# Project Proposal Team 19

**Team Members**

- Dawson Rooney
- Wesley Sportsman
- Christopher Smith
- Trey Werr
- Jacob McNamee

**Project Name:** Palette

## Project Synopsis:

Palette is a 2D platformer where the player uses different colors of paint with unique properties to complete puzzles.

## Project Description:

The core concept behind this project is a platforming game where the player has some control over the platforms. Using this concept as a base, we have decided to develop a puzzle game where the player can use different types of paints with different properties to interact with the environment. All great puzzle games have their own unique twist that makes the player think outside of the box. We wanted to do just that, but with a 2D platformer with paint. Our goal with Palette is to have the player use five different colors of paint that have their own attributes. These paints give us a plethora of puzzle combinations, allowing us to create a variety of maps while also giving the player a novel experience that never drags on.

The end product of this project will be a complete and functional game. The game will have multiple “worlds”, or sets of levels, each of which will focus on one of the paint colors or on a combination of colors. The player will only be able to use the paints specified by a given world.

# Project Milestones:

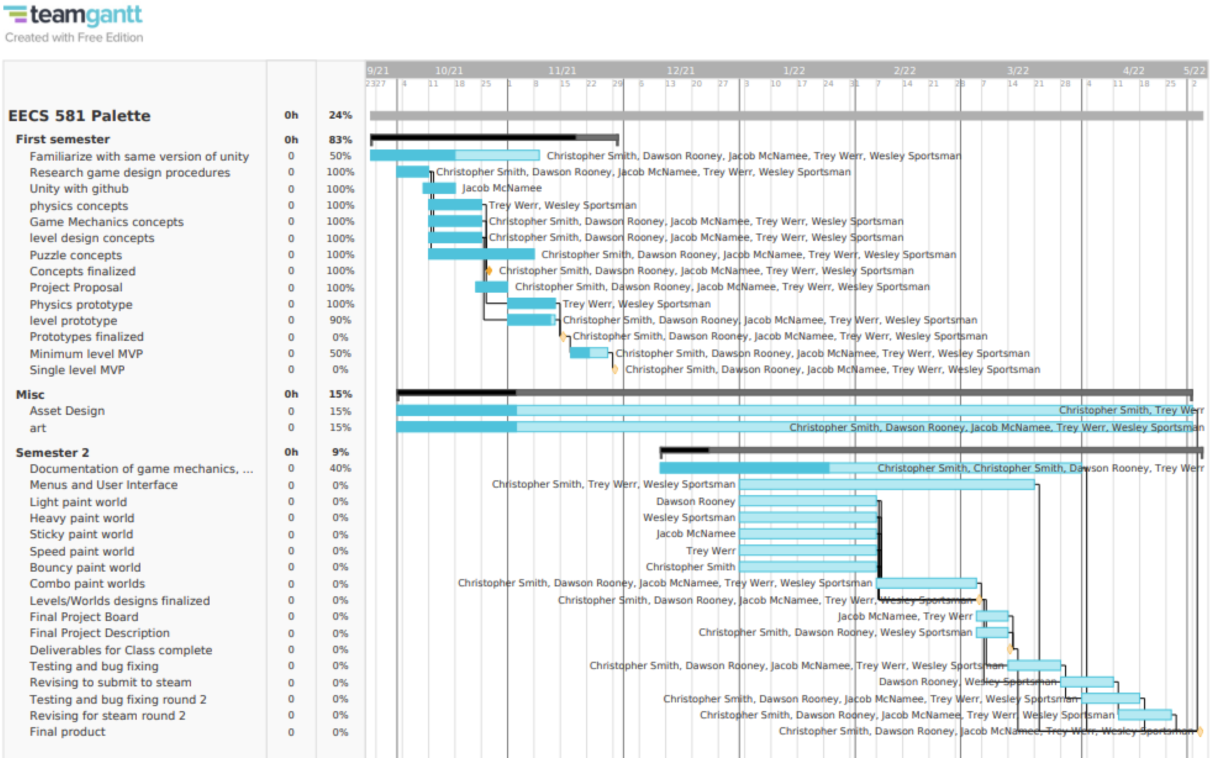
**First Semester:**

Game mechanics/physics/overall design concepts finalized: 10/29/21
Prototypes of above concepts: 11/19/21
Single level MVP: 11/26/21

