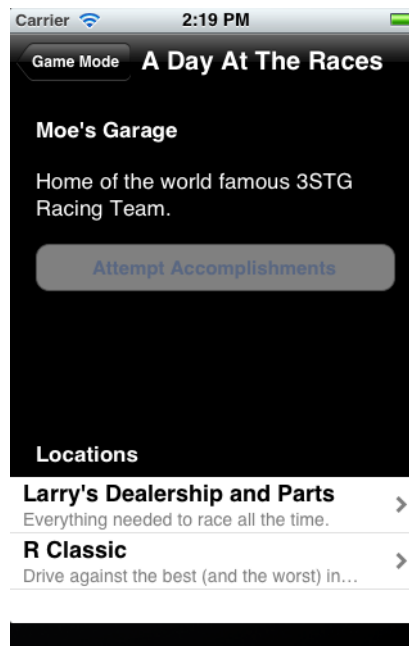


Figure 1: Home Location during game play. There are no Accomplishments here.

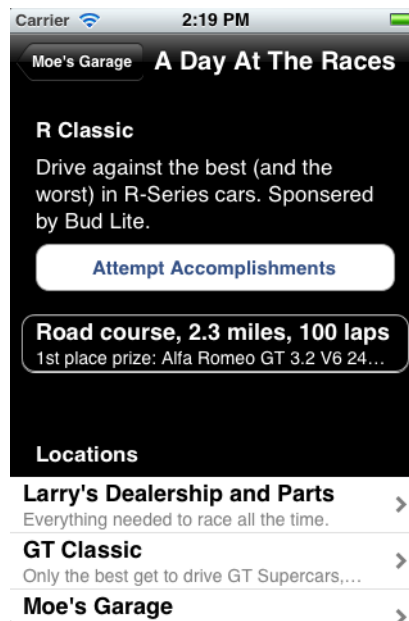


Figure 2: Location with an Accomplishment.

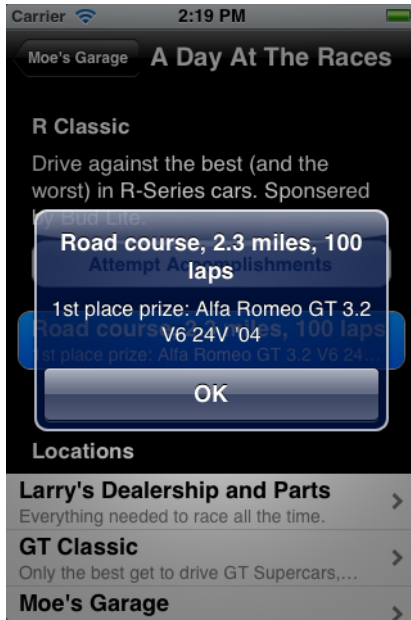


Figure 3: Pressing an Accomplishment will show a detail view.

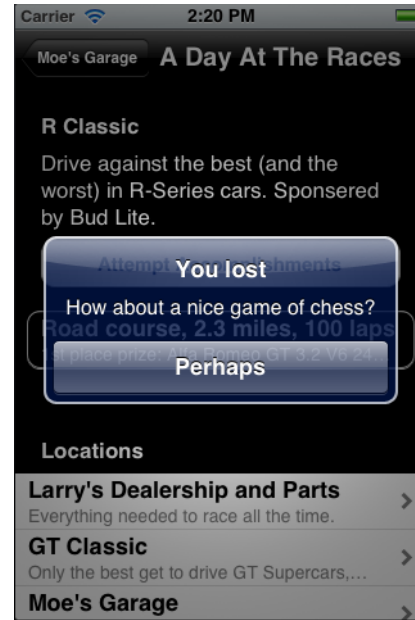


Figure 5: Losing an Accomplishment can have its consequences. Playing chess is but one of them.

