**ColorSpot!**

**Team Unity**

**Kenji Avila**

**Pamela Camacho**

**Donghao Feng**

**Mohammed Kasim Panjri**

**Marzia Stanekzai**

**Vincent Tran**

**CPSC 362-01**

**Sara Ghadami**

Table of Contents

|  |  |
| --- | --- |
| Section | Pages |
| Document Revision History Table | 2 |
| Project Plan | 3 |
| User Stories (Functional) | 4 |
| User Stories (Non-Functional) | 5 |
| Use-Cases | 6 |
| Use-Case Diagram | 9 |
| Sprint Backlogs | 10 |
| Pre-Game Planning | 11 |
| First Iteration Staging | 12 |
| Development Process | 13 |
| User Manual | 14 |
| References | 20 |

Document Revision History Table

1. 3/9/19 - First iteration of document drafted

**Project Plan**

For our project, our team will develop a fully built videogame designed to be both educational and fun to play. The project has currently been given the working title of ColorSpot, and it will be developed with the aid of the Unity engine. We will follow modern game design principles and industry standards, while also incorporating our own unique ideas. We aim to create a fully functional product that could be capable of competing on the market.

The team has estimated that the project will take majority of the semester to develop, and we are prepared to take on the development in phases. Each development phase will span roughly three weeks with a unique milestone to be achieved in each. The major milestones as of now are: creating a working alpha product, polishing the alpha product to our standards, and working on platform mobility (such as migrating the game to mobile). These milestones will allow us to track our progress in an organized manner and adjust the development process accordingly.

The core of our game is based on the “catch and fall” style of gaming that has been seen throughout time. The unique element to our game will come from its incorporation of color design principles into the gameplay to serve both as an educational tool and to increase the degree of challenge. This entails teaching the player about ideas such as the color wheel, primary colors, monochrome colors, etc.… These details will then affect how we design levels, the UI, the UX, sprites, screens, and other foundations of our product.

This venture serves as a challenge to all members of our group. Most of the group has little to no experience using the Unity engine and only some experience with the coding languages associated with Unity. Although we have already begun familiarizing ourselves with our elected tools, we do understand the risks of not being masters of our tools. To ensure progress, we have agreed to set aside time to both learn more about our tools and to openly communicate about issues we face during development.

Regardless of directly applicable experience, the individuals of our team bring their own skill sets that will prove useful during this project. The team members include: Kenji Avila, Pamela Camacho, Donghao Feng, Kasim Panjri, Marzia Stanekzai, and Vincent Tran. The roles of each member have not been explicitly designated at this time, but with further familiarization our roles will be distributed as project needs arise. We aim to both develop our skill sets and have as enjoyable of an experience as possible.

**User Stories (Functional)**

1. As a player, I want a high score per level. So that I can see improvement and have goals.
2. As a player, I want shop items. So that I can use my coins in a gratifying way.
3. As a student, I want a color wheel that is easily accessible. So I can actually learn my color design conveniently.
4. As a player, I want sound control. So that I can not have loud sound all the time.
5. As a player, I want a interchangeable sprite. So that I can have a personalized avatar.

**User Stories (Non-Functional)**

1. As a player, I want game stability. So that the game will run without any risk of bugs or crashing.
2. As a player, I want data correctness. So that features such as my high score will be saved correctly.
3. As a player, I want system compliance. So that the game does not face copyright issues.
4. As a user, I want good resource utilization. So that the game works well on low functioning technology.
5. As a user, I want modularity. So that the game works both on different gaming platforms.

**Use-Cases**

Use Case: Reviewing High Scores

ID: UC-01

Description: Player can see scores they have achieved compared to other scores they have received.

Primary Actor: The player of the game.

Pre-Condition: The player has played the stage at least once.

Post-Condition: The player can see their highest score of a stage.

Failure end conditions: There are no scores to be seen if the stage has not been completed.

Main Success Scenario:

1. Player is at menu screen

2. Player goes to the Level Selector screen

3. Player is able to view their high score per stage played.

Extensions:

None

Use Case: Buying from the Store

ID: UC-02

Description: Player is able to use coins they accumulate while playing to customize the game through items from a store.

Primary Actor: The player.