**Second semester:**

Each paint implemented: 2/04/22
Game ready for testing: 3/04/22
Documentation completed: 3/11/22
Final project deliverables completed: 3/14/22
Game available on Steam: 5/3/22

## Gantt:



| | | | |
|---|---|---|---|
| **EECS 581 Palette** | **0h** | **24%** | |
| **First semester** | **0h** | **83%** | |
| Familiarize with same version of unity | 0 | 50% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Research game design procedures | 0 | 100% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Unity with github | 0 | 100% | Jacob McNamee |
| physics concepts | 0 | 100% | Trey Werr, Wesley Sportsman |
| Game Mechanics concepts | 0 | 100% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| level design concepts | 0 | 100% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Puzzle concepts | 0 | 100% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Concepts finalized | 0 | 100% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Project Proposal | 0 | 100% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Physics prototype | 0 | 100% | Trey Werr, Wesley Sportsman |
| level prototype | 0 | 90% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Prototypes finalized | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Minimum level MVP | 0 | 50% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Single level MVP | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| **Misc** | **0h** | **15%** | |
| Asset Design | 0 | 15% | Christopher Smith, Trey Werr |
| art | 0 | 15% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| **Semester 2** | **0h** | **9%** | |
| Documentation of game mechanics, ... | 0 | 40% | Christopher Smith, Christopher Smith, Dawson Rooney, Trey Werr |
| Menus and User Interface | 0 | 0% | Christopher Smith, Trey Werr, Wesley Sportsman |
| Light paint world | 0 | 0% | Dawson Rooney |
| Heavy paint world | 0 | 0% | Wesley Sportsman |
| Sticky paint world | 0 | 0% | Jacob McNamee |
| Speed paint world | 0 | 0% | Trey Werr |
| Bouncy paint world | 0 | 0% | Christopher Smith |
| Combo paint worlds | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Levels/Worlds designs finalized | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Final Project Board | 0 | 0% | Jacob McNamee, Trey Werr |
| Final Project Description | 0 | 0% | Christopher Smith, Dawson Rooney, Wesley Sportsman |
| Deliverables for Class complete | 0 | 0% | |
| Testing and bug fixing | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Revising to submit to steam | 0 | 0% | Dawson Rooney, Wesley Sportsman |
| Testing and bug fixing round 2 | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Revising for steam round 2 | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |
| Final product | 0 | 0% | Christopher Smith, Dawson Rooney, Jacob McNamee, Trey Werr, Wesley Sportsman |

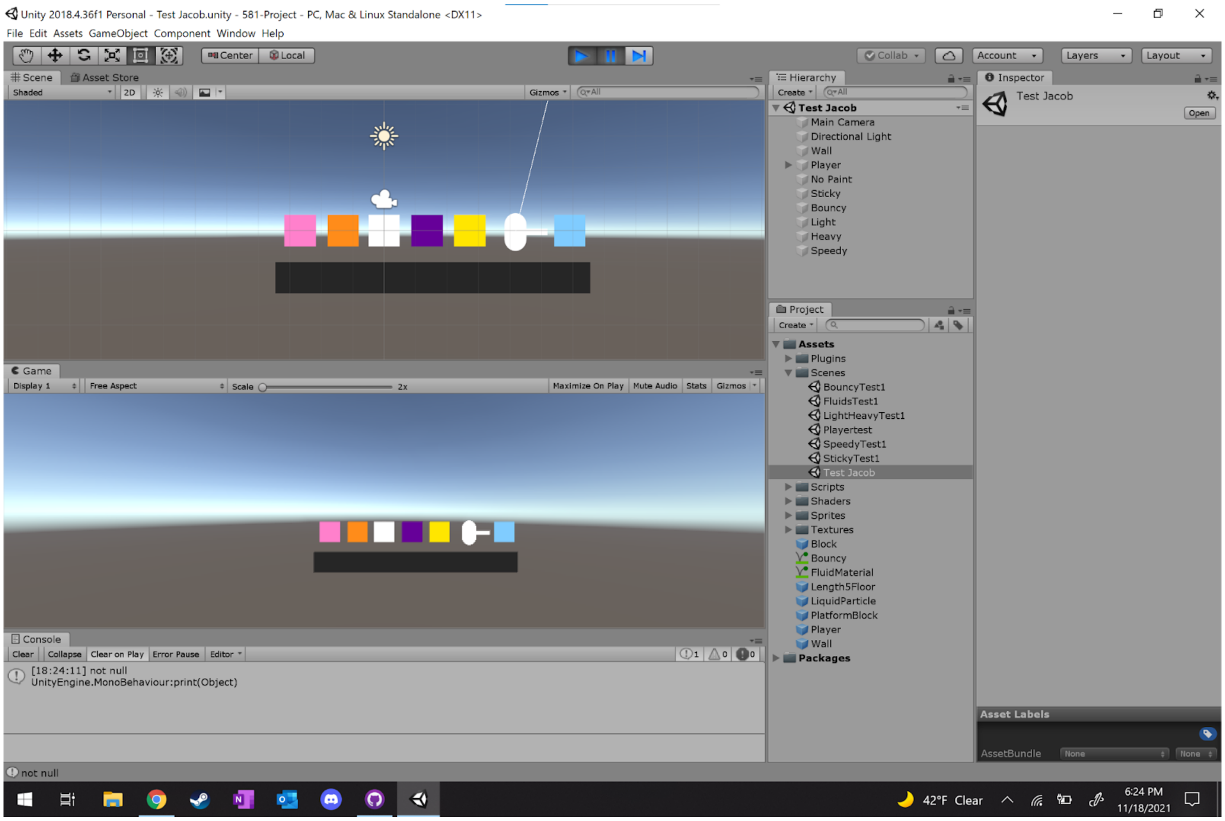
# Project Budget:

$**0** - only requires Unity which is free
$**0** - free online music samples and self-made music
$**100** - free online assets - or collaboration with art department
$**100** - publishing final product on Steam (game marketplace)

# Final Project Design:

## How the Software Works:

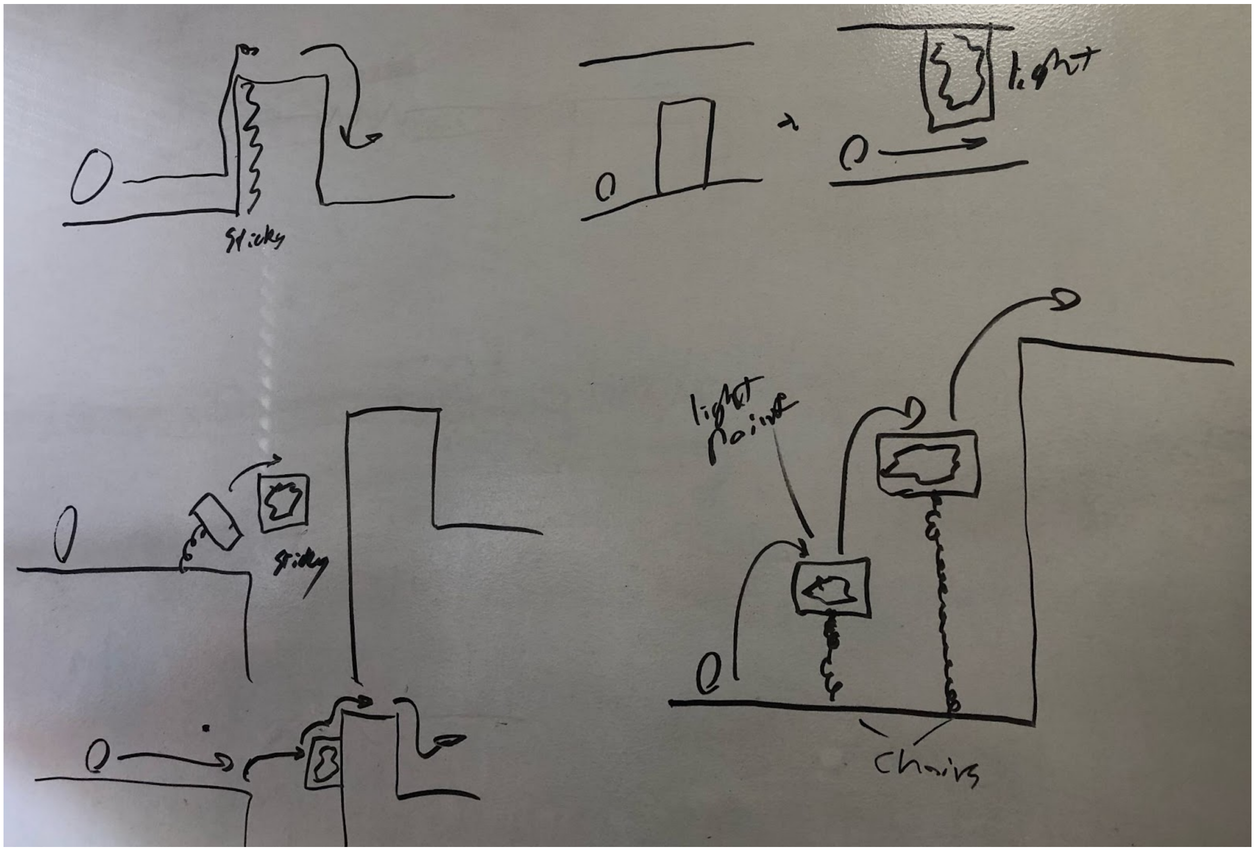
We are developing our game for Windows OS using the Unity game engine.