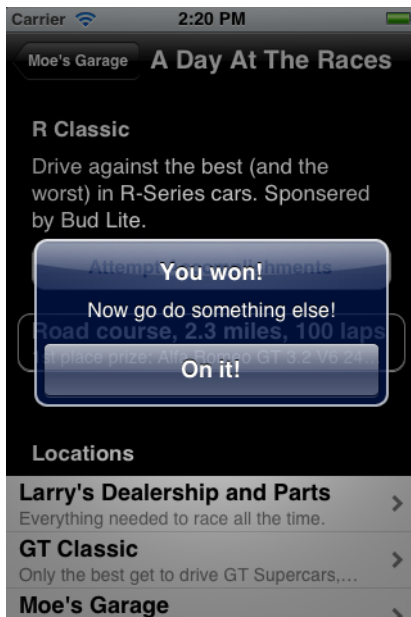


Figure 4: Pressing the “Attempt Accomplishments” button will pit you against the game itself.

### 3 Editing a Game

These data points are where a game can be edited as displayed in the Game Components screen:

- Game Information
- Player Information
- Tasks

When editing any information, it is important to press the Done button if it appears at the bottom of the screen. Otherwise, changes are automatically saved when editing is completed.

#### 3.1 Game Selection

This screen lists all of the games created. New games can be created with the “New” button in blue, and games can be deleted by pressing the “Edit” button. Pressing on a game titles takes you to the Game Mode screen.

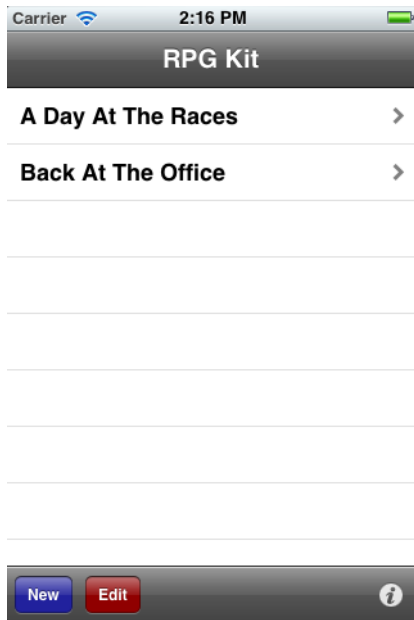


Figure 6: The Game Selection screen is the “home” screen for the app. All navigation ends up here.

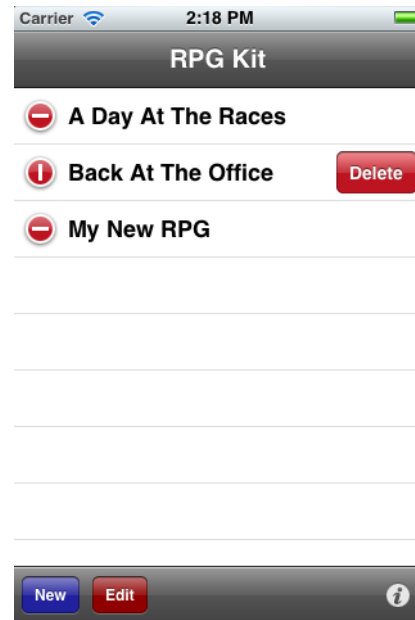


Figure 8: Games, and everything in them, can be deleted with the “Edit” button. Simply press the “Edit” button a second time to cancel without deleting.

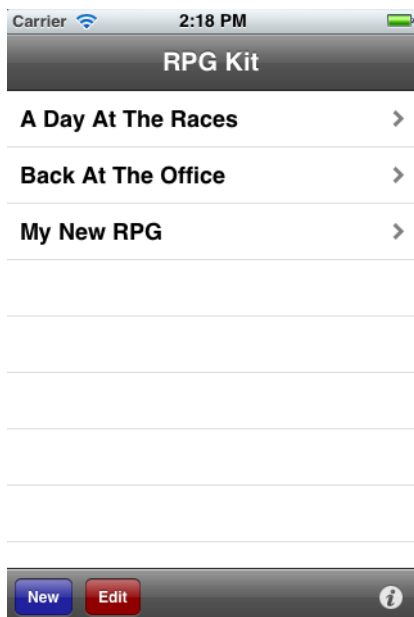


Figure 7: A sample is created by pressing the “New” button.

Pressing the Information icon in the lower-left of the screen takes you to the About screen. At the of there is a button that will reset all of the games back to the defaults. There is no confirmation and undo to this action.