Pre-Condition: Player has sufficient coins to buy an item

Post-Condition: Player can customize the game to their liking

Failure end conditions: Insufficient coins means you cannot use this feature

Main Success Scenario:

1. Player is on main screen

2. Player hits button for “Store”

3. Player is taken to page with features for sale with coin price for each one

4. Player selects the feature they can afford.

5. Sales subtract from total coins player has

6. Feature added to their gaming experience.

Extensions:

None

Use Case: Reviewing the Color Wheel

ID: UC-03

Description: Player can access a color wheel to help them memorize color patterns.

Primary Actor: The player

Pre-Condition: Player has access to game

Post-Condition: Player accessing page to color wheel

Failure end conditions: None

Main Success Scenario:

1. Player is on main menu.

2. Player goes to the Level Selector screen

2. Player selects the color wheel

3. Player is taken to screen with a color wheel where they can study it at their leisure

Extensions:

None

Use Case: Controlling Sound

ID: UC-04

Description: Player has control over the sound

Primary Actor: The player

Pre-Condition: Player has access to the game

Post-Condition: Sound is adjusted to desired volume of the player’s choosing

Failure end conditions: Player does not change sound

Main Success Scenario:

1. Player is on main screen

2. Player selects Settings

3. Player selects the sound option at the Settings screen

4. Player can adjust the sound to their desired volume

Extensions:

None

Use Case: Customizing my Sprite

ID: UC-05

Description: Player has option of changing personalized avatar

Primary Actor: Player

Pre-Condition: Player has sufficient coins to purchase a change

Post-Condition: Player has a new avatar

Failure end conditions: Not enough coins to purchase a change

Main Success Scenario:

1. Player is on main screen

2. Player hits button for “Store”

3. Player is taken to page with avatars for sale with coin price for each one

4. Player selects the avatar they can afford

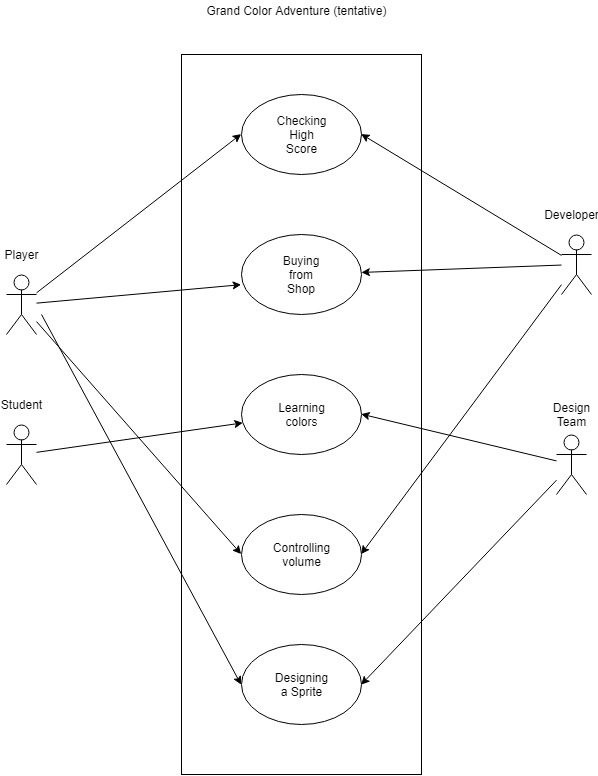
5. Sales subtract from total coins player has

