|  |  |  |  |
| --- | --- | --- | --- |
| **CS102** | **Spring 2015/2016** | Project Group | 5H |
| Instructor: | **Özcan Öztürk** |
| Assistant: | Dorukhan Arslan |

|  |  |  |
| --- | --- | --- |
| **Criteria** | **TA/Grader** | **Instructor** |
| Presentation |  | a |
|  |  |  |
|  |  |  |
|  |  |  |
| Overall |  |  |

~ 2D Rocket Game ~

**Rocket Fool**

**Ömer Ekmekcioğlu, Alperen Erkek, Levent Berk Güngen,**

**Hakan Yaman Sözer, Hakan Türkmenoğlu**

|  |
| --- |
| Requirements & U.I. Report  ( First Draft ) |

# Introduction

## Initial Problem Discussion

Although it is a relatively new field today, space travel is expected to be an integral part of mankind’s future. Therefore, learning about it would be beneficial to everyone, especially if it increases the interest and investments in space programs. However, the physics involved in space flight can make it seem unearthly and difficult to visualize.

Science fiction films that involve space travel often show spacecraft zipping from one planet to another in (almost) straight lines. Although taking such straight paths is intuitive and theoretically possible, in practicality it is extremely inefficient for spacecraft (at least the ones of today that use propellants) because of the effects of gravitational forces.

In order to help space travel receive the attention it deserves and to prevent the misconceptions about it, it needs to be taught efficiently. A fine way to do this would be to show a simulation, or to boldly go a couple steps further and exhibit a game. As often suggested, well-designed educational games are great for teaching because they can make the teaching material both easier to understand via the aid of technology and more interesting via the element of fun. With this cause in mind, the Rocket Fool team plans to create this product.

## Product Overview

The product is

* a computer game, which can fit into the genres “simulation”, “educational” and “puzzle”;
* primarily about piloting a rocket from planet to planet in a 2D space environment with realistic gravitation but reduced distances while managing additional concerns like fuel consumption and avoiding obstacles;
* a desktop application that can be downloaded from an appropriate website, because this format is good for games that do not require online connection and have immersion;
* targeted to a general audience although it is expected to be most beneficial for high school or university students;
* to be developed using game design wisdom acquired through research (see References & Found Research Material) and the examination of existing products with similar goals and features (see 2. Similar Products Review).

# Similar Products Review

## *Kerbal Space Program*

*Kerbal Space Program* is an award-winning, independently-developed simulation game that allows players to control various aspects of an aerospace program run by little green men called “Kerbals”, from financial policy management to multi-staged space craft design, to flying these crafts across a virtual 3D solar system (see [1] for more information).

The game has highly realistic models of gravitation, mass distribution, thermodynamics, air resistance, collisions, material stress, and more. Its interface includes tools like time warping, patched conics, and quick-saving at any location. It serves as an excellent example of a game for space enthusiasts although it can be noted for having a steep and possibly-overwhelming learning curve (it *is* rocket science after all). Rocket Fool’s game is meant to be substantially easier while remaining largely realistic.

The game also has a version, *KerbalEdu*, specifically designed for classroom environments, reinforcing the idea of how games can be great for teaching real-world physics.

## *Into Space 2*

*Into Space 2* is a popular, upgrade-based flash game that involves piloting a rocket during its ascent into orbit in a 2D atmosphere. It has additional gameplay elements like power-ups, obstacles, achievements, and missions to successfully enrich what otherwise would be a very repetitive and simple task. The game is also notable for its good use of sound effects and pleasing graphics (see [2] for more information). Rocket Fool’s game is expected to be more realistic than *Into Space 2* and it will have much less of a focus on the launch phase of a mission.

## *Angry Birds Space*

As a part phenomenal mobile device game series *Angry Birds,* the game *Angry Birds Space* is based around launching birds to attack pigs in small, 2D space environments with little planets that affect the birds’ trajectories with their (extraordinarily) strong gravitation. It makes players have to think about the effects of gravity while trying to certain places in space in order to solve the puzzles. The levels are quite short and depend mostly on the launch angle and speed, as with other *Angry Birds* games (see [3] for more information). Rocket Fool’s game is expected to have longer levels with much more focus on in-flight decisions.

