Orbital Delivery Service

Michael Krcmarik

Andrew Rodman

Project Description

Orbital Delivery Service is a 2D moon lander style game where the player must land a cargo ship on various worlds at the intended landing zone while avoiding obstacles with limited fuel. The player controls the cargo ship by rotating the ship using auxiliary boosters and turning on and off the rear rocket thrusters. There will be a basic powerup system and a high score table to increase replayability. Upgrades will be available by picking up and collecting coins throughout each level and purchasing from the upgrade store. Players earn points based on fuel remaining at the end of each level.

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0		10/17/2016		10/17/2016	*Initial version*
1.1	Andrew Rodman	10/17/2016			Worked on Gameplay, art style, music, and technical description
1.2	Michael Krcmarik	10/19/16			Added/edited gameplay, art style, technical description, and marketing.
1.3	Andrew Rodman	10/21/16			Added some in game artwork and updated technical description
1.4	Michael Krcmarik	10/21/16			
1.5	Andrew Rodman	11/4/16			Added Artwork

Table of Contents

1. Gameplay

1.1. Goals

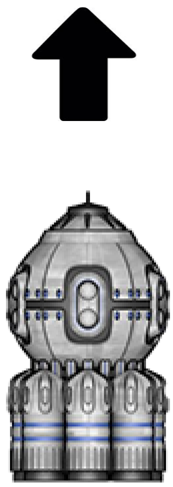
The overall objective in Orbital Delivery Service is for the player to land the cargo ship onto the designated landing spot in each level. The player has the ability to control the cargo ship by rotating the ship or moving it with a gas thruster. Gas is limited so the player must not waste too much or they risk crashing the ship. If the ship does not successfully land the level is over and the player must repeat the level to advance further through the game. The Player will also have obstacles in the way of the path to land that they must avoid. These obstacles can push the landing ship out of the way to prevent a successful landing. This involves the player having to time landings and learn obstacle patterns to pass each level. As the player advances through levels, the difficulty of landing the cargo ship ramps up.

1.2. User Skills

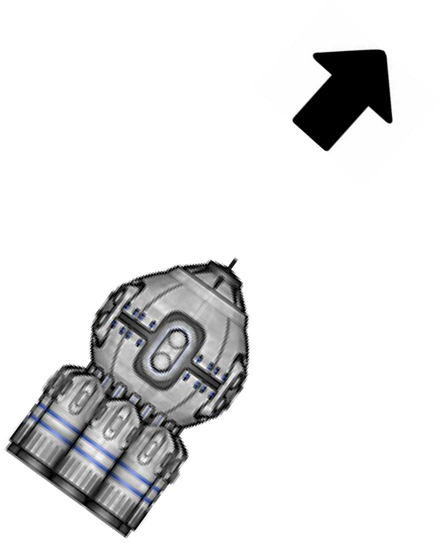
1. Key pressing (timing)
2. Avoiding Collisions
3. Identifying Patterns
4. Managing limited resources (gas)
5. Understanding basic physics

1.3. Game Mechanics

In all the levels gravity is pushing down on everything just like earth. The cargo ship will start on a starting platform away from the landing platform. The player can move the ship by pressing the gas thruster button which will be spacebar. The cargo ship can also be rotated clockwise by pressing ‘e’ or the right arrow. The cargo ship can also be rotated counterclockwise by pressing ‘q’ or left arrow. The cargo ship will go in the direction that it is pointing when the gas thruster is activated. This means the player must be careful and precise when angling the ship and deciding when to activate the gas.



The ship will move straight up when the gas is activated here.



The ship will move at this angle when the gas is activated here.

Landing the cargo ship is a simple idea, but this will be one of the most difficult things to do in the game. To land the ship, all the player needs to do is level the ship and let go of the gas. Some obstacles will block and push the player to prevent them from being able to land the cargo ship. This means the player will have to do more than just simply line up with the landing spot and let go of the gas. The player will have to navigate and around the obstacles and learn some of the patterns in which the obstacles move. The player must be quick and precise.

