SE 3XA3: Design Documentation
Hextris

Team 3, Hextron
Jason Li lij107
Scott Williams willis12
Yousaf Shaheen shaheeny

November 10, 2017

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
Nov 10, 2017	1.0	First Revision (Rev0)
Date 2	1.1	Notes

1 Introduction

This is the module guide for the remake Hextris, an altered version of a open sourced online game that adds elements of tetris to a hexagon based reaction game. This guide is for the explanation of the design as well as the breakdown of the various modules used in the creation of the game. The document is also used to improve updates and fixes breaking down the game into more manageable chunks for future developers and designers to use. It also lends itself to allow for an easier time for validating the requirements set by the SRS in decomposing the projects. The module guide complements the MIS document in providing a better explanation of the separate functions and classes provided by the MIS.

1.1 Overview

The document will begin with the changes as well as the confirmed differences to our project highlighting the new differences and what was kept the same. The document will then list and explain the decomposition of the various modules. The document will then go over the module hierarchy, finishing with a traceability matrix.

2 Anticipated and Unlikely Changes

The following section outlines the changes that were made to the project. Some of these changes are ones we anticipated and some are unlikely changes.

2.1 Anticipated Changes

As the project goes on, the team discover certain aspects of the project that need to be changed in the future. These anticipated changes are all changes that are to be executed by modifying a single module for each change. The following is a list of anticipated changes the team has determined to be necessary for the completion of the project.

AC1: Decreasing the amount of block variations to be added to the final product.

AC2: Development a block destruction algorithm.

AC3: Decrease the scope of some of the features to be added for the user interface.

AC4: Added implementation of a fill algorithm for usage when matching blocks are to be found.

AC5: Added implementation of a fill algorithm for usage when matching blocks are to be found.

2.2 Unlikely Changes

During the implementation and production of the product, certain things problems arise that facilitate the need for a modification to a certain module. These problems can be resolved through modifications to the modules respective to each problem. These problems cause the need for changes that were not expected to be made by the team prior to the implementation.

UC1: Certain animation to the hexagon object will be omitted.

UC2: Physics of the trapezoids will need to be modified to restrict unwanted movement.

UC3: Player controls and menu interface remains unchanged.

UC4: Added background music for a more appealing interface when the game is being played.

UC5: The pause button was changed from P to the Space Bar.

3 Module Hierarchy

The following section is an overview of the modules to be implemented in the final product. The modules are categorized into three categories; Hardware-Hiding Modules, Behaviour-Hiding Modules, and Software Decision Modules. The following is a table detailing the decomposition of the modules into their respective categories.

M1: HexagonBehaviour(Player Controller) - This module dictates the movement and modification of the hexagon game object within the game.

M2: LevelManager - This module manages the boot processes of the game.

M3: MusicPlayer - This module finds and plays the music file while the game is being played.

M4: PauseManager - This module implements the pause functionality of the game.

M5: Spawner - This module generates a trapezoid of a random colour and a random location of the six defined locations to be dropped in the game.

M6: TrapezoidBehaviour - This module handles the main physics of the trapezoids that are to be dropped onto the hexagon object. This includes the motion, scaling, collision, and position of the trapezoids. This module includes two scripts; TrapezoidTransform and Trapezoid-Collision.

M7: Game Interface - This module handles the drawing of the various game objects as well as the various screens. Additionally it handles the frame to frame updates keeping track of time and score. Mostly of the Game interface is embedded and linked with the unity engine.

Level 1	**Level 2**
Hardware-Hiding Module	M2 M7
Behaviour-Hiding Module	M1 M4 M5 M6
Software Decision Module	M3

Table 2: Module Hierarchy

4 Connection Between Requirements and Design

Certain design decisions needed to be made in order to facilitate the requirements outlined in the SRS documentation. For the random trapezoid generation functionality, the team decided to choose between one of four colours and one of six positions for each face of the hexagon. The movement function requirement of the trapezoids was facilitated through setting a velocity vector for each trapezoid pointing to the center of the hexagon. The speed of the falling trapezoids increasing was designed by increasing the magnitude of the velocity vector by a scaling factor that is directly related to the total time the current game is running. The game will support up to eight trapezoids stacked up on each face of the hexagon. A limit was set on the hexagon and is to be checked each time a new block is dropped. If the height of the stack exceeds eight trapezoids, then the game would end. The rotation of the hexagon was is to be controlled by the user. The decision made was that the rotation is to be controlled by the left and right arrow keys on the keyboard. The left arrow key rotates the hexagon by 60 degrees to the left and the right arrow key rotates the hexagon by 60 degrees to the right. For the pause functionality of the game, the team decided to set the time scale value in the code to 0 to simulate the freezing of time in the game. The above requirements and their corresponding design decisions were made by the team to ensure the quality and the completion of the project.

5 Module Decomposition

The following section outlines each module defined above in more detail. The section talks about the secrets and services of each module. It also outlines what the module was implemented by. The section discusses certain aspects of each module that pertains to the functionality of the module in question.

5.1 Hardware Hiding Modules

This section outlines the modules categorized in the Hardware Hiding category.

5.1.1 LevelManager (M2)

Secret: Secret: Loads assets and all files from game src folder.
Services: Boots environment and assests loading the start menu activating all the other scripts to be used when the game is started.

5.1.2 Interface (M7)

Secret: Tracks time for trapezoid generation.
Services: Handles the frame to frame updates keeping track of time and score. Mostly of the Game interface is embedded and linked with the unity engine.

