Collision Star

"Collision Star", a 2D multiplayer game that fuses strategy and adventure. As a party game, 2 players can choose to cooperate or compete at any time.

The design concept of my game project is inspired by the integration of customizable scene and physical elements. Players can continuously modify the game scenes, add objects, and bring changes to the initial environments. Additionally, during the process of setting up the scenes, players can engage in strategic gameplay, enhancing the overall amusement. Furthermore, the physical elements, enable players to navigate through the scenes by bouncing and colliding, collecting points, and ultimately achieving victory.

Players navigate to victory by strategically placing objects, overcoming obstacles, and accumulating points. The game introduces a reverse drag-and-drop mechanism, enabling players to control the move of their objects. During the process of micro, both player can decide to start a move at any time. If one player move at the cost of 5 points, the other player also move after that. The game incorporates elements such as color-changing objects, points-awarding stars, and special features, adding a layer of strategic complexity and enjoyment.

Links:Game：https://github.com/xxsKyrreZLX/CollisionStar

Video：https://youtu.be/YJECfxrV36Y

Design Process

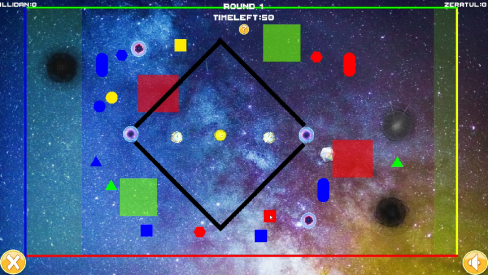
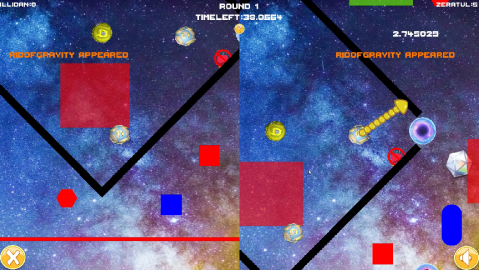
1.Drag and Drop Functionality

Players move their objects by reverse drag-and-drops. By capturing the positions of mouse clicks on objects and their release points, the game assigns movement speeds to these objects. It then converts these speeds into corresponding positions on the user interface (UI), simultaneously reflecting the directional changes through arrows on the UI.

2.Creating Multiplayer Scenes

Designing Different Stages . Each round consists of four stages. In Stage 1, Player 1 selects and places objects, followed by Player 2 in Stage 2. Stage 3 involves continuous gameplay and manipulation by both players. When the round's time is up, it progresses to Stage 4. Either one player's sphere is consumed by a black hole or one player successfully acquires points from the golden sphere will also progresses to Stage 4. Stage 4 displays the scores attained by both players, following which the cycle begins again with Stage 1, but this time, Player 2 selects objects and places them before Player 1.

During Stages 1, 2, and 4, a fixed camera captures the overall scene, while in Stage 3, two separate cameras track the movements of each player's objects, displaying their respective actions.

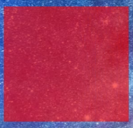
 

3.Designing Map Elements

The map incorporates elements that interact with players or objects within the game.

 Black holes attract and destroy objects, while white holes repel objects.

 Portals will transport objects to the exit.

 The red area is the deceleration zone, while the green one will accelerate.

 The light area will push the object in a certain direction.  
4.Adding Various Scoring Objects

Scoring objects are crucial for achieving victory and require careful numerical and mechanistic design. The collectible objects in the game consist of blocks, scoring balls, and stars. All scoring objects change ownership upon collision with a player's object. If destroyed during the ownership period, the player who last made contact with the object before its destruction earns the points.

1. Blocks

Blocks have a base value of 50 points, and the method to destroy them involves making them collide with the same-colored boundaries of the scene. There are two types of blocks: one type changes color upon each collision with a player's object, while the other type switches colors automatically every few seconds.

1. Scoring Balls

The scoring ball has a base value of 500 points and is unique. When it gets sucked into the black hole, it grants a substantial amount of points and ends the current round.



1. Stars

Stars refresh periodically and have a base score of 15. Players can earn points by colliding with them.

5.Adding Randomly Generated Skills

At regular intervals, the game generates random power-ups at various locations. Players can gain the effects of these skills by colliding with them.