6. Avatar added to their gaming experience

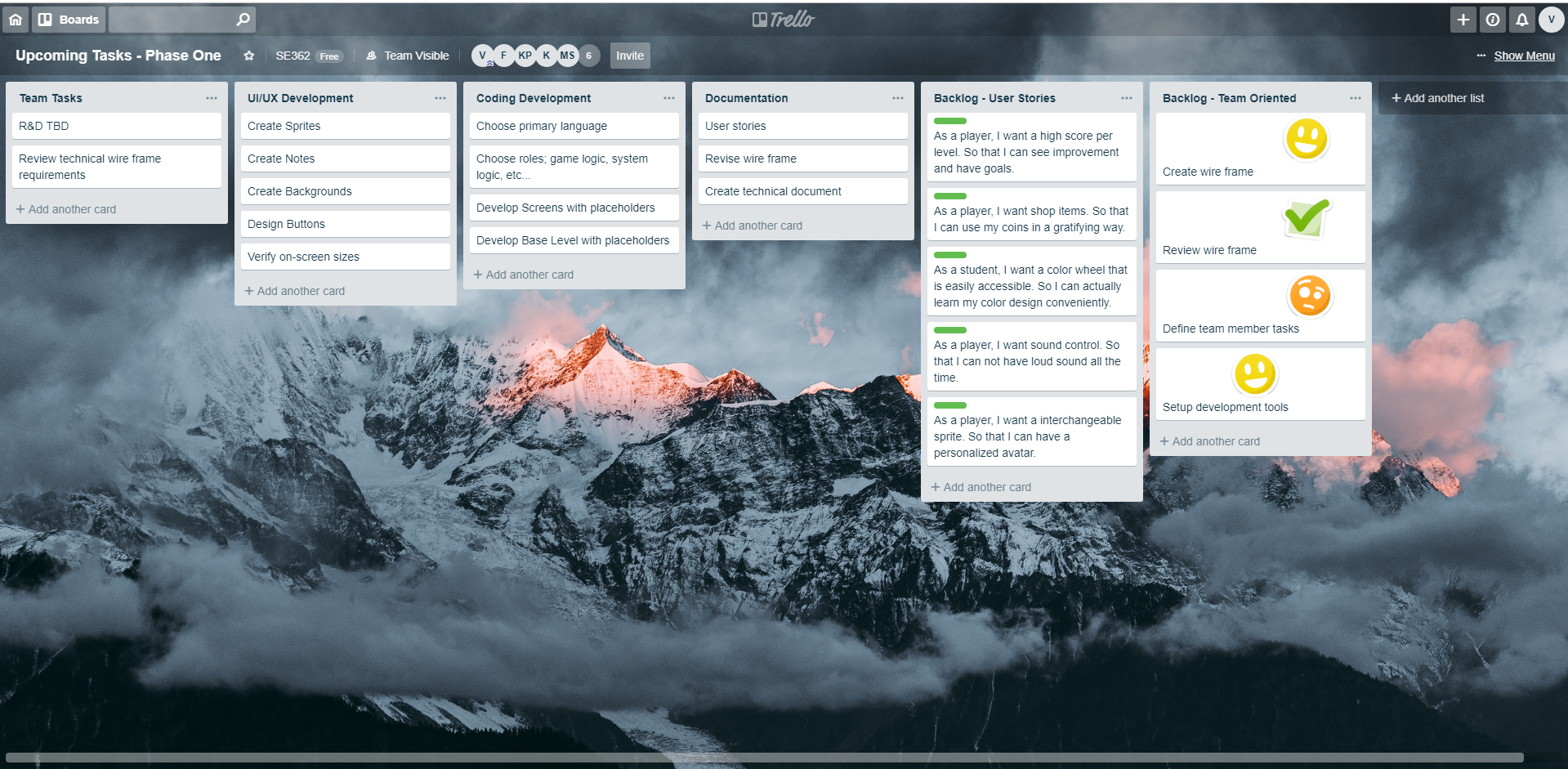
Extensions:

None

**Use-Case Diagram**



**Sprint Backlog**



**Pre-Game Planning**

For the first sprint, the team decided to focus primarily on designing and understanding the foundation of the game. This meant those with roles based on design tasks had to come together to explore what the game would be comprised of, while those with roles based on development had to think of how to implement the game’s composition. This led the design team to develop a wireframe of the game, a mockup of the game’s prototype. It also led the developers to familiarize themselves with a collaborative tool of choice, the Unity engine.

During this process, the team as a whole would come together to explore ideas to ensure a product that took in the input of all members. The team attempted to emulate the process of design thinking, the structured exploration of design goals using the skill sets of varied members. By incorporating each member’s strengths, the team managed to develop the game visually while also planning for the logical implementation of those designs. In class activities such as Planning Poker were utilized during this process as well.

**First Iteration Staging**

For the first sprint, the staging process heavily depended on understanding what elements were necessary to the game. This meant exploring a centralized theme to base our product around, which eventually became color design theory. This transitioned into how we wanted to develop the UI of our product to incorporate the theme while also providing a positive user experience. This resulted in a blueprint for individual screens, guidelines for future visual development, and an overall understanding of the flow of the game.

Following this, the team worked together to create an understanding of the logical implementations of the designs. This meant the formation of the logic behind the game itself and the system interacting with the user. The team put together a basis as to how to play the game, how the game would reward the user, and how those rewards could be used to further incentivize gameplay. Overall, a concept of the project’s structure was created to enable concrete results in the future.