### 3.2 Game Mode

The Game Mode screen is the “home” screen for a game, where you can choose to either play or edit the game. All changes are immediately available during game play. It is not possible to edit a game while it is being played. Pressing “Edit” takes you to the Game Component Selection screen.

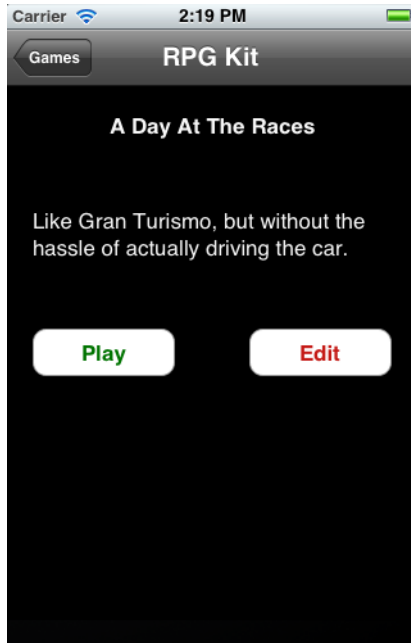


Figure 9: The Game Mode screen is the “home” screen for a game.

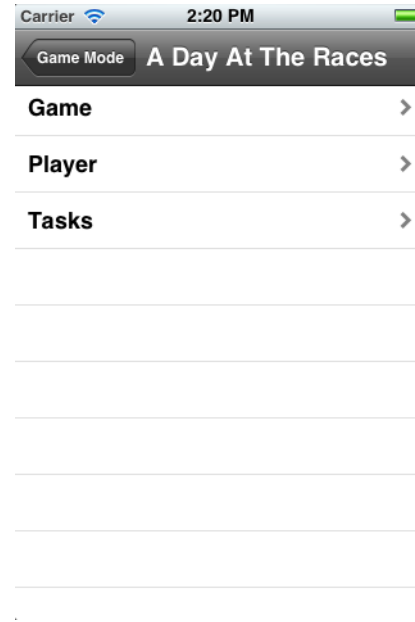


Figure 10: Game Components that can be edited.

### 3.3 Game Component Selection

This screen provides access to the various game elements so that they can be edited. Pressing “Game” and “Player” takes you to their respective metadata editing screens. Pressing “Tasks” takes you to a menu listing all of the Tasks, regardless if they are a Location or Accomplishment, to be then edited.

### 3.4 Game Information

The game information pertains to the world or environment that the game takes place. You can edit the following items:

- Title
- Introduction
- Time Unit—Hour, Day, Week, and Month would be the most common, but any time unit is allowed.
- Money Unit—Dollars, Credits, Gold Pieces are obvious choice, but like Time, any unit is allowed.

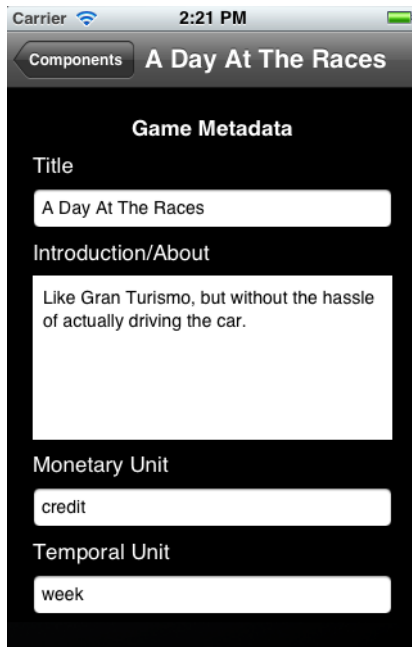


Figure 11: The Game information screen

There is no more or less than one player within a game and they cannot be added or deleted by you.

