

University of Alabama in Huntsville

LOUIS

Honors Capstone Projects and Theses

Honors College

11-30-2020

The Domino Effect

Adam Russell McFry

Follow this and additional works at: <https://louis.uah.edu/honors-capstones>

Recommended Citation

McFry, Adam Russell, "The Domino Effect" (2020). *Honors Capstone Projects and Theses*. 120.
<https://louis.uah.edu/honors-capstones/120>

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.

The Domino Effect

by

Adam Russell McFry

An Honors Capstone

submitted in partial fulfillment of the requirements

for the Honors Diploma

to

The Honors College

of

The University of Alabama in Huntsville

December 6, 2020

Honors Capstone Director: Professor Mary Allen

Professor of Computer Science



Student (signature)

November 29, 2020

Date



Director (signature)

Date



Department Chair

Department Chair (signature)

Date: 2020.11.30

12:54:33 -06'00'

Date

Honors College Dean (signature)

Date



Honors College
Frank Franz Hall
+1 (256) 824-6450 (voice)
+1 (256) 824-7339 (fax)
honors@uah.edu

Honors Thesis Copyright Permission

This form must be signed by the student and submitted as a bound part of the thesis.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Adam McFry

Student Name (printed)

Adam McFry

Student Signature

11/29/20

Date

The Domino Effect

by

Adam McFry

December 6, 2020

HON 499

TABLE OF CONTENTS

SUMMARY	3
BACKGROUND INFORMATION.....	3
PLAYING A GAME OF DOMINOS	3
THE DOMINO EFFECT	3
PROPOSED DEVELOPMENT TIMELINE.....	4
DEVELOPMENT PHASES.....	8
PREPARATION AND TRAINING	8
PROJECT FRAMEWORK IMPLEMENTATION	8
USER INTERFACE IMPLEMENTATION	8
SORTING ALGORITHM IMPLEMENTATION	9
USABILITY TESTING AND UNPLANNED SUPPORT	11
REFLECTIONS AND CONCLUSION	18
REFERENCES.....	19

SUMMARY

The Domino Effect is a mobile application that allows users to sort a collection of dominos. The purpose of this project was to both gain experience in iOS application development and also provide a real-world example for implementing a backtracking algorithm. TDE was developed in five phases:

1. Preparation and Training
2. Project Framework Implementation
3. User Interface (UI) Implementation
4. Sorting Algorithm Implementation
5. Usability Testing and Unplanned Support

BACKGROUND INFORMATION

PLAYING A GAME OF DOMINOS

A game of dominos starts with all domino tiles face down. The players will shuffle the dominos, then distribute them equally among each player. The player with the highest double (a domino with each side having the same value) will place this domino in the center of the table face up. Each player must then make a “train” of dominos using the dominos that they were dealt. A “train” is a series of dominos that are connected such that like numbers touch. (Ex: If the head value is a three, and the player has a domino with the values three and four, they can only connect to dominos that also have a number four.)

Players’ trains must start with the “head” number as indicated by the double in the center of the table. The goal for each player is to sort their hand into a “train” that leaves them with the least number of points at the end of the round. A round of the game is over when a player plays all of the dominos in their hand. When the round ends, the dominos are shuffled again, and the next highest double domino is placed in the middle of the table and the next round begins. The game ends when the round of double zero concludes.

THE DOMINO EFFECT

The Domino Effect (TDE) is an application that is intended to be a companion to a player during a round of dominos. Since the goal each round is to be left with the fewest points in your hand, it is crucial to have the most optimized “train” sorted before play begins. TDE allows users to input the dominos that are in their hand, select the head domino (the double in the center of the table), then sort. TDE then displays the best path for the user to play that round.

