SpaceX Shooter

CSCI 2941 Game Design & Dev 1
Christopher Ngo, Alexander Verkest, & Drake Young

Contents:

Project Description

Inspired by the recent SpaceX launching of the Falcon Heavy rocket and the attached Tesla Roadster, this game involves the player piloting the Falcon and its Roadster through the spacescape while dodging falling asteroids.

Version History
V.1: Basic movement implemented with pivot points
V.2: Added avoidable obstacles through 3 spawners and a point system
V.3: Replaced movement of car with hardcoded movement
V.4: Added coins and powerups as pickups. As well as full menu functionality with a main menu and pause menu. Health when hitting asteroid instead of insta-death after being hit.
V.5: Added music and sound when interacting with the game world, such as explosions and coin collection sounds
V.6: Added credits, re-done scoring, and a controls/backstory scene before game.

1. Characters
    a. The player will control the Falcon Heavy rocket while the Tesla Roadster swings around behind it.
    b. The “enemies” are falling asteroids that will be spawned at varying rates and locations.
2. Story
    a. After years of dedication and drive, Elon Musk and his labor of love are finally able to fulfill his recent goals – launching his car into space. After a successful launch, it is now your mission to make sure the Falcon and the Roadster can travel safely through the solar system.
    2.1. Theme
        a. We want to take the old “space age” feel of games like Asteroids and Space invaders through the use of pixilated sprites for that “retro feel”, as well as the use of music that takes a modern “space” feel with some “retro” elements to provide the same awe of space.
3. Story Progression
    a. This is one of those “infinite runner” types of games, the only difficulty progression involved is that the asteroids will speed up, spawn faster, and spawn in random locations.
4. Gameplay
    a. As stated previously, the player will pilot the Falcon around the screen while avoiding falling asteroids. To increase to the intricacy of this game,

the player will also have to pilot the Tesla roadster safely away from the asteroids while it swings tethered to the back of the Falcon.

4.1. Goal / Winning
   a. The goal of this game is to travel the farthest distance without "sustaining critical damage", which results from the Falcon or Roadster colliding with an asteroid. In this game, the primary goal is to travel farther with each run

4.2. User Skills
   a. The player is able to travel up/down/left/right around the screen. It is up to the player to be able to dodge around the asteroids as they fall across the screen.how to use controls, expected level of logic, etc.

4.3. Game Mechanics
   a. The fundamental mechanics of this game is similar to most infinite runner games. Obstacles endlessly spawn at progressively increasing rates. The novelty mechanic of this game is that the player has to dodge both with the Falcon, but with the Roadster, which swings like a pendulum behind

4.4. Items & powerups
   a. In the current version of the game, the Asteroid obstacles are the only objects in the game that are not controlled by the player. Future power-ups and other obstacles may be added in later versions

4.5. Progression & Challenge
   a. The falling asteroids spawn in random locations at varying rates, adding an element of unpredictability to the game.
   b. In addition to the player having to maneuver the Falcon between a steroids, the player must also have the forethought to navigate the "Roadster pendulum" behind them to safety as well.

4.6. Losing
   a. Losing is when the Falcon or the Roadster collide with a falling asteroid, and sustain "critical damage". In future versions, the number of collisions may be changed, but it is currently a "Game Over" after one.

5. Art style
   a. Sprites are created with scalings of 16px by 16px sprites, with real-world-appropriate color schemes. This gives the game a retro feel while being relatable to the modern context from which the game was inspired.
   b. Some of the animations of game objects are performed by Unity's UI for rotations. The remaining animations are created by Sprite Sheets that consist of 2-3 scenes of animation.
   c. The background is a pixel-art drawing of outer space created in order to match the aesthetic of the other sprites in the game
6. Music & Sounds
   a. The soundtrack for the game was created in Ableton using Massive and Battery (both Native Instruments tools). The original intention was for it to