1.4. Items and power-ups

Throughout each level, there will be optional coins the player can pick up by moving the cargo ship close to them. These coins are used to purchase upgrades for the cargo ship. There will be a powerup store where all available power ups can be viewed and purchased. The ability to land faster, use less gas, and to use more powerful gas are things that can be upgraded by the player throughout the game.

There will also be several power ups within the level for gas replenishment, time boost, and ship repair. These will be hard to reach but can give the player much higher final scores if they have the skill to pull it off.

1.5. Progression and challenge

The difficulty will ramp up each level by creating more obstacles, longer distances between the starting platform and landing spot, moving landing platforms, and more difficult to reach coins. The player can combat the difficulty by upgrading the cargo ship's abilities and by playing more carefully.

Once the cargo ship successfully lands on the target, the player advances to the next level. The player receives 50 points for completing a level and the leftover gas amount is converted into points. For example, if the player completed a level and has 64% of their gas remaining, they would receive 114 points for that level.

1.6. Losing

The only losing condition for Orbital Delivery Service is if the cargo ship crashes. The ship is considered crashed if it touches the bottom of the screen or loses enough health by hitting objects in the level. Running gas to 0% percent is not considered a losing condition because the player still has the chance to land if they run out of gas mid air and land soft enough. When the player loses the level they are presented with a window that gives them a choice to restart that level or to quit the current game.

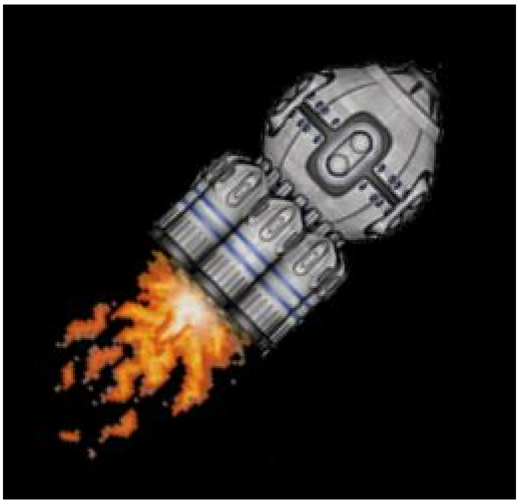
2. Art Style

Orbital Delivery Service is a 2D game with 2D sprites. Each level will have a unique setting and background. Some obstacles will have moving animations that can push the cargo ship out of the way. The backgrounds sprites will be implemented with parallaxing so as the ship moves around they shift their perspective and move to give an immersive effect. The levels will be created by having a large base image with various environmental components layered on top. There will be a camera that follows the ship's position in the level.

All art assets will be purchased from the Unity asset store (free) or hand-made in Photoshop or another editing software. The game will have a mostly hand drawn/cartoon feel that may have a retro look in its final form.

Here are a few pieces of in game artwork :

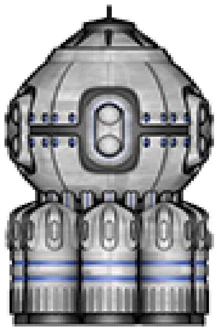
Cargo ship while thrusters are active



A coin that can be collected for upgrades:

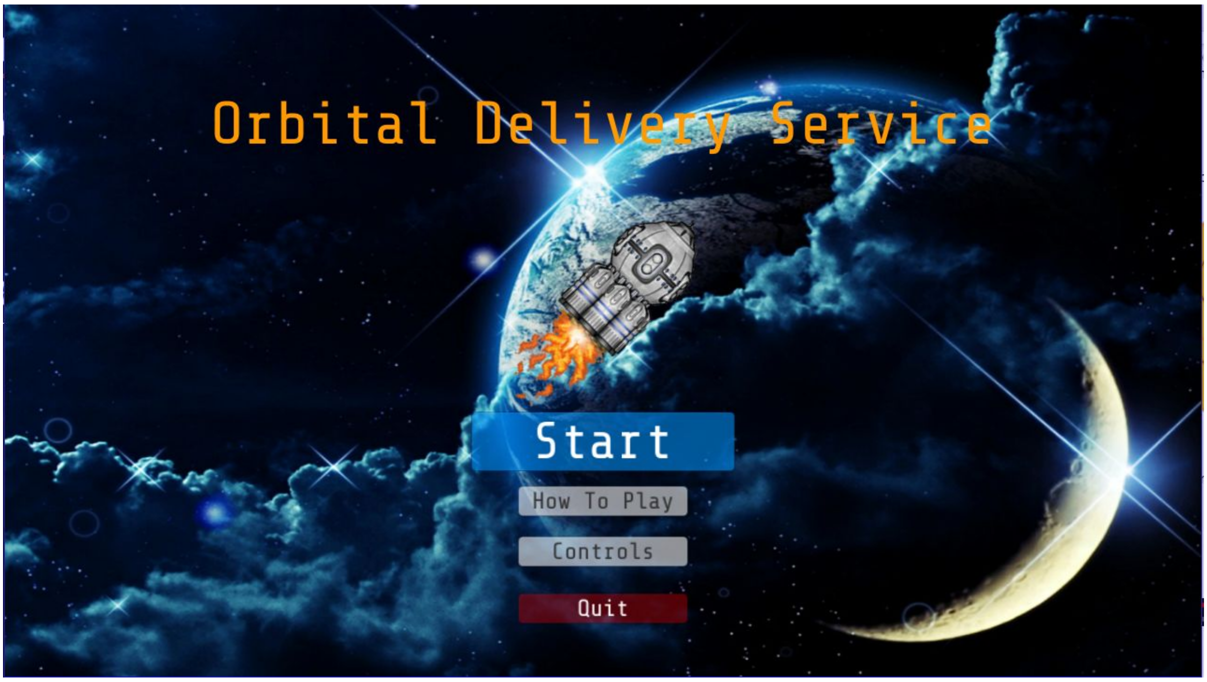
The cargo ship when thrusters are inactive:

Enemy UFO:

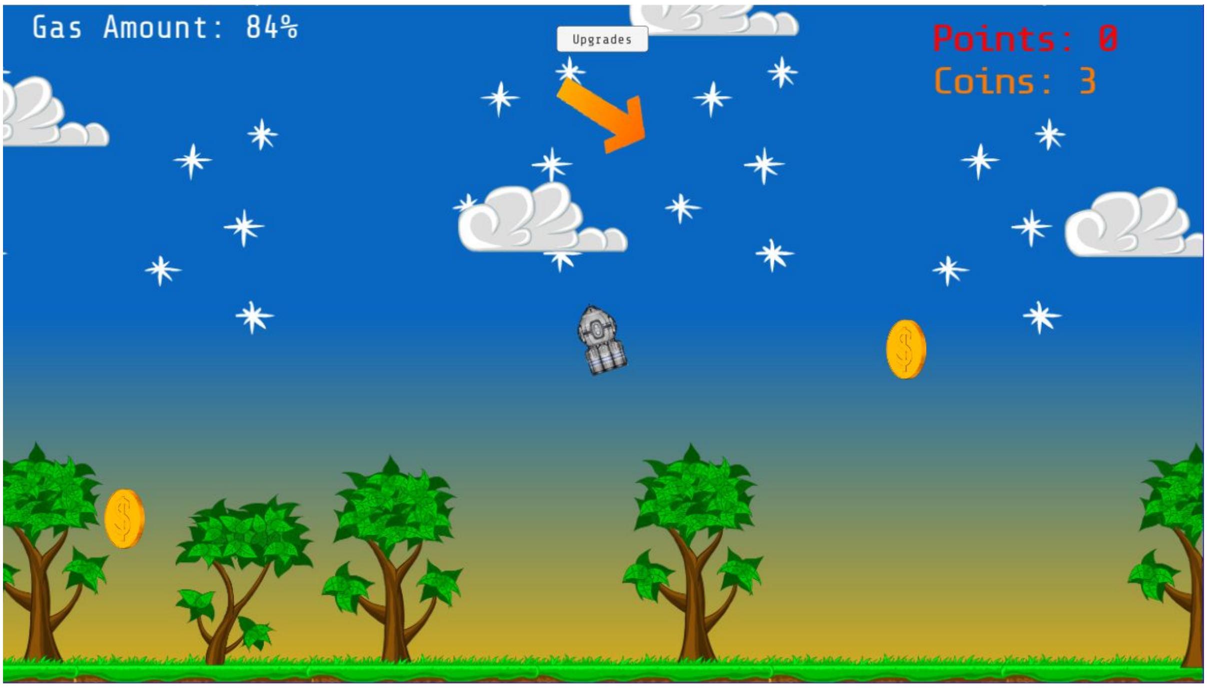


Screenshots from the actual game:

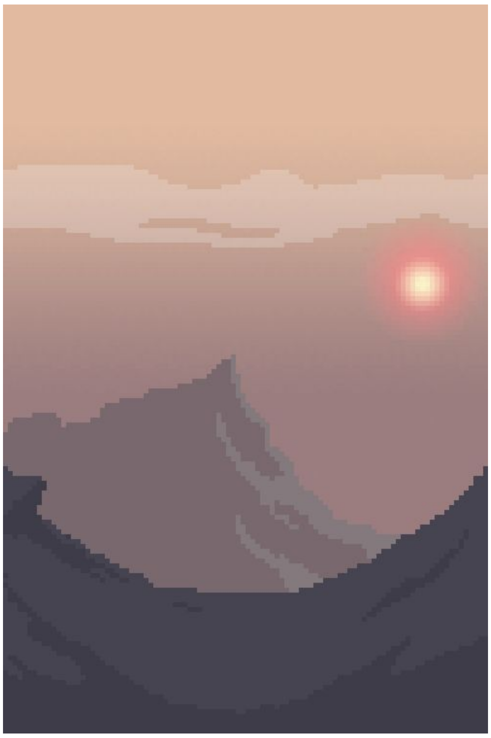
Menu:



In Game Level:



Some additional environment and landing pad inspiration:



Note: All art belongs to respective owners, these are just basis for our self created art.

3. Music and Sounds

The music will be high quality electronic audio tracks. The technical description section has more details with the technology used to create the audio. The music will be somewhat fast paced and have a Sci-Fi feel to it which should complement the gameplay. The instruments will have lots of delay and reverb effects to give it a spacey type feel.

The sound effects are important in this game for feedback for the user. Whenever the player presses the gas button and there is still gas, a powerful rocket thruster sound will happen until the player lets go of the gas button. There will also be sound effects when the player picks up a coin, lands successfully, crashes, or when gas is close to empty.

4. Technical description

Initially Orbital Delivery Service will be a standalone PC game developed in Unity3D. Since Unity allows for building on many platforms we may release the game for Mac or one of the console stores if it becomes a polished product.

For project management we will be using HacknPlan, Github, and Google Drive. We will be creating tickets in HackNPlan based on this document and any tweaks we encounter during development and chunking out tasks into individual pieces for an agile style approach.

For ship movement we will be using a Rigidbody2D with a controller script attached that will handle the rotation and movement from the thrusters and managing user input. The game will initially be designed for use with the arrow keys but we will add controller input if time allows. The background parallaxing will be accomplished by adding many different sprite layers then using a script that shifts them relative to the movement of the player. We will also create pre-fabs for our powerups that will contain the mesh, sound effect, and other components required to affect the state of the game (more gas, repair ship).

Music and sound effects will be made using Ableton Live 9 Digital Audio Workstation with Native Instruments Massive synthesizer plugin. Massive is a software synthesizer capable of producing a wide range of custom sounds while being not difficult to use.

5. Marketing & Funding

For feedback and for practice on future projects we will record a few gifs/videos of our gameplay to try to get some feedback from the gaming community on social media.

Postmortem: Released a few gifs of gameplay on Twitter and got some likes and retweets in response.

5.1. Demographics

The demographic for the game is everyone who likes casual but fun games. It has no violence other than an exploding ship graphic if the player crashes so it is appropriate for most if not all ages. It has simple controls and will be based on timing and execution rather than requiring complex button combinations or a long learning curve.

5.2. Platforms & Monetization

Initially the game will be a standalone windows executable built from Unity that will be distributed for free through Google Drive or itch.io. Ads may be added later or IAP (in app purchases) for power-up packs or other features to monetize it if released through an application store.

5.3. Localization

Initially the game will just support English, but multilingual support could be added if released to Steam or another large software broker.

6. Other ideas

- High Score Table
- How to Play button