5.2 Behaviour Hiding Modules

This section outlines the modules categorized in the Behaviour Hiding category.

5.2.1 HexagonBehaviour(Player Controller) (M1)

Secret: script that waits for player input to rotate the hexagon by 60 degrees.
Services: Waits for user input to dictate the movement of the hexagon for the player and then rotates the parent hexagon and all of its connected child trapezoids by the 60 degrees as well.

5.2.2 PauseManager (M4)

Secrets: Halts the other operating scripts of the game.
Services: Brings up pause menu and stop other game scripts from continuing

5.2.3 Spawner (M5)

Secret: Creates trapezoids at random with randomized pre-set color.
Services: Creates the blocks for the player to match with fully randomized parameters for color and spawn location. Continues to create trapezoids and increases speed incrementally to increase difficulty.

5.2.4 TrapezoidBehaviour (M6)

Secret: Applies gravity effects on spawned trapezoids and moves them towards the hexagon.
Services: Handles all aspects of the trapezoid blocks in regards to its characteristics. This includes the motion, scaling, collision, and position of the trapezoids. This module includes two scripts; Trapezoid Transform and Trapezoid Collision.

5.3 Software Decision Modules

This section outlines the modules categorized in the Software Decision category.

5.3.1 MusicPlayer (M3)

Secret: Loads background music file.
Services: Loads and plays mp3 background music file and keeps it playing.

6 Traceability Matrix

The following section outlines two traceability matrices. One outlining the Trace Between Requirements and Modules and the other outlining the Trace Between Anticipated Changes and Modules.

Requirements	**Modules**
F1	M5, M6, M7
F2	M2, M6
F3	M2, M5, M6, M7
F4	M1, M6
F5	M1, M7
F6	M4, M2

Table 3: Trace Between Requirements and Modules

Modules	Anticipated and Unlikely Changes
M1	UC3
M2	AC3, UC1
M3	UC4
M4	UC3, UC5
M5	AC1, AC3, UC2
M6	AC1, AC2, AC4, UC2
M7	AC3, AC5, UC3, UC5

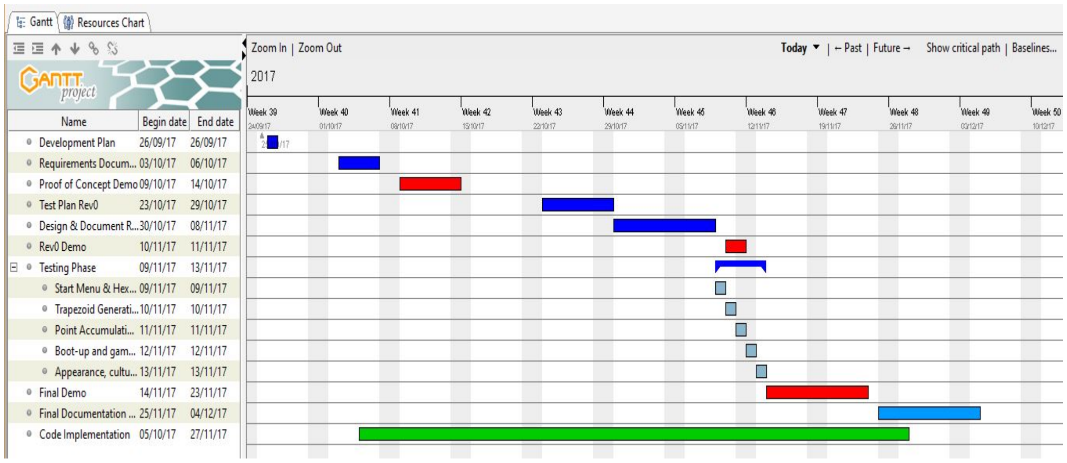
Table 4: Trace Between Anticipated Changes and Modules

7 Use Hierarchy Between Modules

Use Hierarchy between Modules: This section explains some of the interaction between the various modules. Starting with the level manager which is responsible for the startup assets of the game and the instantiation of most of the other modules. The Hexagon behaviour, Trapezoid behaviour, Game interface and Music player is then used. The spawner is non dependent on most of the other modules, continuously creating blocks until the game is paused or the game is stopped. Trapezoids generated are taken by the trapezoid behaviour which then scales and applies gravity to the blocks. Colliding with the player, the player control dictates the movement of the center hexagon. All the modules operate under the game interface which is unitys platform for frame updates and time tracking. Music player is also an independent module that provides the background music regardless the processes of the other modules.

8 Schedule

The following is the Gantt Chart schedule of the project.



Name	Begin date	End date
Development Plan	26/09/17	26/09/17
Requirements Docum...	03/10/17	06/10/17
Proof of Concept Demo	09/10/17	14/10/17
Test Plan Rev0	23/10/17	29/10/17
Design & Document R...	30/10/17	08/11/17
Rev0 Demo	10/11/17	11/11/17
Testing Phase	09/11/17	13/11/17
Start Menu & Hex...	09/11/17	09/11/17
Trapezoid Generati...	10/11/17	10/11/17
Point Accumulati...	11/11/17	11/11/17
Boot-up and gam...	12/11/17	12/11/17
Appearance, cultu...	13/11/17	13/11/17
Final Demo	14/11/17	23/11/17
Final Documentation ...	25/11/17	04/12/17
Code Implementation	05/10/17	27/11/17

Figure 1: Gantt Chart