consist entirely of 8-bit sounds, but after I made some progress in production, it seemed more appropriate to incorporate 8-bit sounds into a more modern style of music given the context of what our game was referencing. As a result, I spent a lot of time walking through tutorials for making bass wobble noises using digital synthesizers like Massive and Sylenth. The coin sound effect was made in Massive, following an 8-bit sounds preset tutorial for Massive on YouTube. The explosion and laser noise were both royalty-free samples that were found on SoundBible. There was also scripting involved for triggering different sections of the background music at certain score values, but I couldn't quite figure out how to get them play after the current loop played out completely as opposed to cutting off the current loop with a new one as soon a point value was reached. As a result, a lot of the user's that we tested on said the background music sound more cohesive if I just let the audio clip for background music play through from beginning to end.

7. Technical description

    a. Utilizing Unity's game libraries, we intend to create an engaging movement system utilizing vectors and colliders for the main gameplay loop. Power ups will use enabling and disabling player's attached scripts.

Personalized Learning Objectives:

Drake – Art & Animation:

At the start of the semester I claimed in my personalized learning objective that I wanted to improve my artistic abilities with a focus on animation. After working on this project for the 2D Game collab, I discovered that I wanted to focus more on sprite-focused pixel art. I was confident in my artistic abilities with traditional art, but creating digital artwork was an entirely different challenge. I wanted to keep it simple with sprites. In the first version of the game, the sprites were only 16x16 pixels. For the second half of the games development, I wanted to challenge myself even further by working with larger 64x64 sprites. I also wanted to touch on the animation aspect from my original objective, so I implemented my first sprite sheets. I was pleasantly surprised at how simple it was to create the sprite sheets and concatenate the sprites into a single animation. I am personally satisfied with how much I have learned this semester.

Alex – Scripting:

I want to explore game design as a career, I have creative ideas for games, but I wanted to expand my skills of designing games in the scope of scripting. It was a little difficult at first, seeing as I had never worked with Unity's engine or C# before, but working through many tutorials online and the large amount of documentation for the Unity Engine, I started making real progress. Our gameplay feature of the car went from something that

was thrown together in the Unity UI to a hard-coded movement system. My skills with Unity changed from cross referencing tutorials and documentation to free form and critical problem solving without any outside help. I'd say that I can confidently use Unity to create games now, and am satisfied with the progress I've made in learning how to code a game and with the Unity Engine.

Chris – Audio:

While I was responsible for a small portion of the scripting, my main area of focus was designing the audio for this game. I think this correlated directly to my personal learning objective for the class. I had intended to learn about sound design for sounds and music for the game using software outside of the resources that we were given in class. For the first game, the only audio that was involved was background music as well as the explosion sound upon collision with an asteroid. For the final project, I used multiple VSTs (Battery and Massive) in conjunction with Ableton in order to create the background music, as well as some of the sound effects. In a sense, I wanted to see how sound effects are designed before they are offered in the Unity Asset Store or through some other medium. I was able to accomplish this for this project by looking at tutorials for each VST, and then looking up tutorials/presets for the sounds that I was hoping to create (i.e. the Mario coin sound). While I used a minimal amount of premade sounds (I worked on creating a laser shot sound for a while, but I couldn't get it to the point where it was something that I was satisfied with), I made sure that the samples that I used like the laser shot were at least royalty free. On top of gaining some proficiency in C# with Unity, I think I achieved my goal of learning more about sound design and am satisfied with the song that I put together for the in-game sound track. While it was kind of labor-intensive, I thought the coolest part of designing the sound/soundtrack for the game was learning to use Ableton. At one point, I was sitting with drumsticks, playing out a drum beat on my kitchen table, then going into Ableton, and clicking out the pattern of the individual drum hits for my song.

Other Categories:

Platforms:

This game was initially designed with the intention of a mobile game, however, current controls were designed for PC play

Additional Features:

There are currently no additional features, however, there is the possibility for some in future versions.