# SE 3XA3: Module Guide
# DinoDodger

Team 39, S.R.A Squad
Razan Abujarad, abujarar
Shrey Mittal, mittas1
Zhiwen Yang, yangz18

November 10, 2017

# Contents



# List of Tables



# List of Figures

# 1 Introduction

DinoDodger is a redevelopment of the game T-rex Runner by Chromium. In contrast to the original game, DinoDodger is written in Java, only for PC platforms. This program incorporates design principles such as encapsulation, data hiding and modularity. Refer to the MIS for details on the different modules.

The original project was used to determine the requirements for the game along with additional modification unique to DinoDodger. Refer to SRS for documented functional and non-functional requirements.

Note: The following paragraph has been used from the template:

The documentation is organized as follows : Section 2 lists the anticipated and unlikely changes of the software requirements. Section 3 summarizes the module decomposition that was constructed according to the likely changes. Section 4 specifies the connections between the software requirements and the modules. Section 5 gives a detailed description of the modules. Section 6 includes two traceability matrices. One checks the completeness of the design against the requirements provided in the SRS. The other shows the relation between anticipated changes and the modules. Section 7 describes the use relation between modules.

# 2 Anticipated and Unlikely Changes

## 2.1 Anticipated Changes

**AC1:** The platform is only running on PCs

**AC2:** Spacebar as input for the jumping animation

**AC3:** The usage of the Character module

**AC4:** How different should the characters and landscapes be from each other

**AC5:** Implementing the default selection of the character and landscape

Table 1: **Revision History**

| Date | Version | Notes |
|---|---|---|
| 2017/11/10 | 1.0 | MG creation |
| 2017/12/4 | 2.0 | MG Revision |

### 2.2 Unlikely Changes

**UC1:** Sprite animation algorithm

**UC2:** Changing UI library

**UC3:** I/O devices used (Input: Keyboard; Output: Monitor)

**UC4:** Programming language that the program used (JavaFX)

**UC5:** The options given to user at the end of the game (Restart, Play again, Close)

## 3 Module Hierarchy

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in Table 2. The modules listed below, which are leaves in the hierarchy tree, are the modules that will actually be implemented.

**M1:** UI Module

**M2:** DinoDodger Module

**M3:** SpriteAnimation Module

**M4:** PlayScene Module

**M5:** Character Module

**M6:** Pterodactyl Module

**M7:** Cactus Module

**M8:** PointCounter

## 4 Connection Between Requirements and Design

Refer to Table 3.

| Level 1 | Level 2 |
|---|---|
| Hardware-Hiding Module | M1<br>M4<br>M8 |
| Behaviour-Hiding Module | M2<br>M1<br>M4<br>M6<br>M7 |
| Software Decision Module | M5<br>M3 |

Table 2: Module Hierarchy

# 5 Module Decomposition

## 5.1 Hardware Hiding Modules

### 5.1.1 UI Module

**Secrets:** The implementation of the character decision button functionalities as well as theme decision functionalities.

**Services:** Determines which character and theme are chosen by the user when the buttons are clicked.

**Implemented By:** OS

### 5.1.2 PointCounter

**Secrets:** The implementation of the algorithm that determines the score of the user.

**Services:** Counts the users score as long as the game is running.

## 5.2 Behaviour-Hiding Module

### 5.2.1 Character Module

**Secrets:** The implementation of the JUMP method and makes use of the Sprite Animation.

**Services:** Allows the user to make the character jump to avoid obstacles.

#### 5.2.2 Pterodactyl Module

**Secrets:** The implementation of the Pterodactyl obstacle objects.

**Services:** Generates pterodactyl obstacles at random time intervals.

#### 5.2.3 Cactus

**Secrets:** The implementation of the algorithm that generates a random number/size of cacti.

**Services:** Generates randomly sized cacti obstacles at random intervals.

#### 5.2.4 PlayScene Module

**Secrets:** The implementation of the background motion including the ground element, cloud element and cacti elements.

**Services:** Allows the background of the game scene to move in synchronization with the character. This module allows the user to view the obstacles and allows the user to provide a response to overcome the obstacles.