# Product Features

## Core Features

* Approximately 10 levels
  + The levels have varying objectives to keep things interesting such as but not necessarily space exploration, cargo delivery, satellite positioning, racing other rockets, rescue missions, or searching for resources.
  + Objectives will be marked with beacons.
  + The levels are connected by a story line.
  + The level maps have several planets, moons, and obstacles.
  + Bodies have very high densities in order to reduce the otherwise astronomical distances while retaining strong gravitational forces.
  + No procedural generation
  + The window will be a “camera” centered on the rocket on all times while the entire level does not have to fit on the screen.
* The controls: The rocket can be tilted using the left and right arrow keys. The up and down arrow keys can be used to increase or decrease thrust respectively.
* The physics engine:
  + Realistic gravitation
  + Decrease in rocket mass as fuel is burned
  + Collisions: The rocket will be destroyed if it collides with anything, including planets (landing can only be initiated by heading to the right spot).
* Finite rocket fuel and fuel management (to make players care about efficiency)
* Information windows at starts of levels to teach about the game and the physics and mathematics involved:
  + Topics covered for sure are: Newton's law of universal gravitation, elliptical orbits and eccentricity, the Hohmann transfer, the bi-elliptical transfer, gravitational assists, gravitational potential energy and escape velocity.
  + Topics that would ideally be covered include: Newton’s laws of motion, Kepler’s laws of planetary motion, the delta-v concept, uniform circular motion.
* A toggled in-game display to show information like the forces acting on the rocket and possibly equations
* Take-offs and landings shown as basic cut scenes that the players do not control
* Trajectory estimations: The position of the rocket after X minutes can be calculated by the program using patched conic approximation, possibly while displaying the equations, and shown.
* Progress saving
* A pause menu

## Desirable Features

* Take-offs and landings shown as basic cut scenes
  + There will be multiple good or bad take-off/landing scenarios based on the player’s decisions made on a window before the launch (eg. how much cargo to pack, how much initial thrust to give, approach speed, etc. )
  + These decisions will be made on a Pre-Launch Window where the player will configure the rocket according to suggestions made by the game and what the player thinks would do.
* Sound: Sounds for thrusters and for collisions/crashes. Music and interface-related sound effects can be added depending on what is available. They will be composed or used with appropriate licensing. If there is sound, a muting option will also be provided.
* Score keeping means (eg. based on remaining fuel or mission duration)
* Time-warping: speeding up time to quicken the travel over long distances (if the develops decide that it is worthwhile)
* Pop-up boxes with the “Did you know?” sort of interesting facts about space travel
* Enabling/disabling full-screen mode

## Optional Features & Possible Future Additions

* Easter eggs
* A branching story line
* Multi-stage rockets
* Additional levels
* Power-ups
* Tracking of the rocket’s location across maps (like a trail)
* Some planets that have atmospheres with basically-modeled atmospheric drag or realistic atmospheric drag
* Rocket customization/upgrading options (in terms of aesthetics or engineering)
* A level editor
* Multiplayer features
* Achievements & stat keeping (eg. number of crashes)
* In-Game shop
* Wormholes / black holes
* Adjustable difficulty (eg. different engine efficiencies)
* Limited electrical power

## Not Planned Features

* 3D graphics
* Thermodynamics
* Realistic lighting
* Orbital decay other than atmospheric drag
* Extravehicular activities
* Plane piloting
* Changing engine ISP’s
* Telemetry issues
* Random equipment failure

## Stylistic Choices

* Game title, slogan: (undecided)
* Game’s overall attitude: humorous, informative, casual
* Graphics: Cartoonish, consistent graphics with basic animations for details like rocket plumes are preferred. They will be drawn or used with appropriate licensing. A rounded, simplistic, smooth art style with block colors is preferred.
* Sound: Medium/high-quality sound effects are preferred. Possibly, sound effects for aesthetics (eg. clicking buttons) will also be included.
* Setting: The game is set in a not-so-distant future where space travel is uncommon but not as rare as today. Planets may be based on our solar system in order to give information about them (or not).
* Story goals: The story will have at least enough depth to give the player a sense of purpose and a slight curiosity about what happens next. Easter eggs and funny references may be additional means to enrich the player’s experience.
* Characters: Characters may be loosely or strictly based on Bilkent University students and faculty, if the needed consent and permissions are acquired. Some aliens may too be included. Characters are not expected to be deeply developed throughout the story.
* Music: Music that fits the setting and attitude will be sought.