PROPOSED DEVELOPMENT TIMELINE

1. Preparation and Training (**Expected Completion: August 31, 2020**)
 - a. Description
 - i. Before beginning the design and implementation of The Domino Effect (TDE), there is necessary training for iOS Development. This includes the following areas:
 1. Swift
 2. XCode
 3. Apple Developer Profile Setup
 - b. Details of the Task
 - i. The primary source for the training for this project comes from <https://academy.codewithchris.com/>
 1. The iOS Foundations course is designed to introduce the basics of Swift coding, how to set up, build, and deploy iOS applications on iPhone and iPad.
 2. This course introduces several design patterns and their implementations Xcode. Such design patterns include:
 - a. Model View Controller (MVC)
 - b. Protocols and Delegates
 3. This course highlights the basics of designing and formatting applications. Topics covered in this area include:
 - a. Main Storyboards
 - b. LaunchScreen Storyboards
 - c. UI Elements (labels, buttons, stack views)
 - d. UI Formatting (constraints, Safe Area, Device Class Formatting)
 4. This course also introduces the distinction between Swift and Cocoa Touch classes and appropriate use cases for each.
 - c. Definition of Done
 - i. Once the training is complete, there should be several sample applications that have been designed, built, and tested. These apps include:
 1. War Card Game
 2. Match App
 3. Quiz App
 4. News App

2. Project Fundamentals (***Expected Completion: August 31, 2020***)
 - a. Description
 - i. To begin this project, there needs to be an Xcode project created and a GitHub repository set up to manage the project throughout the development process.
 - b. Details of the Task
 - i. Create Xcode Project
 - ii. Create and link GitHub repository
 - c. Definition of Done
 - i. Xcode Project Exists
 - ii. GitHub repository exists and is linked to the Xcode project

3. UI Implementation (***Expected Completion: September 30, 2020***)
 - a. Description
 - i. This phase of the project will include designing the UI elements and navigation path for the application. The appropriate storyboards and links should be in place to navigate between them. Basic operational logic should be in place such that a happy glad path can be navigated without the implementation of the actual sorting of dominos.
 - b. Details of the Task
 - i. Graphics and UI Elements
 1. Finalize buttons, title screen, app icon, and domino sprites. Having the dominos at this stage is not necessary but would be desired.
 2. LaunchScreen storyboard should have whatever animation or imagery that will be in the final version of the application.
 3. Formatting for various device sizes should also be accomplished here. The constraints for each class should be completed here. This includes the title screen image placement and button placement throughout the application.
 4. Navigation animations should be completed. Settings menu, volume controls, and help menu implementation will happen at this stage.
 5. Decide in a method to display the dominos once they are sorted. (List, stack view, actual board, etc.)
 - c. Definition of Done
 - i. A user should be able to run the application and do the following:
 1. Access all menus from the title screen buttons including:
 - a. Volume
 - b. Settings
 - c. Help
 2. Tap to Begin

3. Reach each storyboard/ screen as if the sorting implementation were done. (The screens might be blank at some places, but that is expected; implementation will come in the next phase.)
 4. Graphics are finalized.
4. Sorting Algorithm Implementation (**Expected Completion: October 31, 2020**)
- a. Description
 - i. This phase of the project provides the application with the logic behind how the dominos are sorted. This will also be the phase in which the dominos are actually displayed for the user.
 - b. Details of the Task
 - i. Algorithm Analysis.
 1. Using research and knowledge from CS 317, consider what techniques and sorting algorithm formats might best fit this use case. Consider the algorithm in terms of big O notation and decide on how this might affect TDE.
 2. For this instance, the sorting algorithm is intended to function such that dominoes are placed in order by like components to form a “train”. (Ex: [3|6] -> [6|2] -> [2|4], etc)
 3. This application is intended to provide the most efficient line of dominoes so that the user is left with the fewest remaining points (dominoes that are not included in the sorted train).
 - ii. Algorithm Implementation
 1. Implement the algorithm. Make sure it works on sample data (the dominos are sorted appropriately). Make sure that the results are repeatable for different sets of data.
 - a. Finalize the format of the “Domino” data type (class members and format)
 - b. Finalize the DominoManager class and its member functions including:
 - i. Filtering the dominos based on user selection (ignore duplicates that may be selected)
 - ii. Users must have a way to select what the “head” of the train is. (i.e. if the head of the train needs to be a 3, then the user needs a way to specify that.)
 - iii. Sorting the dominos and returning an array of dominos in the order that is specified from the user.
 2. The screen should display the dominos in sorted order. Once the algorithm is written (domino model), the Domino

ViewController should be implemented to display this information.

