



# DISPLAY

## Requirements Specification

Version 1.3

Prepared by:  
James Treneman

Prepared for:  
Steve Sheehy  
CS 401 Capstone

# 1. Table of Contents

- 1. Table of Contents..... 2
- 2. Introduction ..... 2
- 3. Objectives ..... 2
- 4. Scope ..... 3
- 5. User Requirements ..... 4
- 6. System Requirements..... 5
  - A. Title Screen ..... 5
  - B. Team Select Screen ..... 5
  - C. Battle Screen ..... 6
- 7. Non-Functional Requirements..... 8
- 8. Glossary ..... 9

## 2. Introduction

This specification document details DISPLAY’s user experience by defining its scope and requirements for program response and performance.

## 3. Objectives

DISPLAY’s main objective is to provide value to the player. In other words, it must be fun to play. It strives to reach this goal by delivering bright fireworks animations dictated by a simple strategy and skill check. The visuals, strategy, and skill check must all work in concert to deliver value, a.k.a. fun. DISPLAY is designed to inhabit the casual puzzle game genre. It was conceived and produced to demonstrate Lab Cat Games’ ability to execute a small project. Its secondary objective is to generate revenue by running advertisements.

### 3.1 Provide a fun, valuable experience

- 3.2 Bright fireworks animations
- 3.3 Simple strategy and skill check
- 3.4 Casual puzzle genre
- 3.5 Generate revenue through advertising

## 4. Scope

DISPLAY is a small, lightweight game with a scope to match. The biggest impacts are the multiple platforms it's planned to release on. Handling a cross-platform represents a significant boost to project complexity. Each platform needs version control, validation principles, and pipeline development. DISPLAY is initially playable as a .exe and HTML5 game, published on itch.io. With a secondary goal of publishing on Android and IOS app stores, DISPLAY requires more testing and hardware to undergo QA/QC than a single-platform product. Additionally, a dynamic monetization strategy requires analysis along extra dimensions when publishing cross-platform. The game design and production itself does need a significant amount of iteration but its premise and game flow are concise, with everything taking place on the menu and battle screen.

This document uses **story points** to help discuss timelines and quantify human resource requirements. For DISPLAY, story points are estimated to be four hours of one producer's time. Story points are not meant to be taken as literal work hours, but as guidelines to track progress with. The project's total story point tally is estimated to be around 200. With completing a prototype and releasing the first alpha build accounting for half of the total, DISPLAY is a 3-5 month endeavor for a single developer.

- 4.1 Release as HTML5 game
- 4.2 Release as mobile game
- 4.3 Monetize through advertising
- 4.4 Iterate until quality project emerges
- 4.5 Story points equivalent to ~4 developer hours
- 4.6 Projected ~200 story points to complete

## 5. User Requirements

Enjoyers of DISPLAY find value in simplicity and ease-of-use. The heavily constrained nature of player interactions appeals to DISPLAY's casual audience. The user experience is simple for elegance's sake, but also for reliability.

5.1 Deploy seamlessly, requiring minimal setup which is one reason HTML5 is the main target platform.

5.2 Players interact with Display by clicking the left mouse button (or tapping the touchscreen) and by pressing, holding, and releasing during battle.

5.3 Battle procedure can also be executed using the spacebar.

5.4 Random user inputs will not impact the experience; filtering most of the input stream out ensures player intent is the only signal reaching the game state.

5.5 Players expect a smooth-running game; DISPLAY runs at 60 frames per second to meet this requirement. Due to the screen-filling fireworks animations with high frame counts, dips in performance are noticeable if the system is taxed. However, sprites are low resolution and can be pre-fetched while players navigate the select menu to make sure there are no visible drops in frame rate during battle. Other than loading the animations, DISPLAY has a tiny resource profile, suitable for legacy devices up to 10 years old.

5.6 DISPLAY has one layer of progression: unlocking rival teams. This requires DISPLAY to keep persistent data and does so with a local save txt file. It also means that users need an easy way to reset their progress if desired. That functionality, along with a sound toggle are available through the settings menu screen.