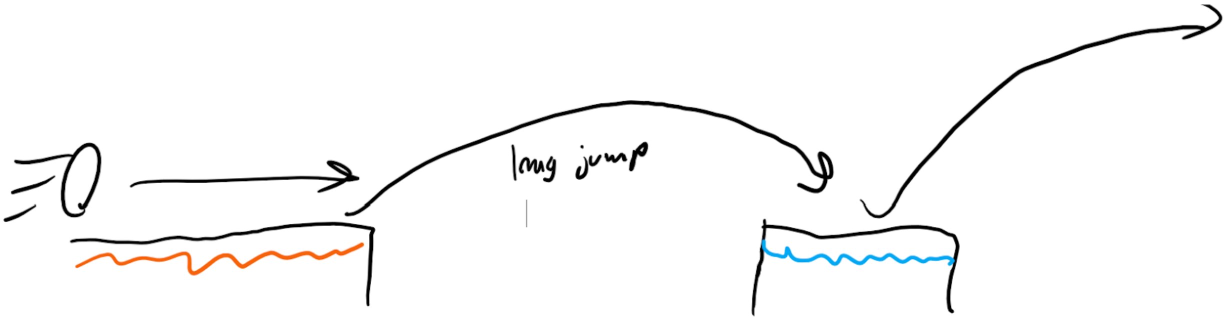


```
[18:24:11] not null
UnityEngine.MonoBehaviour:print(Object)
```

```
not null
```

The game is a puzzle-platformer whose main feature is the ability to use multiple colors of paint to affect the world. We have planned for there to be five paint colors. Green paint is sticky, and makes objects cling to surfaces they collide with. When the player collides with a sticky surface, they will stick to it as well. The player will be able to move along sticky surfaces, effectively climbing along things which are painted green. Using the jump button, the player can jump off of the sticky surface. Blue paint is bouncy and can be used to jump higher or bounce objects. When the player presses the jump button while interacting with bouncy paint, they will be able to jump even higher. Orange paint makes the player and objects speed up and can be used to make interesting platforming challenges. More specifically, orange paint will cause things moving on the painted surface to speed up, applying a force on the colliding object parallel to the painted surface. For physics objects, this means that an object that is already moving will start accelerating when on an orange surface. If the object itself is painted orange, it will gain speed whenever it is in contact with another surface. The painted object can even become a hazard if it starts moving fast enough to hurt the player. In the case of the player, they will only receive the speed boost when they are pressing a movement key, otherwise they will quickly come to a halt due to friction. Yellow paint makes objects lighter so that they float into the air. This is accomplished by applying an upward force on the object. Purple paint makes objects heavier by increasing their mass in the physics engine. Below is an image of some level concepts involving sticky and light paint.