- c. Definition of Done
 - i. Users should now be able to play a complete happy glad path of the application. Such a path should include:
 1. Tap to begin
 2. Select the dominos to sort
 3. Tap sort
 4. Display the correctly sorted dominos
 5. Exit
5. Testing and Unplanned Support (***Expected Completion: November 30, 2020***)
- a. Description
 - i. This section should be the final stage of the application development process. Once the previous functionality is finished, this time is dedicated to discovering any errors that might exist and polishing up the application for publication.
 - b. Details of the Task
 - i. Test with many different sets of data.
 - ii. Test the app with uncommon use cases outside of the happy glad path.
 - iii. Allow others to test the app and acquire feedback of the design.
 - c. Definition of Done
 - i. The application runs as expected and there remain no bugs for a Version 1 release.
6. Project Delivery (***Expected Completion: December 6, 2020***)
- a. Description
 - i. Once the development of TDE is complete, the paperwork as defined by the UAH Honors College should be complete and the application can be presented. If time allows, TDE will be published to the Apple App Store.
 - b. Details of the Task
 - i. Complete the application.
 - ii. Complete the paperwork.
 - c. Definition of Done
 - i. Honors Capstone is completed, submitted, reviewed by the personnel of the UAH Honors College, and approved.

DEVELOPMENT PHASES

PREPARATION AND TRAINING

Before development work could begin on The Domino Effect, I spent several months training and building several sample iOS applications in Xcode. The primary source for training was provided by CodeWithChris, an online learning community that offers classes on iOS development. These trainings provided tutorials for learning the basics in iOS development, including how to appropriately use design patterns such as Model View Controller (MVC) and Protocols and Delegates. In addition to these design patterns, these classes also provided insight on how to interact with and manipulate UI storyboards with Xcode. These modules also covered how to use several essential UI elements for iOS development including buttons, labels, images, stack views, and collection views. The tutorials resulted in the creation of several sample applications. The applications created included a quiz app, a card matching app, a news app, and a card game app. These applications provided the base level knowledge that was used later in the creation of TDE.

PROJECT FRAMEWORK IMPLEMENTATION

This phase of development was where the project files and GitHub repository for The Domino Effect were created. GitHub was the optimal choice for version control for this project, as it integrates very well with Xcode. During this phase, sound effects and background music were created for TDE using GarageBand.

USER INTERFACE IMPLEMENTATION

The User Interface design was the most labor-intensive portion of this project. The graphics for buttons, logos, and icons were commissioned from a local graphic design freelancer, Ashlee Jones. While the graphics were being finalized, the screen layouts were implemented, and the supporting navigation code was added. This phase of the project was the foundation of the application since it included completing all of the segues and menus for the application.

Each screen that the user interacts with is a view controller. For each of these view controllers, there is a supporting Swift class that includes all of its variables, functions, and transition information. The functions for sorting and displaying the final list of dominos were left as TODO items that would be implemented in the next phase.

Each domino in TDE is based on a Domino struct object that contains the basic attributes for a domino including the head number, the tail number, a Boolean value to

indicate whether or not the domino has been selected by the user, and the image name that the domino will display for the user. A DominoManager class was created to hold the list of all of the dominos. This class is static because TDE only ever needs the one instance of DominoManager to keep up with the details of the dominos.

Additionally, the sound manager was implemented in this phase. Because the background music and sound effects needed to be accessed from every view controller, the sound manager was also implemented as a static class. This allowed any of the view controllers to access a single instance of the sound player without creating separate players for each screen.

SORTING ALGORITHM IMPLEMENTATION

The sorting capability for TDE is the main functionality of the app. This phase of development is the last requirement to make the app function as intended. For this, there were several days of research to better understand how the sorting should behave. The first step was to play a round of dominos and detail how a user manually sorts the dominos. To optimize the best list of dominos, a decision tree was used to visualize every possible path to take.