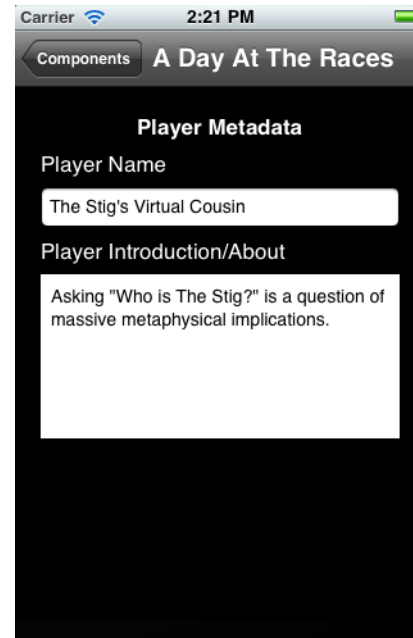


Figure 12: The Player information screen

### 3.5 Player information

The player has these points of information to edit:

- Name
- Bio

### 3.6 Tasks

Tasks are a little complicated but not too much so. A Task is a generic object that can be used as either a Location or Accomplishment. Tasks are linked together according to their usage and are added and deleted via the “Tasks” menu item in the Game Component Selection screen.

### 3.6.1 Editing tasks

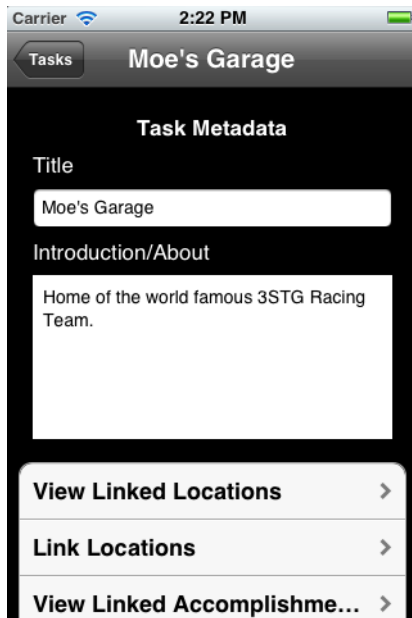


Figure 13: The Task information screen

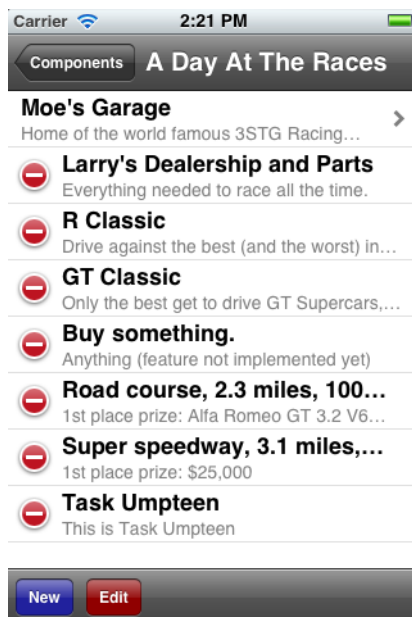


Figure 14: Tasks can be edited and deleted, but they must also be linked according to their use later.

### 3.6.2 Locations

A Task that links to other Tasks are known as Locations. These are Game-level objects that the player can travel to and from. Tasks that are listed as Accomplishments are things the player can do when in a given Location. A Task can link to any number of Accomplishments and Locations.

### 3.6.3 Accomplishments

Accomplishments are types of Tasks that the Player is pitted against at a given Location. Accomplishments have a variety of Attributes that are used to reward or punish a Player based on how well they did at a given Accomplishment. Accomplishments cannot link to other Accomplishments in that when viewing a list of Accomplishments during game play, there is no way to select an Accomplishment to go to another list.

### 3.6.4 Linking Tasks

Tasks can be linked via their respective menus in the Task Metadata screen. There are two separate views for managing linking. “View Linked Locations/Accomplishments” shows those Tasks that are linked as their respective paradigm (sibling vs. child). Tasks can be delinked from this menu by pressing the “Edit” button. “Link Locations/Accomplishments” shows all of the tasks that are contained within the game as a whole.



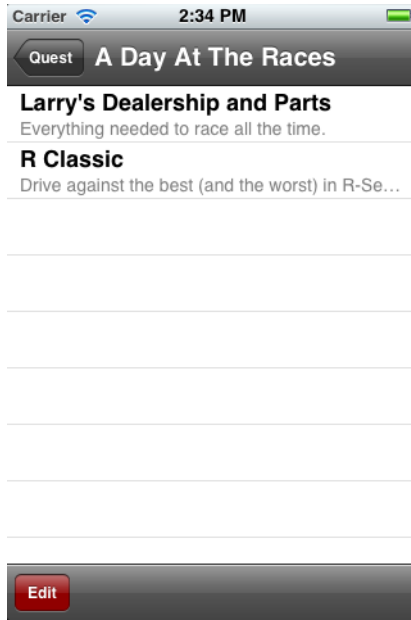


Figure 15: Locations Linked to another Location.