## 6. System Requirements

DISPLAY's game flow proceeds as follows:

- 6.1 Players begin on the **Title** screen, navigate to the **Select** screen, then to the **Battle** screen.
- 6.2 After completing a battle, it's back to **Select** to choose their next match.
- 6.3 All screens feature a single button header menu used to change settings or close the game.
- 6.4 DISPLAY is designed specifically to avoid unintended inputs impacting gameplay, so it should never flow without user intent. Much of this is accomplished by turning off input checking for transition periods and other potentially ambiguous moments.
- 6.5 Screen-by-screen system requirements:

### A. Title Screen

**Active inputs:** left mouse click, touch screen tap, spacebar press.

**Outcomes:**

All of these inputs advance the game to the team select screen, playing a short jingle. Users are then locked out from any interaction for a couple seconds to avoid unintended selections on the next screen.

### B. Team Select Screen

**Active inputs:** Left mouse click, touch screen tap.

**Outcomes:**

- 1) The 'player' box is highlighted, and clicking an unlocked team button fills in this box, indicating the player has selected it for themselves.
- 2) Next, the 'rival' box gets highlighted, and clicking a team button will either add it to the rival box or do nothing (if locked).
- 3) After both teams are selected, the battle button lights up and animates, inviting players to begin when ready. Clicking any of the 'boxes' or buttons on this screen plays a sound effect, with a locked button being negatory and an active one sounding affirmative.

- 4) The Battle button lights up and animates. When clicked, the game plays a jingle and flows to the battle screen. It also locks out player interaction for a couple seconds to avoid accidental inputs.

**Note 1)** Once the player has chosen their team, they may click either the 'player' or 'rival' box (which highlights it) to change a selection.

**Note 2)** A fresh playthrough of the game begins with only one team selectable to play as (player team), and one to play against (rival team). The locking system ensures that players cannot battle against the same team they have chosen for themselves. So, once a player chooses their own team, that team is locked out when selecting their rival. **EXAMPLE:** Team DAZZL cannot battle against rival team DAZZL.

**Note 3)** Unlocks occur when the player beats a new team in battle. After victory, the defeated rival becomes an option during player selection and the next team on the list unlocks as a rival. **EXAMPLE:** If team STARZ is unlocked but hasn't been defeated yet, the player cannot select STARZ as their own team, but they can and should understand to select STARZ as their current rival. Once the player beats STARZ in battle, it unlocks for player selection and the next previously locked team on the list, say team FLASH, becomes available as a rival.

## C. Battle Screen

**Active inputs:** Left mouse click/release, touch screen tap/release, space bar press/release.

**Outcomes:**

- 1) Rival launches their fireworks first. During the rival's turn, the player cannot use the launch button (which is grayed out to convey its disabled status) but are free to select from any of the fireworks in inventory by clicking one of the 5 inventory buttons. This choice determines what type, color, and power level fireworks will be launched on the player's turn. A few seconds after the rival's shot explodes, the launch button lights up and enables, indicating that it is now the player's turn.
- 2) Once a shot is in the air, it gets 'judged' and icons appear on the team's corresponding side of the screen to show how many points have been earned. Initially, a shot is only judged for accuracy and power; final comparison happens after the player responds with their own fireworks. Rivals have a 'critical strike' chance to get the accuracy bonus, and each

- successive rival has a higher chance, ramping up difficulty as the player progresses.
- 3) The player must select their fireworks of choice by clicking one of the 5 inventory buttons (or do nothing if they are content with their current selection), and then press the launch button.
  - 4) Pressing the launch button initiates the launch procedure, bringing up the launch meter. The meter is a stacked series of colored bars that grow and shrink vertically. There is a larger target on the top bar of the meter that the player needs to try and time their release for. If they manage to nail the top bar (5% accuracy) their fireworks receive an extra point. However accurately the player times it, upon releasing the launch button their shot will be fired into the sky like a flare, exploding after a short ballistic arc.
  - 5) The rival and player have both launched their shots, and final judgement occurs. The last comparisons for color and shot type happen. The shot type comparison is dictated by the rock > paper > scissors relationship (mega > spinner > multi in DISPLAY). Responding to the rival's shot with the correct counter earns the player an extra point. If the types are the same no point is awarded to either team. The color point goes to the player if they can avoid matching the rival's shot color, but if the colors do match, it's the rival's point. The team with the most points wins the exchange and earns a star coin.
  - 6) After an exchange, the sequence starts back over at step 1 (rival launches), repeating until all 15 shots have been launched by each team, concluding the battle. The match winner is determined by the team with the most star coins.
  - 7) The battle ends and transitions back to B) Team Select Screen, where any new unlocks become accessible. Input is blocked for a couple seconds during the transition to avoid accidental inputs.

