STARFURIOUS

Description

Starfurious provides a retro space shooter entertainment experience, inspired by old school stand-up arcade machines of similar theme common in the late 1970's and early 1980's. Shoot at and destroy a never ending advance of alien spacecraft while dodging incoming missiles and enemy ships. Do well, and achieve a place of notoriety on the high scores board.

Create custom flight paths for enemies using an easy to understand 2D path editor, also included in the project. High scores functionality is also included, providing functionality to capture high scores, player initials, and to set default values.

The complete project includes full, well document C# source code.

How To Play

Play is simple and straight-forward: Move left and right(arrow keys) and press Fire(spacebar) to destroy enemy ships. Avoid both the enemies and their missiles. Basic enemies require a single hit to be destroyed while more advanced enemies require up to three hits.

Once all your ships have been destroyed, it's game over!

Customization

Path Editor

Creating custom paths and applying them to enemies offers a fantastic way to create different and interesting opponents. The scene "PathMakerScene" is built specifically for creating and testing your flight path creations. Data should be stored in the "Data" directory.

Creating and editing paths is a fairly easy process. Start by:
1). Open the PathMakerScene Scene
2). Select the RFPathManager gameobject.

At this point, you may Create a new Path or Load and edit an existing path.

RFPathManager Editor overall options:

- Create Path: Initiate and start editing a new path
- Load Path : load a path for editing
- Save Path : save the current path data
- Clear Path : clear the current path data
- Edit Mode : When in edit mode, left clicks in the scene view will create a new path segment and add it to either the current path head or tail.

- Add To: Head or Tail : When adding path segments, add to either the head segment or tail segment.

RFPathManager Editor Segment Information:

- “S” button: Click to select the current segment
- Seg Type Drop down: Sets the line segment type
    - Fixed: Segment step point locations are programatically set
    - Straight : Segment step points are set in a straight line
    - Bezier Curve: Segment steps are calculated using a bezier curve equation. Control points are used to modify the curve
    - Catmul-Rom Spline: Segment steps are calculated using a Catmull-Rom spline formula. Control points and alpha value are used to modify the curve
- Segment Steps: The number of points computed for a segment. When set as a curve, more points will result in a smoother transition for a path traveller.
- Alpha : Alpha value to use when the segment type is a Catmull-Rom spline.

Path Editing

- Create a new path or select an existing path for editing using the buttons on RFPathManager (as indicated above)
- Click and drag an endpoint to update its location. Endpoints are marked as white dots.
- Click a segment step to select a segment; segment steps are marked as red dots when not selected, and green dots when selected.
- Line segment types may be selected under SEG TYPE in the RFPathManager editor window.
- Segment Steps may be changed in the RFPathManager editor window. Segment steps represent the number of points in a particular segment. The more points available, the smoother the transition between points, especially when the segment is a curve.

Testing the path:

- Select the PathTraveller game object in the scene hierarchy.
- Set the Start Path to your newly created / edited path
- Ensure AutoStart is set to true (on the PathTraveller component)
- Press Play.
- Adjust move speed for

.

High Scores

To manage high scores data, open the MAIN Scene and Select the RFHighScores game object. Select the RFHighScoresManager component to:

- Clear Scores: Press the “Clear Scores” button located on the bottom of the RFHighScoresManager component.
- Set Default High Scores:
    - Clear Current Scores
    - Update the values for the “Default Scores”
    - Press the “Set Default Scores” at the bottom of the RFHighScoresManager component.

High scores are rendered by the component “HighScoreRenderer” located in the prefabs directory. Change the font settings and color for a unique look.

Enemies Value / Scoring

Enemy score values and life points may be changes on the enemy prefabs by updating values on the EnemyLogicController component.
Score Value: The value to add to the player’s score when the enemy dies. Minimum required value is 1.
Life Points : The number of times it takes for a player’s missile to collide with and kill the enemy. Minimum required value is 1.

Be sure to update the Start scene enemy values / hits information to reflect alternative scoring and hitpoints. To do so, open the start scene, expand the Canvas gameobject in the hierarchy, and update the label values in found in the EnemyData_? game objects.

Other

Enemy and player ship graphics are an easy change out. Enemies consist of prefabs in the Prefabs directory; simply update the sprite renderer. The player is a game object in the root of the Main scene hierarchy. Again, simple update the sprite in the sprite renderer component.

Project Setup Requirements / Information

Input Manager

- Input manager must provide axis setup for Fire1, Horizontal, Vertical

Standard Assets

- A subset of Unity’s Standard assets is included to support the CrossPlatformInputManager and mobile updates.
- **Known Issue**: When switching build targets, Unity throws a warning and a null exception error on code in MobileControlRig.cs. That code is included in standard assets, and the errors have no effect on the output build or during Editor Play testing.

Attributions

Ship graphics and missiles were constructed used graphics from http://kenney.nl/
Font: Upheaval Created by Kent Bryant

Change Log

Version	Date	Notes
1.0	11/16	Initial Release
1.0.1	03/17	● Added Mobile input control ● Updated to work with CrossPlatformInputManager ● Bug Fix: Player could move while dead, causing respawn location to appear different than death location.
1.0.2	12/18	● Updated project to 2018.2 ● Bug Fix: Crossfade on intro scene did not always fade canvas groups to 0 and 1, resulting in “ghosted” panels ● Updated code to fix deprecation warnings
1.0.3	3/19	● Updated Projet to 2018.3 ● Removed extraneous files
1.0.4	7/19	Updated to Unity 2019.1