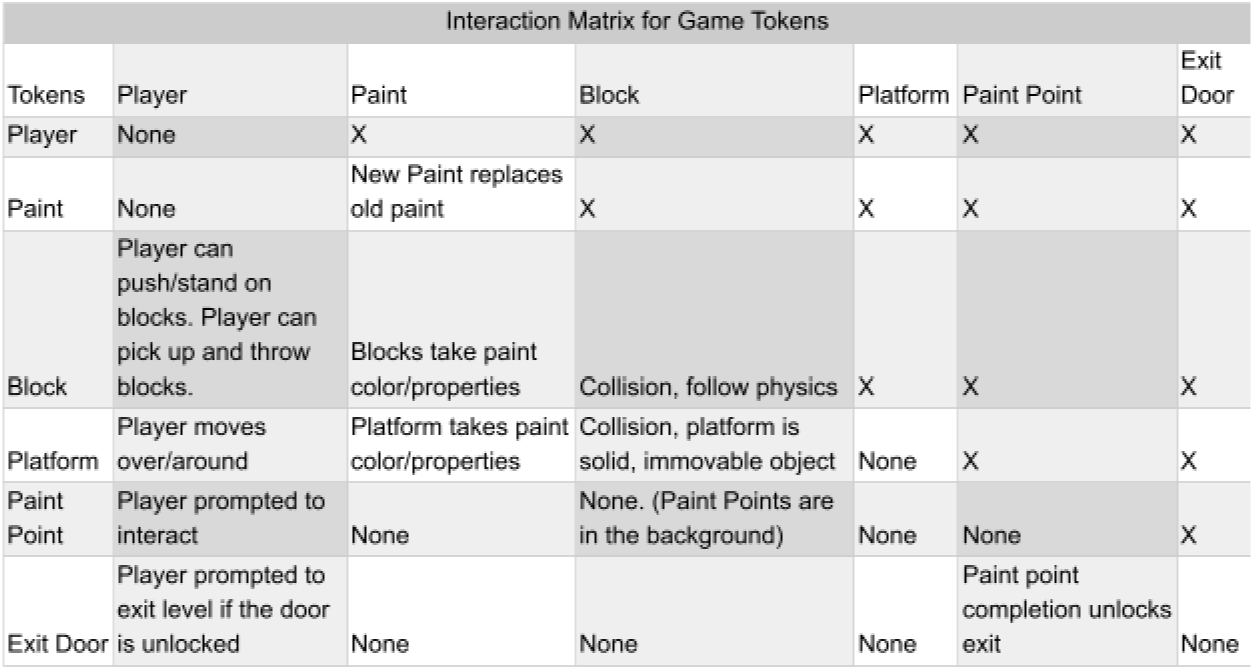
This picture details a use case for speed paint (orange, left) that is used to allow the player to jump a long distance onto a bouncy surface (blue, right) as means of completing a puzzle.

The player starts a level with a limited amount of paint. By constraining the amount of paint they player has we force them to be strategic in the use of their paint.

They cannot simply paint all of the surfaces (the brute force method), they must think carefully and try multiple things to solve the puzzles. There will be two ways for the player to refill their paint supply. One is a paint pool, which is a vat of a single color of paint. We will also use paint pools to show off our fluid physics. The other paint refill option is a single-use paint pickup which will replenish a finite amount of paint.

The player will have a particular set of paint colors available to them on any given level. Using these paints, they must figure out how to progress through the level and get to a door which lets them leave the level and advance to the next one. Levels will include pipe-like objects called “paint points”; these will protrude from the background of the level and hook up to a network of pipes. The player uses paint points to inject paint into the background, returning color to their world. The player must get to these paint points and recolor the world before they can exit the level.

The table below shows how various components will interact with each other in the game:



Interaction Matrix for Game Tokens

| Tokens | Player | Paint | Block | Platform | Paint Point | Exit Door |
|---|---|---|---|---|---|---|
| Player | None | X | X | X | X | X |
| Paint | None | New Paint replaces old paint | X | X | X | X |
| Block | Player can push/stand on blocks. Player can pick up and throw blocks. | Blocks take paint color/properties | Collision, follow physics | X | X | X |
| Platform | Player moves over/around | Platform takes paint color/properties | Collision, platform is solid, immovable object | None | X | X |
| Paint Point | Player prompted to interact | None | None. (Paint Points are in the background) | None | None | X |
| Exit Door | Player prompted to exit level if the door is unlocked | None | None | None | Paint point completion unlocks exit | None |

This chart denotes interactions that each in-game entity has with other in-game entities. The chart is symmetric so an “X” represents an equivalent cell across the diagonal.

There are multiple fail conditions (ways to lose) in our game. A player could die from a map hazard, such as a pitfall or a laser. Moving platforms/walls could push a player into a hazard, or even crush them. If an object strikes the player with enough force, such as from the acceleration of orange speed paint, the player will die. If the player gets stuck somewhere or runs out of paint and cannot proceed, they must restart the level.