MANUAL SORTING

The sample hand of dominos that was initially test included the following dominos:

- [9|8]
- [8|1]
- [5|7]
- [5|6]
- [6|4]
- [5|3]
- [2|5]
- [9|3]
- [9|5]

The head number for this testing was 9. The user sorted the dominos in their hand into every possible path combination. The decision tree that was created based on these paths is shown in the figure below.

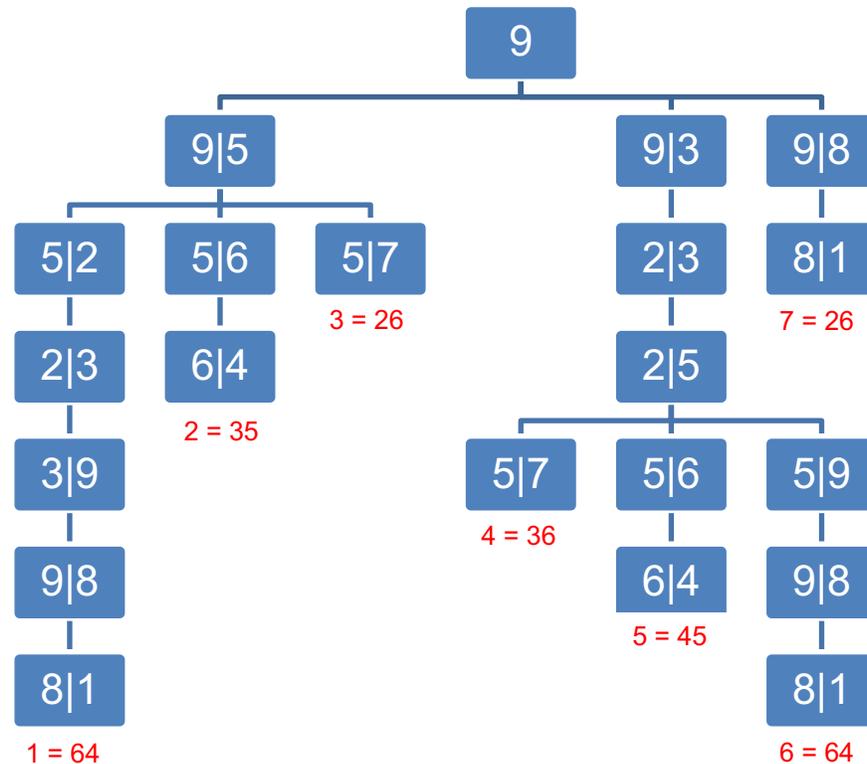


Figure 1 (Domino Decision Tree)

For each path in the decision tree, the path index and sum of the path are displayed in red. For this sample set, there are seven different paths that the user can take. A path ends when there are no remaining dominos that can connect to the end. (For instance, the paths ending in 1 are terminated at this point because there are no remaining dominos that contain a 1 that can be appended to it.) In this particular example, there are two best paths, since two paths contain the highest point value of 64. Since the goal of TDE is to give the user the best option, the first path will be presented in the event of a tie.

ALGORITHM TRANSLATION

Once the manual way of sorting dominos was recorded, this needed to be translated to the algorithm that will sort the dominos in TDE. Since the sorting process involves finding all possible paths, a backtracking approach was used. Backtracking is an algorithmic technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time (GeeksforGeeks).

For this backtracking use case, the algorithm will be given a list of dominos selected by the user and the head number that leads the train. The algorithm will then find all the dominos that contain this head number and create a new list with these dominos. If this list is not empty, which indicates that there are still dominos to add to the path, the sort function will call itself and sort with this new list. Since the sort function continues to call itself until it reaches a domino that has no matches, it is considered to be recursive. Once the end of a path is reached, the algorithm records this path and backtracks until it finds another path or has exhausted all possible options.

When the sorting is complete, and the list of paths is found, the total of each path is calculated. The path that has the greatest total will be displayed to the user.

