Rainbow Blight

A Twin-Stick Shooter Game

Project by: Chris Hume, Chris Dillman, & Ryan Smalley

Description: Rainbow Blight is a twin stick shooter game in Unity, with C# scripts, and used GitHub to collaborate. A twin stick shooter is a shooting game, where movement and aiming are controlled independently. For our project we chose a top-down perspective (directly above the game space). It is survival based, so you score is based on how long you live, and how many rainbow circles you took with you before dying. The enemies (rainbow circles) spawn out blue boxes. To help the player survive there are blue speed boost circles and walls to block the enemies (they block you as well). Once defeated you will be give you final score and the option to replay or quit.

Controls: As they are required to use the program, here are the controls for the game:

W-Move Forward Left Click-Shoot/Buttons  
A-Turn Left Mouse: Aim  
D-Turn Right R- Restart the game  
S-Move Backward

New Concepts: Using GitHub with Unity was new to all of us. While there was a script online to make them play nice with each other, it didn’t work. We tinkered with what GitHub would ignore, until we found a configuration that allowed us to share our progress freely. We all have never worked on a game like this. There were some tutorials on Unity’s site for similar projects that help us get a basic understanding.

Division of Labor: Every week we would meet and discuss our progress. During each meeting we would discuss what needed to be done next and assign tasks appropriately. Below is a brief breakdown of the work. While this doesn’t cover everything, it gives the basic idea.

Brainstorming: Everyone  
Going over Unity/GitHub Basics: Everyone  
Movement: Chris Hume and Chris Dillman  
Shooting: Ryan Smalley  
Level Design and Spawning: Chris Hume  
Post-Game Menu: Ryan Smalley  
Collision: Chris Dillman  
Enemy AI: Chris Dillman  
Merging GitHub: Chris Hume  
Finishing Touches: Everyone  
This Report: Ryan Smalley

Issues and Workarounds: The shooting and movement separation went through several revisions before completion. This was problematic because, both were trying to use the rotation of the character, and collisions were altering rotation. On our 3rd version of the shooting and moving system we got them independent of each other and fixed the collision rotation quirk. Also as stated previously GitHub and Unity weren’t playing nice together, but enough tinkering in team meetings and we figured it out. Unity also would in the beginning not show the correct scene, and would show previous ones, despite them being deleted. This problem was likely caused by the GitHub and Unity issue, or it may have been the files were being opened. We stopped using Unity’s Open command, and instead just opened it from the scene file directly.