# User Interface

The user interface is designed to be conventional and intuitive via the following:

* Once it is run, the game will not require any inputs until the Starting Screen appears.
* Starting Screen
  + It will serve as a quick one-picture introduction to the game.
  + Clicking/pressing a key will proceed to the Main Menu.
* Main Menu
  + The menu will have buttons to each of these screens: Level Selection Screen (play button), credits, quit, options. There will also be a mute button in the corner.
  + The user will left-click a button once to use it. No other inputs apply.
  + Menu design will match the art style of the game and be intuitive to use.
* Options Menu Screen
  + These options will be provided with single-left-click buttons: full-screen, erase save file, mute, back to menu
* Credits Screen
  + All necessary accolades, disclaimers, and etc. will be displayed. There will be a single-left-click back button.
* Level Selection Screen
  + There will be a button for each level and the back button, all single-left-click buttons.
* In-Game Screen
  + The rocket can be tilted using the left and right arrow keys. The up and down arrow keys can be used to increase or decrease thrust respectively.
  + Mission objectives will be displayed on the upper left corner of the screen.
  + Mission objective locations will be highlighted by beacons/crosshairs.
  + Stats, like the remaining amount of fuel will be displayed on the upper right corner.
  + The bottom-middle part of the screen will have a toolbar for various single-left-click buttons (see other bullet points).
  + The window will be a “camera” centered on the rocket on all times while the entire level does not have to fit on the screen.
  + The camera will have 2-3 zoom settings, each centered on the rocket. Alternatively, a mini-map may be used instead. The buttons for zooming, as well as the mini-map would be located at the bottom middle. The mini-map can be minimized to be re-opened from the toolbar.
  + The camera may rotate only by 180 degrees only when toggled by a button on the bottom middle. This would help confused players regain their orientation.
  + The button to toggle the overlay to display forces, as well as the one to calculate and display the rocket’s trajectory will also be on the bottom middle.
  + Trajectory estimations will be shown via patched conics.
  + Pressing “Esc” or the spacebar will open the pause menu, which offers single-left-click buttons for un-pausing, returning to the Main Menu, and muting the game.
  + Additional text will appear as pop-up windows with single-left-click buttons on them to close them.
* Cut scenes
  + The cut scenes are expected to be less than 30 seconds long and users will be shown and given the option to skip them using the space bar.
  + Dialogue will appear in speech bubbles.
  + The cut scenes may proceed as comic panels or basic animations.
* Pre-Launch Window (if implemented)
  + Pressing “Esc” or the spacebar will open the pause menu, which offers single-left-click buttons for un-pausing, returning to the Main Menu, and muting the game.
  + A vertical slider on the right side of the screen will be clicked and dragged by the player to decide on the initial thrust given to the rocket.
  + Horizontal sliders arranged in a vertical list will be on the left side of the screen, with which the player adjusts values such as the amount of resources to pack.
  + A large single-left-click launch button in the middle of the screen will start a launch cut scene.

# Conclusion & Reflections

This report describes the motives for designing this game, which are making it easier to learn orbital mechanics and other real world physics concepts related to space travel, as well as increasing public interest in space travel because it is an integral part of the future. The report outlines all the planned features for the game and introduces its user interface with sufficient detail to estimate the player’s overall experience. Additionally, the report highlights existing games with similar purposes and features to show the feasibility and desirability of this project.

Assuming that the core and desirable features are successfully implemented, upon completing the game, all players are expected to have greatly increased familiarity with and competence in several principles of space flight and gravitation. The game is also projected to be an entertaining experience for the player that may kindle their interest in space flight.

# References & Found Research Material

## References

1. Wikipedia. (2016, February 27). *Kerbal Space Program*. Retrieved February 27, 2016, from https://en.wikipedia.org/wiki/Kerbal\_Space\_Program
2. Barbarian Games. (2012, October 04). Into Space 2. Retrieved February 28, 2016, from http://www.kongregate.com/games/BarbarianGames/into-space-2
3. Wikipedia. (2016, January 23). *Angry Birds Space*. Retrieved February 27, 2016, from https://en.wikipedia.org/wiki/Angry\_Birds\_Space

## Found Research Material

* Harbour, J. S. (2012). *Beginning Java SE 6 Game Programming* (Third Edition ed.). Boston, MA: Course Technology.
* Rosenfeld Media. (2012, May 17). 10 tips for building a better game. Retrieved February 23, 2016, from http://www.creativebloq.com/inspiration/10-tips-building-better-game-5126304
* McMillen, E. (2009, December 30). Opinion: Indie Game Design Do-s and Don't-s: A Manifesto. Retrieved February 28, 2016, from http://webcache.googleusercontent.com/search?q=cache:http://www.gamasutra.com/view/news/117521/Opinion\_Indie\_Game\_Design\_Dos\_and\_Donts\_A\_Manifesto.php&gws\_rd=cr&ei=a8jSVoG0EYH5UN6ulsAE
* Giancoli, D. C. (2009). *Media manager Physics for scientists & engineers with modern physics, 4th ed. Giancoli.* Upper Saddle River, NJ: Pearson Education.