USABILITY TESTING AND UNPLANNED SUPPORT

TEST DELIVERY

The usability test was delivered via Apple's Test Flight beta testing software. The users were sent a link to their iPhones where they were able to install Test Flight and download the latest beta version of The Domino Effect. Once installed, the users were sent the usability test, and they were given a thirty-minute time limit to complete as many of the tasks as possible.

TEST AUDIENCE

Six users were chosen to test The Domino Effect. Their characteristics are listed in the table below.

First Name	Last Name	Age	iPhone Model	Familiar with Dominos?
Alex	McFry	25	11 Pro Max	Yes
Brooke	McFry	50	7 Plus	Yes
Ashlee	Jones	28	11	No
Amber	Lai Hipp	32	12 Pro	No
Breanna	Hunt	23	XR	No
Tegan	Guillebeau	24	11 Pro Max	No

Figure 1 (Usability Test Audience)

USABILITY TEST DETAILS

For the usability test, the users were given a list of ten tasks to complete. Tasks 1-9 provided explicit instructions for the user to follow. These tasks collectively formed the base path of execution for this application. Task 10 instructed the user to use the app without instruction or a specific goal. The purpose of this task was to gather feedback about how users might use the TDE beyond the intended scope. The complete list of tasks is listed in Figure 2.

Along with each task, users were also given details about what the expected results were. If the user received a different result, they were told to provide details about what happened. This was particularly helpful for task 10, as the users were able to send specific details about crash reports, provide step-by-step information about what they did if a crash did occur, and they were also able to describe how they prefer to use TDE. The main goal for this testing was to determine whether or not users were able to successfully use TDE, gather information about any errors that exist, and determine if the application was ready to be deployed to the general public.

#	Task	Expected Result from TDE
1	Tap Help Button.	Help menu displays without error.
2	Tap Sound Button.	Sound stops without error.
3	Record a list of at least two dominos that you want to have sorted. If you require assistance, a photo of a sample hand of dominos will be provided.	n/a
4	Tap Start, then select a Head Domino.	The picker view should allow the selection of a lead number without error. The lead number should be displayed in the appropriate box without error.
5	Select the list of dominos to be sorted.	The user should be able to select the dominos without error. Dominos should change shading to indicate that they are selected.
6	Tap Sort.	List of dominos should be given to the user.
7	Record the list of sorted dominos that is returned on screen.	This list should be the most efficient path for the user to play the game with.
8	Verify that the dominos line up sequentially such that the tail of one domino matches the head of the next.	Dominos should be displayed properly.
9	Tap restart.	User should be able to return to the selection screen without error.
10	Test edge cases and interact with the application however you would like.	Varies. Ideally, there are no crashes or errors, but users should report any significant findings.

Figure 2 (Usability Test Tasks)

DATA COLLECTED

For this usability test, the data that was collected included: crash reports from the users while using TDE, questions or comments for each task in the usability test, and the duration to complete the usability test.