**Development Process**

The team decided that the there would be two main focuses for the development of the game. These focuses were split as creative design and logical design. The creative design team would be in charge of the user interface, user experience, and overall creative highlights of the product. The logical design team would be in charge of ensuring functionality and feasibility of the product.

The team agreed to use the Unity engine as a framework to help in the game’s development. With this in mind, the creative design team explored our options in regards to visuals. The Unity engine is able to utilize both bitmap and vector images. The creative design team decided to use vector images for visual assets as they are easily scalable and provide greater flexibility. A majority of the vector design work will be done with Adobe Illustrator as it is used in professional environments and made available to students by the campus.

The logical design team faced the issue of what programming language to use alongside the Unity engine. The two prominent choices were either C or Python. Due to the experience of some members, it was agreed that Python would be the language of choice. This means the project will be developed with the Python language working with the Unity engine.

**User Manual**

|  |  |
| --- | --- |
|  | **Start Screen**   * Title: identifies game; graphic only * Start: take user to level selection screen * Store: take user to store screen * Settings: take user to settings screen * Sprite: random owned character chosen for showcasing * Note: random owned note chosen for showcasing   Users will be taken to the Start Screen upon launching the game. From here they may navigate to all indicated screens by selecting their corresponding buttons. |

|  |  |
| --- | --- |
|  | **Level Selector Screen**   * Sprite: user chosen character chosen for showcasing * Arrows: changes current level; event to update screen * Level Display: number indicates current level; stars indicate previous performance * Description: text explaining goal of the current level; explains color concept of the level * Home: take user to the home screen * Color Wheel: visually display the colors associated with the level goal * Coins: indicate how many coins the user has   Users will be able to choose the level they wish to play on this screen. The arrows will allow the users to navigate between levels. The home button can be selected to go to the Start Screen and the color wheel can be selected to bring a larger color wheel to view. By selecting the level button in the center, the user will be taken to the game stage. |

|  |  |
| --- | --- |
|  | **Stage Screen**   * Sprite: user chosen character * Notes: object to be caught by user; color of note will affect point increase * Timer: display how many seconds left for the stage * Goal: text displaying the color goal of the stage * Pause: pause the system; trigger the pause screen to appear * Bonus Icon: user chosen character; when pressed it will activate the character’s bonus * Color Wheel: visually display the colors associated with the level goal * Streak: indicates how many catches have been made without error   The user will play the game on a Stage Screen. Individual stages will be customized to unique goals, but overall gameplay will remain the same. The user will be rewarded for catching a correct note in any of the three possible positions. They may choose to use the sprite bonus icon at the bottom left to use a sprite bonus. They may select the pause button at the top right to open the Pause Screen. |

|  |  |
| --- | --- |
|  | **Pause Screen**   * Score: display the current score * Stars: display the current stars earned * Resume: remove the pause screen; game will resume * Settings: brings user to the Settings Screen * Home: return to the Home Screen   The user may view their current performance at this screen. They may also navigate back to the game stage, to the settings menu, or return to the Start Screen from this menu. |

|  |  |
| --- | --- |
|  | **Stage End Screen**   * Sprite: user chosen character * Score: amount of points earned this playthrough * Stars: amount of stars earned this playthrough * Resume: take user to the Level Selector Screen; automatically increment the current level * Store: take user to the Store Screen * Home: take user to the Home Screen   The user may view their performance on the completed stage on this screen. They may also navigate to the Level Selector Screen, the Store Screen, or the Start Screen. |

|  |  |
| --- | --- |
|  | **Store Screen**   * Sprite: user chosen store object * Name: name of current store object * Coins: cost of current store object * Sprite\_Col: change category to sprites * Note\_Col: change category to notes * Backdrops\_Col: change category to backdrops * Arrows: navigate through current store objects of current category * Home: take user to the Home Screen * Purchase: allow user to purchase the current store object * Coins: show current count of coins owned by user   The user may purchase any store objects. The column on the right will be used to navigate between categories: sprites, notes, and backdrops. The row above the bottom UI will be used to navigate between objects of the current category. The bottom UI will allow the user to return to the Start Screen, complete a purchase of the current store object, or view their current coin count. |

**References**