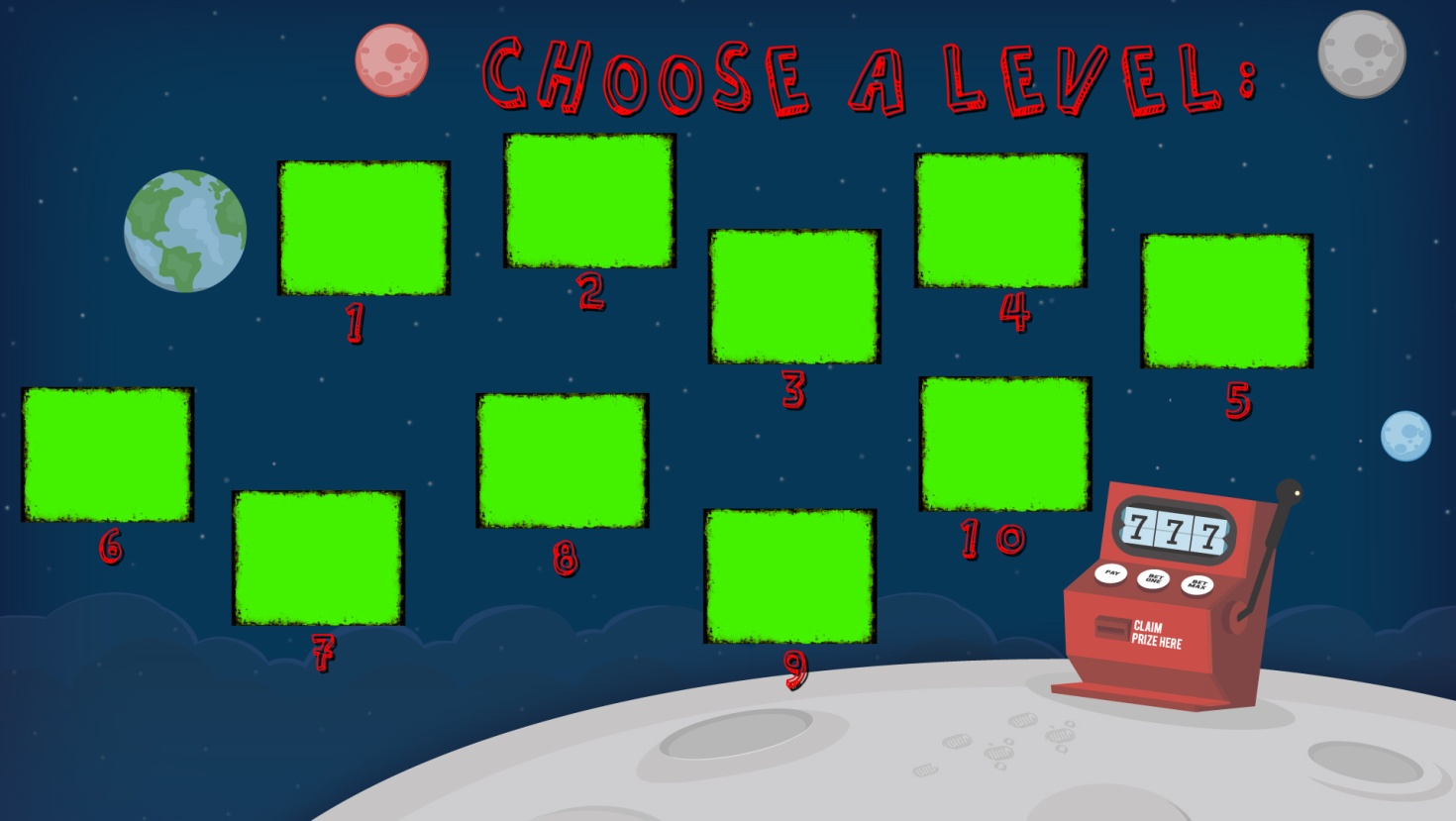
# Concept Art



Starting Screen (made on *Adobe After Effects*)