TEST RESULTS

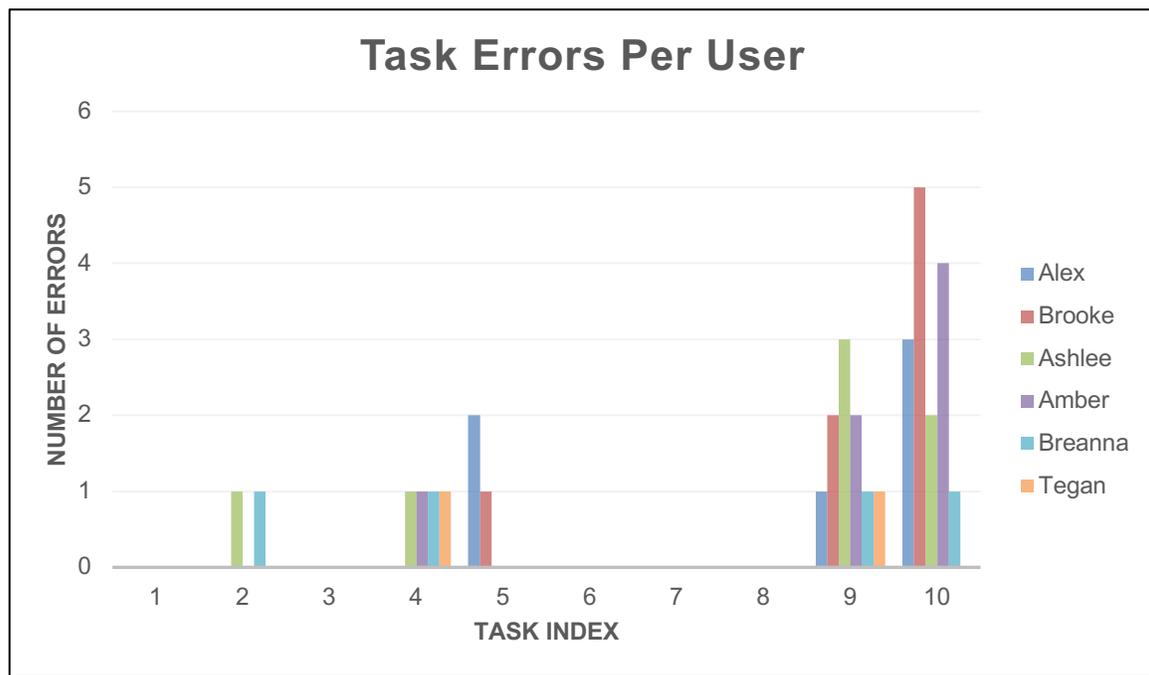


Figure 3 (Task Errors Per User)

When the usability tests were collected for each user, the data was then broken down by task. For each task, the number of errors that the user reported per task was recorded in Figure 3. In this instance, an error is classified by any result that a user receives that does not match the expected outcome. Multiple errors are indicative of the user experiences more problems than the specific task failing. It was discovered that none of the users experienced errors for tasks 1, 3, 6, 7, or 8. Tasks 2, 4, 5, and 9 show several errors, and task 10 had the highest ratio of errors of all of the tasks.

