Asteroid Dodge

Melvin Bosnjak, Ryan Dougherty, Samuel Bridge, Zane Yarbrough

Final Project Report

Abstract

Our project is a web based game called Asteroid Dodge. In this game, the user controls a ship by using the left and right arrow keys and avoiding the oncoming asteroids. The goal of the game is to get as far as you can without crashing into an asteroid and collecting the crystals that spawn as you go. With these crystals, users can purchase new ship skins and galaxies to explore. Users can also go to the leaderboard page to view the current top highscores.

Introduction

Our project is a fully functional asteroid dodging video game developed in Javascript and running on ASP.Net. The game features a small ship that the user can control by pressing the left and right arrow keys. The goal of the game is to dodge incoming asteroids and stay alive for the longest possible time to increase their score. Occasionally, little blue space-crystals will appear from the top and sides of the screen. The user can collect the crystals and use them as currency to buy items in our in-site store.

Our game's store features 6 ship skins and 4 background skins that the user can purchase. Each skin is worth progressively more crystals, to encourage our users to keep coming back and playing more. Our store also allows you to select your current skin by simply clicking on a skin of your choice. When you navigate back to the game, you will see your newly selected skin is saved and displayed.

Lastly, our game has a leaderboard page that displays all saved user scores in decrementing order. It displays in multiple pages and allows you to search by username. All these features come together to create a fun well-built game.

Feature Table

Feature Name	Scope	Primary Programmer	Time Spent	File/Function	LoC

Game JavaScript, and game page	other	Melvin	20 hrs	Home/Index.cshtml, wwwroot/js/game.js	500
Leaderboard View	UI	Melvin	2 hrs	Leaderboard/index.cshtml	30
Store View	UI	Melvin, Ryan, Sam, Zane	10 hrs	Store/index.cshtml	70
AJAX calls to update score, crystals, and get current ship	other	Melvin, Sam	2 hrs	wwwroot/js/game.js	30
AJAX calls to update/get background	other	Zane	1 hour	wwwroot/js/store.js	50
Store Controller	Backend	Ryan, Zane, Sam	6 Hrs	/Controllers/StoreController.cs,	260
User Framework	other	Zane	4 Hrs	Areas/Identity/...	~
Leaderboard Controller	Backend	Ryan	1 Hr	/Controllers/LeaderboardController.cs	72
Home Controller	Backend	Ryan	2 Hrs	/Controllers/HomeController.cs	95
Database Model	Backend/ Model	Ryan, Zane	4 Hrs	Models/*	~180
AWS	Deploy	Ryan, Zane	2 hours	Proj file	null

Individual Contribution

Team Member	Time Spent on Proj	Lines of Code Committed
Melvin	30 hours	~700
Ryan	18 hours	~600
Sam	12 hours	~300
Zane	16 Hours	~400

TODO: everyone add their individual contributions

Summary

Overall, we were all very happy with how the project came out. We all exceeded our expectations. We worked well as a team, we believe our team performance deserves a Superior level. Our weekly meetings and continuous slack communication worked well to keep up with the weekly sprints. We split up the project evenly between each of us and when one member was struggling on their assignment, others were able to pitch in and help.

Some obstacles that we had to overcome were merge conflicts, AWS services, and game code. The merge conflict obstacle comes with any group software project. It just took some time and coordination to fix. For AWS services we originally put one person in charge of it but after it gave us some struggles we put more than one team member on it to help overcome it. The game code was a lot more work than anticipated, not so much in the game logic but in the drawing of the sprites and making everything look smooth.

We all have a lot of pride in this project and will be definitely be showing it off to friends and family in the future. It feels like a real game and the leaderboard and store features cement it as a web-game rather than just a project.