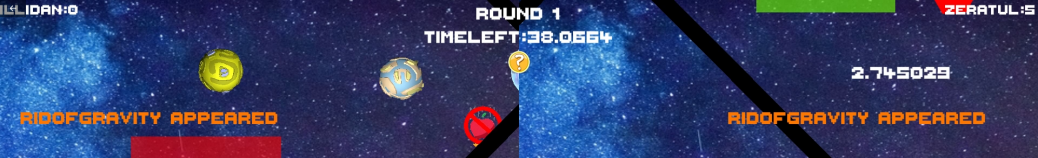
Mass increase Not affected by gravity Launch missile to attack

1699588820445 1699588771191

More points in the coming period Earn points directly  
Improving the Experience:

(1)More Comprehensive UI

After inviting classmates and friends to play, they provided feedback that the game rules weren't easily understandable, leaving them unsure of what was happening and how to play. Therefore, I continuously worked on enhancing the UI experience.



(2)Introducing More Intuitive Scoring Methods

Based on players’ feedback, it was apparent that the scoring system lacked clarity and variety. They found it hard to gain score and they were not informed they achieved something. To address this, I introduced randomly spawning stars that grant relatively fewer points so that they won’t have great impact on the game's outcome, thereby offering players a sense of satisfaction and progressing the gameplay. Additionally, I implemented visual effects, such as trailing effects behind the player's ball during high-speed movement, and particle effects upon the destruction of various objects, all of which significantly enhanced the overall game experience. Furthermore, for color-changing blocks, the destruction effects now smoothly transition based on their respective colors, eliminating any jarring visual experiences.

(3)Abandoning Unreasonable Demands

Throughout the development process, I abandoned and modified several initial plans. For instance, one plan involved the player's trail having the ability to destroy objects. Initially, the idea was to encourage players to attempt high-speed movements by allowing the trail to destroy objects and earn points. However, this idea was eventually discarded. Firstly, this mechanism would encourage players to continuously charge their power bars, potentially leading to brute-forcing levels, which could detract from the thoughtful gameplay experience. Secondly, the trail effect only appears when the object reaches a certain speed and continues for a specific duration, which often results in the player's camera moving away from the trail, leading to confusion regarding how points were earned, thus creating a chaotic gameplay environment. As a result, despite my initial excitement about the idea, the trail effect now serves purely as a visual element, contributing to a more reasonable and coherent gameplay experience.

Reflections:

(1)Camera Switching

During the development process, I encountered unexpected challenges, such as displaying arrows on each player's object during multiplayer object manipulation. To resolve this, I extensively studied multiple canvases and cameras and associated them with the objects and arrows to ensure accurate arrow display. After numerous adjustments, I successfully enabled seamless camera switching during gameplay. Specifically, the global camera is used for selecting and placing obstacles, while two separate cameras follow each player's actions, displaying on the left and right halves of the screen, each with distinct, non-interfering backgrounds. By setting rendering relationships and assigning rendering layers to the objects, each camera is equipped with its own background. This solution significantly enhances the game's playability and competitiveness by providing clear instructions during multiplayer battles.

(2)Generating objects

To streamline the gameplay experience, I directly instantiated prefabs for the corresponding objects when players placed them in the scene. Additionally, for creating skills, I realized the similarity in triggering mechanisms among various skills. As a result, I consolidated the handling methods for all power-ups within the game manager for the skills to access. In the skill generator, I set index to randomly generated skills, enabling them to invoke functions and set images. These efforts reflect my attempt to design my code in line with the ECS framework, minimizing script-related resource consumption and reducing confusion in game logic handling, such as the Update method, thereby enhancing code clarity and readability.



1. Player feedback is extremely important.

I invited some players and other developers to try my game and their feedback provided me with a lot of valuable insights for improvement. One player felt that the UI prompts in the game were not clear enough, and he wasn't sure when certain actions could be taken or what was happening. Another developer suggested that I find ways to introduce more scoring methods and use special effects to make players feel it COOL. These ideas gave me excellent directions for improvement. I believe that communicating with other developers and having players try out the game is crucial in game development and I’m glad to bravely ask them to have a try.

（4）Project Expectation

The game was expected to support 2-4 player online gaming without the need for split-screen gameplay, assuming each player with an independent screen on a single machine. I can remove the pausing with an input to manipulate objects, granting players the freedom to continuously operate objects. With the expanded operational space for players, the game could incorporate more complex and challenging interactive elements, making the gameplay even more exhilarating. However, due to my limited experience in online development and the independent nature of the game's production, testing and creating an online version proved challenging, leading to the decision to create a local split-screen two-player game. If given the opportunity in the future, I hope to implement an online version of the game.