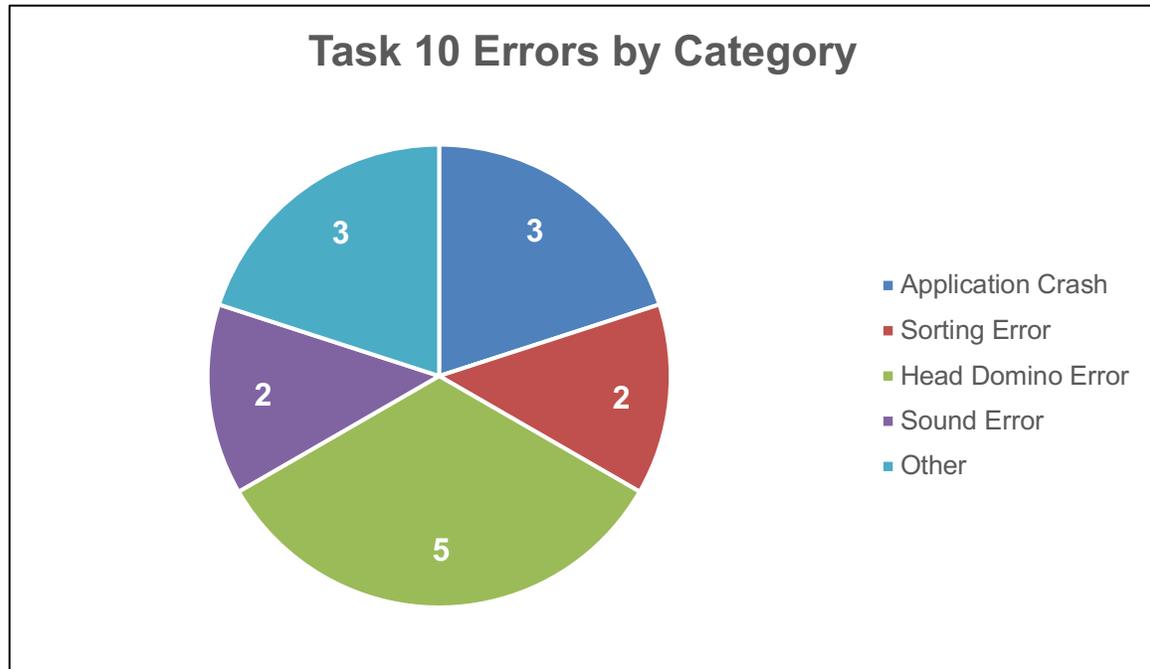


Figure 4 (Task 10 Errors by Category)

Since task 10 showed such a high amount of error reports, further analysis was done on the task feedback to understand the categories of errors that were reported. The results were divided into five categories:

1. **application crash** (the app closes and prompts the user to submit a report)
2. **sorting error** (the sort button shows the user an unexpected value)
3. **Head Domino error** (the Head Domino selection does not work as expected)
4. **sound error** (the background music and sound effects does not work properly)
5. **other** (any error that does not fall into the previous categories)

The errors that arose for task 10 were somewhat equally distributed, though there were more problems with the Head Domino than any other component.

FINDINGS AND DATA ANALYSIS

For the tasks given to the users in the usability test, five tasks showed erroneous results when the users attempted them. This section of the report seeks to give a more in-depth analysis of each task users had errors with and discuss what these errors indicate for the future development of TDE.

Task 2, the sound test, failed for two users. Each user reported that there was no sound when TDE launched, and the button for turning the audio on and off had no effect.

Task 4, selecting a head domino, failed for four users. Three of the users reported that the display window for the head domino number was initially populated when the main menu appeared. The other user that reported the error noted that, while the head domino box was empty on the main menu, it would not let them select a number at all. After restarting their device, this user noted that everything then worked as expected.

Task 5, the domino selection task, failed for two users. Both users indicated that when they selected a domino, the shading did not change. They each also noted this only happened with dominos that had a 6 in it. Alex reported that there were two errors here, noting that the sound did not work for selecting these dominos either. There was no report of any other number of dominos exhibiting this behavior.

Task 9, the reset task, presented errors for each user. Alex, Breanna, and Tegan all reported that the reset button did allow the user to go back to the main screen, but the sound was not working when they tapped the button. While this error did not indicate an issue with the reset functionality, they deemed it noticeable enough to note in their reports. Both Brooke and Amber experienced this issue as well, and they both also reported that the background music stopped when they were taken back to the main menu. Lastly, Ashlee received all of the previous errors in addition to not being able to go back to the main menu at all. After closing the app and opening it again, she was able to complete this task.

For task 10, the users were asked to use the app without instruction. The intention of this task was to allow users to use the application as if they had downloaded it from the App Store with no prior instructions. This task provided the most insight into problems that exist in this beta version of TDE. As shown in Figure 3, each user except for Tegan was able to produce at least one additional error during this task. Breanna and Ashlee were able to produce errors here as well, but they are in line with errors that they previously reported in other tasks. Breanna reported sound issues again with the reset button, and Ashlee reported the same issue as well as not being able to hear the background music after returning to the main menu.

For Amber, Brooke, and Alex, there was a pattern in their chosen paths for this step that resulted in a multitude of errors. Each user made a point to reset after they sorted a list of dominos and then proceeded to exit out to the title screen of the app. From here, they tapped the start button, then attempted to select more dominos at a rapid pace. For each

user, the application crashed while they were attempting to select dominos. They each sent in crash reports that detailed these steps. All three of these users also reported issues with the Head Domino value not displaying correctly. Brooke and Alex also noted that while sorting some dominos, the application presented a set that included dominos that they had not selected.

The other miscellaneous errors that occurred on task 10 included the dominos suddenly missing from the selection window, the reset button had changed font size, and the help button stopped working after approximately 6 taps.