The main mechanic of the game is painting things. The way we are going to implement this is different depending on what is being painted. Currently we have two broad categories of game objects; blocks and platforms. Blocks are the physics objects and will move dynamically in the level; additionally, the player can pick up and move blocks if they are small enough and light enough. When a block is painted, the entire object will take on that color. Platforms are static objects and cannot be moved by the player. Platforms will be segmented on the back-end of the program so that they can be painted in small chunks rather than as a whole. This will be designed to mimic how real paint functions.

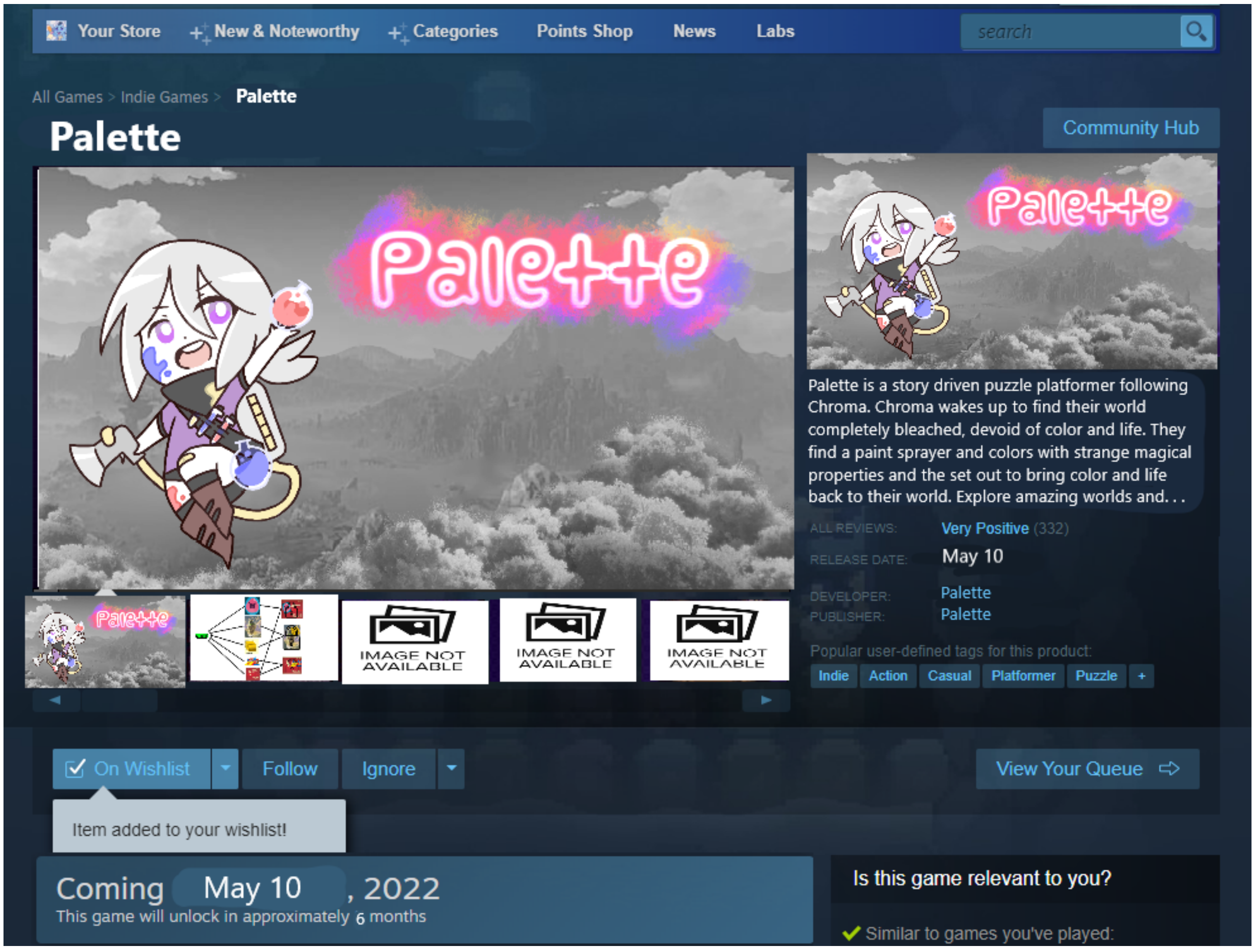
## Design Constraints:

Our design constraints are heavily based on how our paints function and how the world interacts with them. This means that our levels have to be based around our paints. This limits our puzzle ideas for individual paints, but allows us to create more complicated puzzles when combining multiple paints. Multiple Paint worlds are a stretch goal as the design process moves forward, we intend to work diligently to get to the point we can finish them. Whenever a paint interacts with a block, it covers the entire block and now has the effect of the paint being put on it. This means there can only ever be one paint color on a single block, meaning that multiple blocks may need to be used in order to complete specific tasks (ex: a block needs to stick against a wall for a player to jump on, but another block is needed to bounce the character up the wall), or that the puzzle itself may not be feasible in our system.