#### 5.2.5 Sprite Animation Module

**Secrets:** The implementation of the character's movement animation i.e the way in which the character moves through the game and the motion of the characters legs.

**Services:** Allows the character to seem to be running such that the user can understand that the character is moving if the game is running.

### 5.3 Input Format Module - UI Module

**Secrets:** The algorithm that converts the buttons clicked by the user to a character selection.

**Services:** Converts the users input from the UI buttons to character selection by reading the button values corresponding to specific characters.

## 5.4 Software Decision Module

### 5.4.1 DinoDodger

**Secrets:** Implementation of the message passing between UI to Animation modules.

**Services:** This module allows the communication between the UI module where the user inputs data and the Animation module where the user views and manipulates the character. This module does not interact with the user but with other modules to bring the game together.

# 6 Traceability Matrix

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

Refer to Section 2 in the SRS for requirements and Section 2.1 for anticipated changes.

| Req. | Modules |
|---|---|
| 2.2.1 | M1 |
| 2.2.2 | M1 |
| 2.2.3 | M1 |
| 2.2.4 | M1, M4 |
| 2.2.5 | M4, M5 |
| 2.2.6 | M4, M1 |
| 2.2.7 | M1 |
| 2.2.8 | M4 |
| 2.2.9 | M4, M5 |
| 2.2.10 | M7 |
| 2.2.11 | M4 |
| 2.2.12 | M4 |
| 2.2.13 | M4 |
| 2.2.14 | M4, M1 |
| 2.2.15 | M4 |
| 2.2.16 | M1 |
| 2.2.17 | M1 |
| 2.2.18 | M1 |
| 2.2.19 | M1 |
| 2.2.20 | M1 |
| 2.2.21 | M8 |
| 2.2.22 | M6 |

Table 3: Trace Between Requirements and Modules

| AC | Modules |
|---|---|
| AC1 | M1 |
| AC2 | M4 |
| AC3 | M5, M4 |
| AC4 | M1, M4 |
| AC5 | M1, M2 |

Table 4: Trace Between Anticipated Changes and Modules

# 7 Use Hierarchy Between Modules

A Uses Hierarchy exists between the modules such that :

- ~~DinoDodger module uses Module and UI module~~
- ~~Animation module uses the Character module~~
- ~~Character module uses the Sprite Animation module.~~
- UI module uses PlayScene module
- UI module uses DinoDodger module
- PlayScene module uses Pterodactyl, Character and Cactus modules
- PlayScene module uses PointsCounter module
- PlayScene module uses DinoDodger module
- Character and Pterodactyl modules use the SpriteAnimation module
- Cactus module uses DinoDodger module

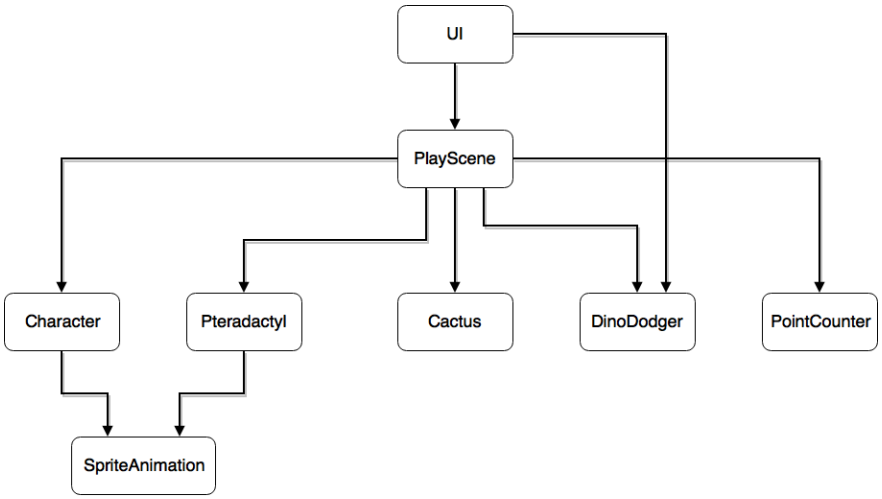


Figure 1: Use hierarchy among modules