**Note 1)** Audio: a variety of sound effects play throughout the battle stage. Every press of a valid button generates a negative/affirmative sound depending on the button's state. The meter itself has an initial startup sound and a looping charging sound. Executing a perfect launch plays an extra chime to convey success. The fireworks launch itself has a cannon firing initial sound, and fireworks exploding secondary sound. Each point added during judging plays a pop sound and earning a star coin results in a treasure sound effect. Completing the match triggers a success or fail jingle to signify its end result.

**Note 2)** DISPLAY follows a metering principle during the battle sequence. There is roughly a 1 second minimum time between any specific action taking place or information been displayed. The intent is for the game to flow at a relaxing rhythm and build excitement through incremental reveals. This gives players

time to savor information updates and ponder their situation each time.

**EXAMPLE:** Points are tallied up one by one, with a second between each, even though they are calculated instantly.

## 7. Non-Functional Requirements

Guidelines that dictate DISPLAY's production objectives and outcomes.

- A) **Footprint:** Being lightweight, DISPLAY is built to stay under a self-imposed 50 mb cap on project size.
- B) **Execution Speed:** DISPLAY plays at 60 frames per second. Any device unable to maintain that speed should be outside of the 10 year legacy support window.
- C) **Usability:** DISPLAY is designed for maximum usability; one finger is all that a player needs to play through the entire game.
- D) **Accessibility:** LAB CAT GAMES always designs for colorblind players and avoids unfavorable green/orange or green/blue matchups related to game cues. Color based details also differentiate themselves with value and not hue alone. DISPLAY has minimal text, prioritizing iconography and other universal methods of conveyance. Players never have to read more than one or two words at a time, if any. This makes localization a snap for international audiences.
- E) **Security Restrictions:** DISPLAY's main concern is plagiarism and utilizes style and comment signatures in every code asset in case of blatant copying. Its original art is protected through copywrite and the same goes for sound assets.
- F) **Organizational - Programming:** Gamemaker Language (GML) is used to produce DISPLAY, which is akin to JavaScript (syntax, dynamic typing, high abstraction, compiled).
- G) **Organizational – Environment:** GML is the language used within the Gamemaker Studio 2 IDE and DISPLAY is developed using version 2023.8.2.108. Art assets are crafted in GIMP 2.10.8, and sound effects are generated with Audacity 2.3.3.
- H) **External Considerations:** Advertising vendors will need to be vetted and tested to ensure ethical content is being show to DISPLAY users. Any locations that DISPLAY is published will need LAB CAT GAMES basic financial information to share revenue, and it's important they are vetted and reputable.



## 8. Glossary

**Fireworks/Shot:** The main resource that gets used throughout a DISPLAY battle. A team starts with 15 shots total (5 slots with 3 rounds each). Each shot fired gets launched into the air and explodes like fireworks. Strategic decision making allows a player to counter their rival's shot with one that will be judged more favorably.

**Launch Button:** Large oval button near the bottom and in the center of the battle screen. When red, the player can press it to initiate the launch sequence. It turns grey to indicate that it is disabled during the rival's turn or if the player doesn't have any munitions selected (no ammo).

**Launch Meter:** Large vertical stack of colorful bars that grows and shrinks in the center of the screen. Activated by pressing and holding the launch button, it will grow and shrink continuously until released. At the very top is the target bar, and if the player times it perfectly they will release at the maximum height, earning an extra point. Their accuracy with the meter also informs the shot trajectory; poorly timed launches result in badly aimed shots that aren't high or centered.

**Rival:** The player's opponent. The 'computer' controlled enemy that always launches the first shot. Each rival has a unique fireworks inventory, requiring dynamic player strategy. Their shot selection is randomized across their inventory. Defeated rivals can be selected as the player's own team, providing access to different shot inventories.

**Skill Check:** While holding the launch button a meter appears and increments up and down. Players must utilize their timing skill to earn an extra point by releasing the button at its zenith.

**Story Point:** Estimated to be half a day's work (4 hours) for one developer.

**Strategy:** The fireworks types have a rock > paper > scissors relationship that goes mega > spinner > multi. The player can earn an extra point by deploying the correct counter to the rival's shot type.

**Inventory:** The 5 fireworks options that each team has access to. Each of the 5 slots starts with 3 rounds that can be fired before going empty. Each shot has a power level (1-3), shot type (mega > spinner > multi), and color (blue, orange, red).