Another design constraint is player interaction with the world. The levels are based around the limitations of the player, such as their movement speed, jump height, and size. Since players can die by hazards, this creates a balance between level complexity and the amount of hazards and fail conditions. We want the player to think outside the box while also not feeling as if they can't complete a level.

Our technical design constraints are largely determined by our choice of game engine. We have chosen to build our game in Unity. This has many advantages, since Unity has built in libraries for physics, audio, level creation and visuals. Unity uses scripts to make game objects function, and the scripting language Unity uses is C#. We are also constraining our project to be specifically for Windows OS, as that is the operating system that all of the team members use and is the most widespread amongst games on Steam.

Our business constraints are fairly simple for this project. The due dates for our project are known, our team composition is set for the duration of the project, and we have an easy budget. We will place the game on Steam once we are done, so we must meet Steam's requirements.



Palette

Palette is a story driven puzzle platformer following Chroma. Chroma wakes up to find their world completely bleached, devoid of color and life. They find a paint sprayer and colors with strange magical properties and the set out to bring color and life back to their world. Explore amazing worlds and. . .

| | |
|---|---|
| ALL REVIEWS: | Very Positive (332) |
| RELEASE DATE: | May 10 |
| DEVELOPER: | Palette |
| PUBLISHER: | Palette |

Popular user-defined tags for this product:
Indie Action Casual Platformer Puzzle +

Item added to your wishlist!

Coming May 10, 2022
This game will unlock in approximately 6 months

Is this game relevant to you?

✓ Similar to games you've played:

# Ethical Issues:

As a non-violent video game, there are very few ethical issues with the production and release of Palette. The most relevant sections in the ACM code of ethics are 1.4 (Be fair and take action not to discriminate) and 1.5 (Respect the work required to produce new ideas, inventions, creative works, and computing artifacts).

The ACM code of ethics section 1.4 states to take action not to discriminate. Our game is marketed towards people of every demographic, so we need to be mindful of the hardships people might have when playing. The largest difficulty is that color plays a very large part of our game, and this could be exclusionary to people with color blindness. One possible solution for this is to include specific geometric patterns within the paint, so that the different colors can be differentiated through an alternate means. Another difficulty stems from the unique clues that games, and especially puzzle games, give their players. These clues are often subtle, and people who have not played many video games might not pick up on these hints. We should both include these hints, and make them obvious enough that new players can see and understand them.

ACM code of ethics section 1.5 requires us to respect the work required to produce new ideas and creative works. Our game will contain a lot of content, primarily art, music, and sound, that will have been created by people not within our group. These works should be attributed to their proper creators and these creators should be credited within our game. Additionally, we must abide by the rules these creators place upon the use of their work, including following any applicable copyright. As the original creators, their time and effort should be respected, so we should respect them to the best of our abilities.

## Intellectual Property Issues:

For the beginning stages of our project we will have few intellectual property issues as most things we work with will be part of the default Unity engine or products of our own code. Issues may arise if we use code snippets from other projects. One way to ensure this doesn't become an intellectual property issue is to only look at open source projects for inspiration.

Later in the project we may have intellectual property issues when it comes to the visual and audio design of our project. We can either make assets ourselves or find a free repository of assets to fill out the visuals of our game. If these are insufficient we can use our budget to acquire licenses to use other art, or we can collaborate with the KU art department to have custom-made assets. These latter solutions require more care to ensure that no intellectual property rights are infringed, but they also may result in a more aesthetically pleasing game so the hassle could be well worth the time and money required.

## Change Log:

- Mark Combo paints as stretch goal
- Music no longer is a collaboration with the Music Department, now self-made or online assets.
- Updated images to better fit requirements
- Project Budget updated to allow for some paid art assets if we need them.