RECOMMENDATIONS

Based on the results from all six users in the usability test, it is clear that the sound mechanism is not functioning as it should. There should be a more thorough investigation into the sound manager of TDE to better understand why the sound will spontaneously stop working for certain buttons.

Upon further inspection of the sorting errors, it was determined that this was a result of the Head Domino value being improperly reset. This also caused the other Head Domino errors that were reported. The reset mechanism in TDE should be reconfigured so that these errors do not persist in future versions of TDE.

Another area of concern is that the application crashes that were reported. Three separate users were able to crash TDE, though it still is unclear what is causing the crash. It appears that something is wrong when the users try to select dominos too quickly, so this process should be retested in order to more accurately understand what is causing the issues.

As a whole, the results of the usability test have been very informative. As the developer of the application, these error reports and user notes are crucial in making sure that these issues do not arise once this application is published to the App Store. It is often hard to think outside of the test cases I typically use as a developer, and I have a good idea about what areas I need to repair based on this feedback. While it is a good thing that the actual sorting algorithm appears to be functioning as intended, a few values are not being assigned properly, and there are a few areas of the user interface that need to be redesigned.

REFLECTIONS AND CONCLUSION

Based on the results of the usability test, there are several bugs that have been fixed in the latest version of TDE. The head domino value is now being reset appropriately, the sound errors have been fixed, and the sorting function is working as intended. A new beta version of The Domino Effect has been published to Test Flight, and a new round of testing is currently in progress. The crash reports are also still being analyzed, as there is not much conclusive evidence as to what common step triggers a crash. Once these errors are more thoroughly investigated, TDE will be published to the App Store.

Upon completion of this beta version of The Domino Effect, it is very apparent how ambitious the initial development timeline was. The algorithm research and implementation ended being completed two weeks later than originally planned. The UI design also took longer to complete, as there were several times throughout the process that new elements needed to be added or other elements needed to be removed entirely. While there was a significant portion of time spent at the beginning of this project for training and preparation, there did need to be more time planning specifically for this application. Doing so would have prevented the time loss from having to fix the UI.

As a whole, the making of The Domino Effect has been very informative. The entire development process has given me a better understanding of how an iOS application is developed from start to finish. This endeavor has also given me valuable experience in developing custom sorting algorithms and using backtracking logic. Even though there were errors and stumbling blocks throughout the process, each obstacle was a learning opportunity for what not to do in the future. As a first-time app builder and aspiring iOS developer, this project has proven to be a great introduction to the process, and it has also confirmed that iOS development is a passion that I want to continue pursuing in my career as a software engineer.

A demo video demo of the TDE is available here:

https://drive.google.com/file/d/1zVEjJotlZ2zjThaZwhVM_wPpmykg7e8O/view

The source code is available here:

<https://github.com/armcfry/TheDominoEffect>

REFERENCES

- Affairs, Assistant Secretary for Public. "Reporting Usability Test Results." *Usability.gov*, Department of Health and Human Services, 9 Oct. 2013, www.usability.gov/how-to-and-tools/methods/reporting-usability-test-results.html.
- "Backtracking Algorithms." *GeeksforGeeks*, www.geeksforgeeks.org/backtracking-algorithms/.
- Ching, Chris. "CodeWithChris." *The Leader in IOS Foundations Training*, 21 June 2020, codewithchris.com/.
- "Distributing Your App for Beta Testing and Releases." *Apple Developer Documentation*, Apple, 2020, developer.apple.com/documentation/xcode/distributing_your_app_for_beta_testing_and_releases.
- Isherwood, Matt. "How to Write a User Testing Report That People Will Actually Read." *Medium*, UX Collective, 19 July 2018, uxdesign.cc/how-to-write-a-user-testing-report-that-people-will-actually-read-652d15d2f92e.
- Jones, Ashlee. *ASHLEE JONES MARKETING*, ashleejonesdesigner.squarespace.com/.
- Markel, Michael H. *Practical Strategies for Technical Communication: A Brief Guide*. 3rd ed., Bedford/St. Martin's, 2019.