Computer Science and Engineering

University of Nevada, Reno

Shape Fitter

Team 12

Ryan Devaney

Sanya Gupta

Instructors:

Dr. Sergiu Dascalu

Devrin Lee

Advisors:

Vinh Le

March 30th, 2018

Table of Contents

Abstract
The purpose of this document is to the layout the specifications and design details regarding the implementation of Shape Fitter. Shape Fitter is a game developed through Unity that makes users cover a background circle with other differently sized circles. If the background is still showing that means the user has not placed the other circles correctly and therefore the level is not complete yet. Each level will consist of a timer that will vary as the levels progress. The user must accurately place the circles under the timer to advance to the next level.

Recent Project Changes
No recent project changes have been made since the concept is very new yet solid.

Specification

Functional Requirements

FR ID #	Level	Description
FR_01	1	The game will have multiple levels for the user to play. (Minimum of 10).
FR_02	1	The game will incorporate a timer for each level.
FR_03	1	The game will have a home screen with access to the game and manual.
FR_04	1	Each level will have a timer which is increased by 20 seconds for each level, with the user having 1 minutes to complete the first level.
FR_05	1	Each level will have an "undo" which undos the placement of the last circle and can be used only 3 times per level.
FR_06	1	When a circle is placed, there is some indication that the background circle is covered for the user to see.
FR_07	3	The game will let users log in to an account to keep track of which level they are on with corresponding best time per level.
FR_08	2	The game will have a leaderboard for each level so other users can see what the best time is.
FR_09	1	The game should give some indication that the background is all covered before timer ends.
FR_10	1	Users will place a limited amount of circles to cover a larger circle.

Non-functional Requirements

NFR ID #	Level	Description
NFR_01	1	The game needs to be hosted on either a website or mobile application.
NFR_02	2	The game's home page should load in under 15 seconds.
NFR_03	1	The game should recognize mouse input or touch input.
NFR_04	3	The game will have licensable ambient music playing in the background.

Use Cases

Fig. 1 Use Case Diagram that shows how the user interacts with the game.

Detailed Use Cases

Use Case Name /ID #	Description
Registration/UC_01	User is able to register account and create username and password.
Login/UC_02	User is able to log in to the game with username and password.
Circle Placement/UC_03	User is able to place circle with a mouse click/touch and circle is place accordingly to where user clicked/touch.
UNDO button/UC_04	User is able to UNDO placement of last placed circle upto 3 times per level by clicking an UNDO button.
Between Levels/UC_05	User should see leaderboard or proceed to next level if placement of circles is correct.
Background Opacity/UC_06	User should see an indication of the background circle being covered when a circle is placed, depending on placement of circle.
Navigation/UC_07	User should be able to navigate from homepage to the game manual and back to the home page.

Detailed Templates

Use case name/ID	UNDO button/UC_04
Actor(s)	User playing the game
Precondition(s)	User is in a level and has not used the UNDO button
Flow of Events	1. User is in a level 2. User clicks the UNDO button
Postcondition(s)	User is able to successfully undo the placement of the last circle upto 3 times.

Use Case Name/ID	Circle Placement/UC_03
Actor(s)	User with access to game
Precondition(s)	User is in a level and has to place circle.

Flow of Event(s)	1. User is able to drag the circle around before dropping it to be placed
	2. User lets go of click or stops touching the circle
Postcondition(s)	Circle is placed where user clicked/touched.

Requirement Traceability Matrix

Req.	UC_01	UC_02	UC_03	UC_04	UC_05	UC_06	UC_07
FR_01							
FR_02							
FR_03							X
FR_04							
FR_05				X			
FR_06						X	
FR_07	X	X					
FR_08					X		
FR_09							
FR_10			X				

Design

Scripts

Here are some of the scripts that the game will be using. I'm not entirely sure if this is acceptable for High-level and medium-level design.

PlayerController. This script will be in control of the player. This script will place a circle when the user clicks/taps their phone. This script will also be in controls of how many circles the user can place as well as the size of the circle. This script will only be used in scenes in which the user is actively playing the game. In scene such as the menu this will not be present.

GameController. This script will control how the game is going to operate. It will check win conditions or lose conditions for the player. It will also transition the game from scene to scene based on user input. This script will also deal with the timer and passing information for the leaderboard.