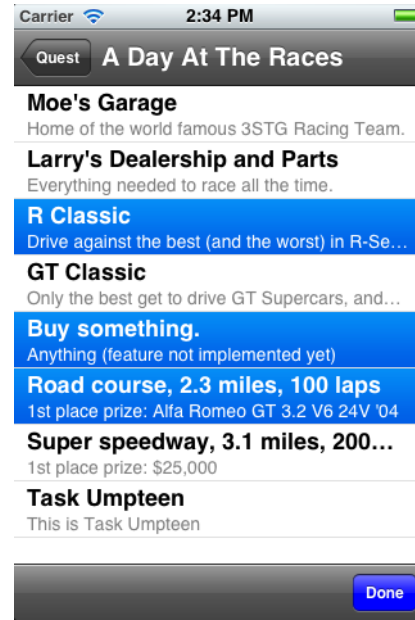


Figure 17: Multiple Locations can be linked at once.

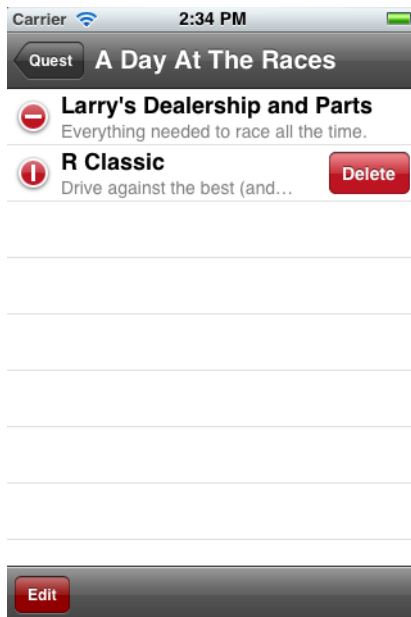


Figure 16: Delinking Locations is a simple matter of deleting from the list, but this does not delete the Location from the game.

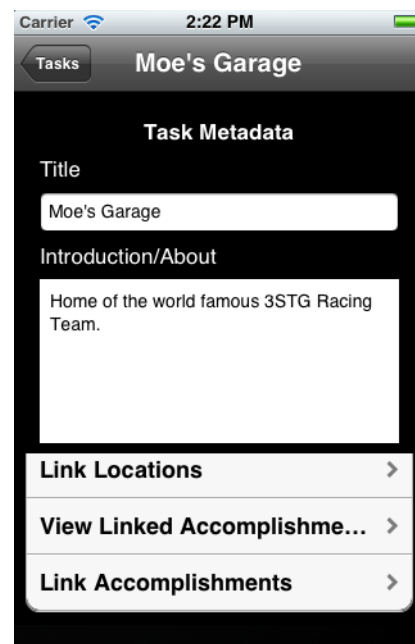


Figure 18: Managing Accomplishments is at the bottom of the Task Information Menu