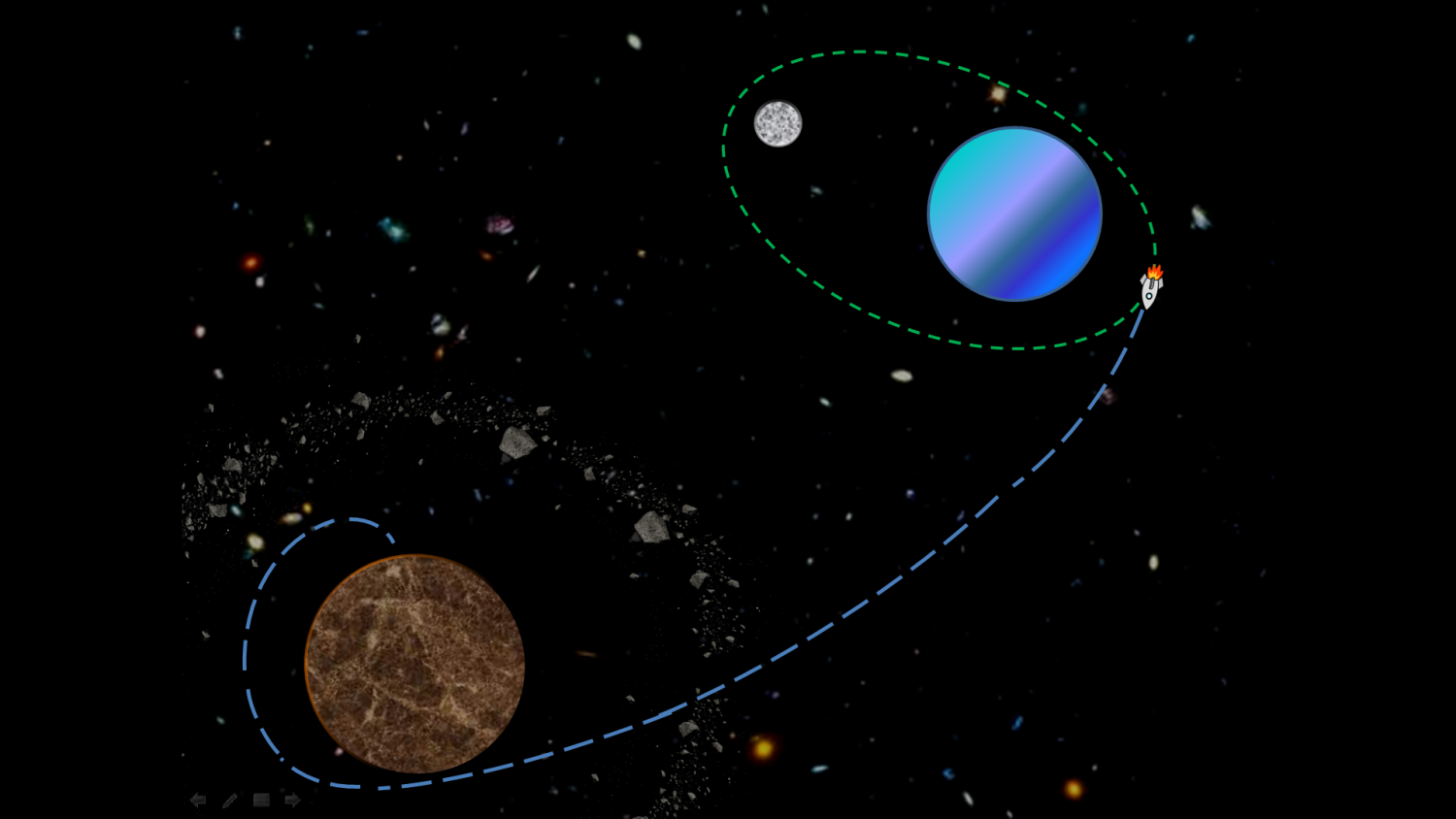
Main Menu Screen (made on *Adobe After Effects*)



Level Selection Screen (made on *Adobe After Effects*)



In-Game Screen interface (early concept idea made on *Adobe After Effect*s)



An example level (early concept idea made on *Microsoft PowerPoint*) exhibiting a blue trail and green patched conics