MenuController. This script will control the main menu. It will be used to navigate the user through various options based on user input. The menu will move the player to scenes that start the game in which the GameController script will take over.

More scripts will have to be added as more features are added into the game, but for now this is what I have done.

UI Screenshots

Fig 2. Shows a screenshot of the game where three circles have been placed.

Fig 3. Shows a screenshot where two circles have been placed. These circles are smaller than the ones from the previous screenshot.

The team is still working on graphics and UI as most of the focus has been on just getting the game working. More screenshots could be added but it would essentially be the same thing of just various sizes and amounts of circles.

Updated Glossary of Terms
- Unity
  - A game development platform and engine that is being used to build this game.
- Leaderboard
  - A scoreboard that displays names and current scores of the leading competitors.
- Drag and Drop
  - Move (a selected item) to another part of the screen using a mouse or similar device.
- User Interface (UI)
  - The industrial design field of human–computer interaction, is the space where interactions between humans and machines occur.
- User Experience (UX)
  - The process of enhancing user satisfaction with a product by improving the usability, accessibility provided in the interaction with the product.
- C#
  - C# is an object oriented programming language created by Microsoft and similar to Java. C# will be used to code the project through Unity.
- Timer
  - An automatic mechanism for activating a device at a present time.

Engineering Standards and/or Technologies
Unified Modeling Language (UML)
The Unified Modeling Language is a standard modeling language, as it helps software engineers understand hardware and software designs related to their project. This allows software engineers from different disciplines to understand the diagrams such as component diagrams, sequence diagrams, etc. For the Shape Fitter Project, UML diagrams were created in order to represent the software components.

Doxygen
Doxygen is a tool that generates documentation by using specialized commenting. The team will incorporate doxygen style comments in their project for every method. Every time a member updates a method or file, the method should be updated to represent new logic and the file header version should also be updated and mentioned the last author.

Github
Github is an online and command line version control system. It is used to monitor changes made to a project as well as host different branches and copies of the project. The team will be

using Github to backup the source code as well as a tool to share changes. Additionally, Github will be used to host the project website.

Slack

Slack is an online, cloud-based collaboration tool. The team uses Slack to communicate with each other regarding project ideas and documentation. Slack is used to primarily for the members to communicate what has been done, what needs to be done, and what impediments stand in the way so each member knows the progress of the project.

Unity

Unity is a game engine that is used by various game agencies to create two or three dimensional games. While Unity provides some inbuilt functions, developers can also write their own functions in C#. The team is using Unity in the development on Shape Fitter. Basically, all the functionalities of Shape Fitter will be coded and developed in Unity.

Updated List of References
Project Domain Book
Unity in Action
Author: Joe Hawking
This book teaches developers how to go about creating games in Unity. It is primarily meant for people who have had prior coding knowledge in languages such as C++/C as games created in Unity use C#. This is the perfect book for the team as they are all experienced programmers and have also prior game design and creation knowledge with other another game engine called Gamemaker and with one member having some experience in Unity itself.
Reference Articles
1. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.4790&rep=rep1&type=pdf
User Interfaces for All: New Perspective into Human-Computer Interaction
This scholarly article outlines how one should go about designing a product that is heavily focused on human-computer interaction so the final product is enjoyable and efficient for the user to use. This is a helpful paper for Shape Fitter, as it relies on the experience of a strong human-computer interaction to create a fun yet challenging atmosphere. This article helps the developers and UX designers create a wonderful and engaging product so it will not fail in the market and continue to flourish.
Project Related Websites
https://unity3d.com/learn/tutorials/topics/2d-game-creation/2d-game-development-walkthrough
Unity 2-D Walkthrough
The website above is a great resource for members new to Unity. It has a link to a video which gives a nice overview of some of the tools and functionalities that Unity provides for its developers. Likewise, there are other links to other specific features such as sprite creation and

rendering. Members should consult this website if something is challenging in terms of implementing as they can gain an idea through some of the tutorials provided.

Contributions

Ryan Devaney

- High and Medium Level Design
- Traceability Matrix
- Use Case Diagram
- Updated UI

Total: 6 hours

Sanya Gupta

- Functional and Nonfunctional Requirements
- Use Cases and Detailed Templates
- Glossary
- References
- Engineering Standards and Technologies

Total: 6 hours