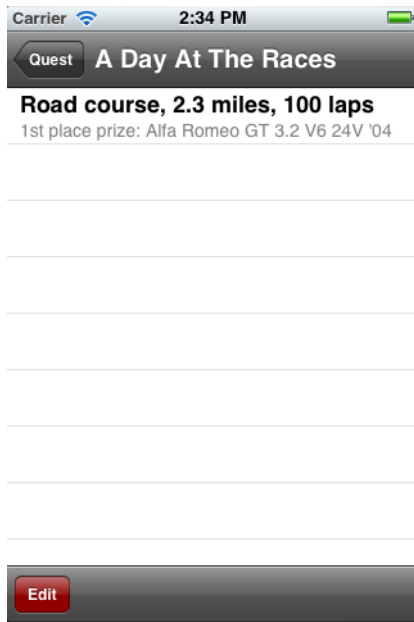


Figure 19: All of the Accomplishments linked to a Location

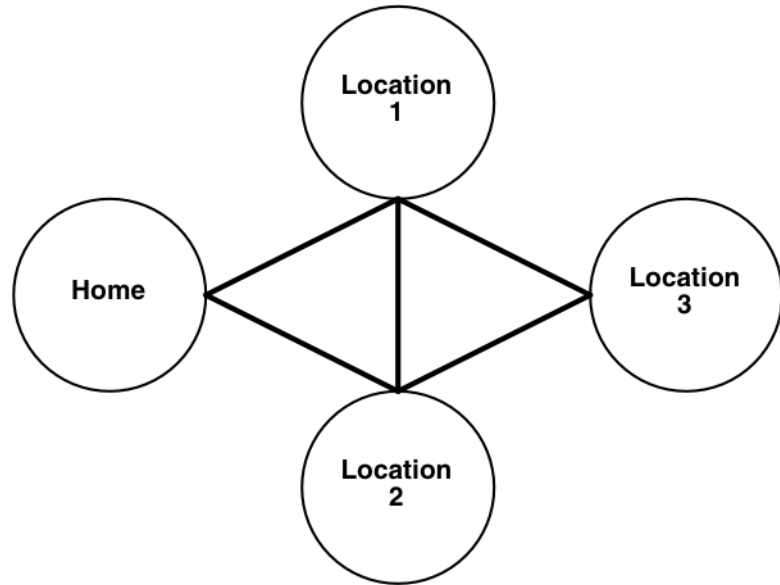


Figure 21: Simple map layout for the Auto Racing and Office games

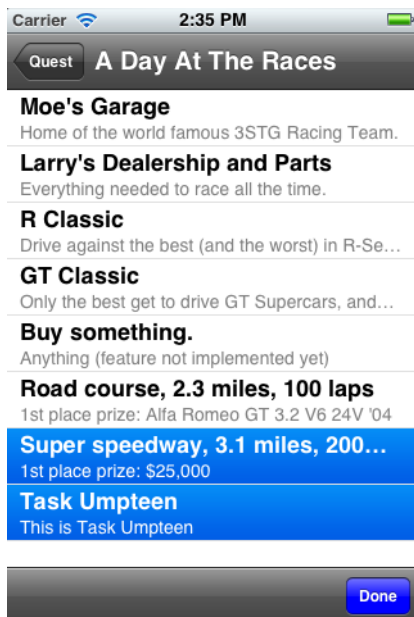


Figure 20: Accomplishments are managed the same way as Locations

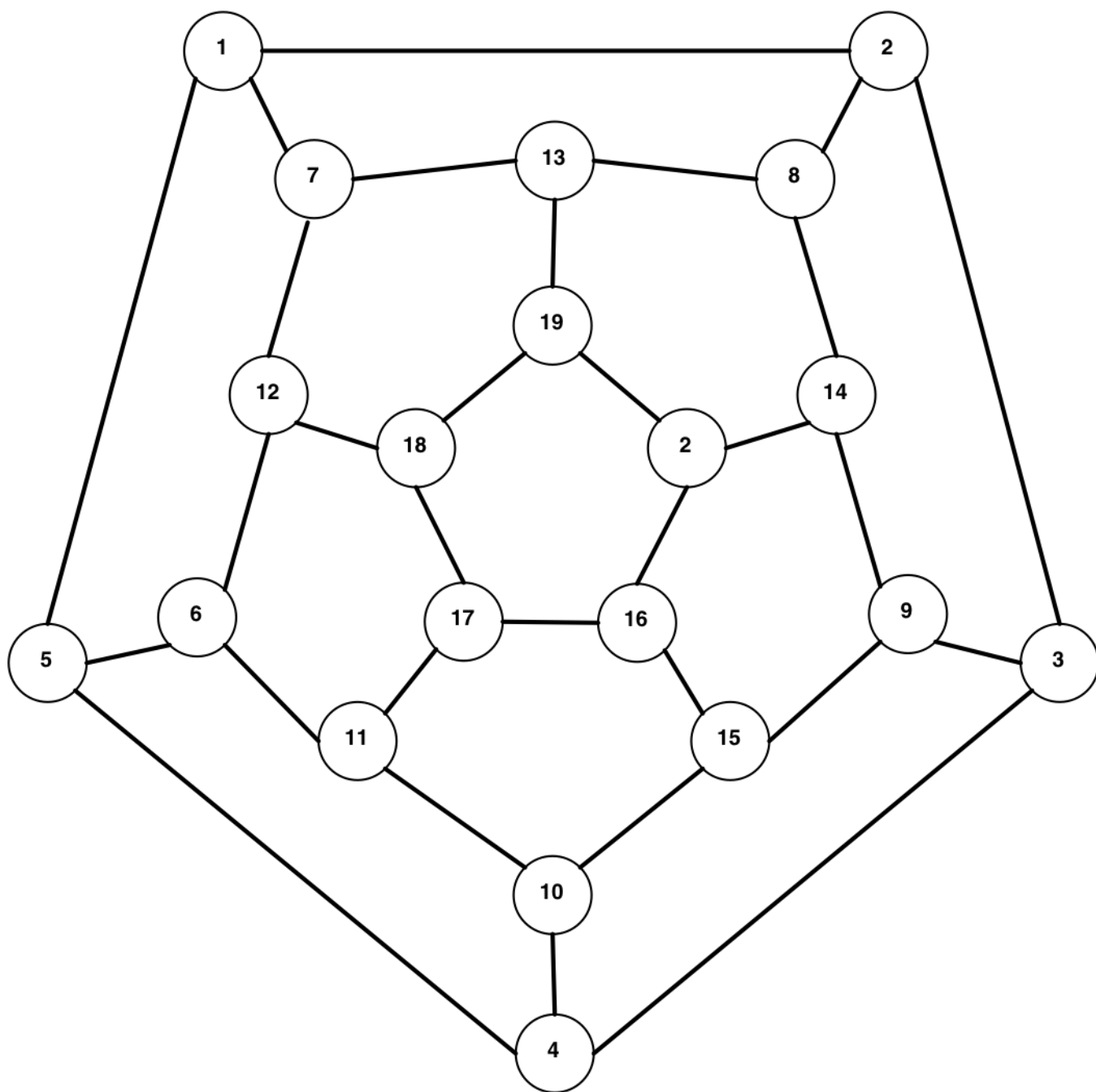


Figure 22: Wumpus map to show how complex linking Locations can become with a bit of effort.

## 4 Technical Manual

### 4.1 Model

RPG Kit’s model is based entirely in Core Data. This offers greater flexibility than SQLite, but retrieving data from the data store is not trivial, so methods to do so are abstracted to private methods within `OCGameDataController` wherever possible.

Regardless of how trivial a task, all view controllers call to `OCGameDataController` for all content management tasks so that they may focus on data display and user interaction management.

### 4.2 View Controllers

Wherever possible, controllers are designed to be as content agnostic as possible.

The `OCObjectLinkerViewController` is the best example of this paradigm. The class is not tied to any specific pair of objects to be linked, but instead is only concerned with an intended (target) parent, and the type of entity to be retrieved and managed. In keeping with the principles of separation of responsibilities, when the user has committed their selection by pressing “done”, `OCObjectLinkerViewController` passes the parent entity and the selected child entities to `OCGameDataController` for the actual task of linking the two. The same principles are applied

for de-linking objects as well.

### 4.3 Data Controllers

#### 4.3.1 `OCGameDataController`

The `OCGameDataController` class lies at the center of the app’s architecture. It provides all of the low-level interaction with the Core Data store and creates all of the sample and starter games for the user.

There are times the view controllers *must* know about how the data is structured not only because they need to populate user interface elements with required game information, but also because the content agnostic classes need to know what entities are being worked with by the user. To help maintain consistency between these disparate classes and the Core Data Store, the `Strings.h` and `Strings.m` files maintain a comprehensive list of `NSString` required for key-values.

#### 4.3.2 `OCGameEngine`

Closely linked to the `GamePlayViewController`, the game engine is critical in game play. It is a “stateless” class, insofar that it is not concerned with the game as a whole, but rather with handling the interaction between a collection of game objects that are needed to perform a task that affects game play.