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

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

~ 2D Rocket Game ~

**Rocket Fool**

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

**Hakan Yaman Sözer, Hakan Türkmenoğlu**

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

# Introduction

## Initial Problem Discussion

Although it is currently a relatively new field, space travel is expected to be an integral part of mankind’s future. Today’s space sector is rapidly growing and with the recent rise of private space companies, space flight is becoming an increasingly larger part of our daily lives. Therefore, it would be beneficial to increase public understanding of the principles of space travel, particularly orbital mechanics, while preventing the spread of misconceptions.

Many science fiction films depict space travel with spacecrafts zipping from one planet to another in (almost) straight lines. While such routes seem intuitive, in reality they would be extremely inefficient; spacecraft have to take gravitational forces into account when planning their routes and this usually results in them drawing various arcs across space.

Inspired by the desire to make space flight easier to understand for everyone, the Rocket Fool team plans to create a video game about piloting spacecrafts in realistic gravitation. Well-designed video games can serve as great supplementary material for teaching because they make concepts easier to visualize and more interesting to learn via the element of fun.

If the game’s learning and enjoyment factors are well-developed and well-balanced, it may additionally stimulate the players’ interest in the space sector and thus benefit the development of the space industry, as other games have done in the past (see [1]).

## Product Overview

* The product is a computer game that can fit into the genres “simulation”, “educational” and “puzzle”.
* It is created with the aim of making various principles of space travel, such as orbital mechanics, easy to understand in an entertaining way, with the additional aim of stimulating interest in space flights.
* It is primarily about piloting a rocket from planet to planet in a two-dimensional space environment with realistic gravitation but reduced distances while managing additional concerns like fuel consumption and avoiding obstacles.
* It is a desktop application that would be downloaded from an appropriate website, because this format is good for games that do not require online connection and are immersive.
* It is targeted to a general audience although it is expected to be most beneficial for high school or university students, as well as for space enthusiasts.
* It is to present comprehensive amounts of information on the subjects that it is teaching.
* It is to be developed using game design wisdom acquired through research (see References) and the examination of existing products with similar goals and features (see 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 three-dimensional solar system (see [2] 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 made for space enthusiasts although it is notable for having a steep learning curve: Many players tend to be overwhelmed by the large number of factors to consider all at once and often several hours of effort involving multiple attempts are needed to be successful (it *is* rocket science after all).

Because of Rocket Fool aiming to teach specifically about the matters of gravitation and piloting to a general audience, the team’s game is meant to be substantially less overwhelming than *Kerbal Space Program* via the use of pre-built spacecraft, the beneficial elimination of the third dimension (see 3.1.1), and the provision of extensive tutorial/teaching content.

## *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 [3] 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 [4] 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

### Levels and Maps levels

* + There are approximately 10 levels.
  + A level’s map has several planets, moons, and/or obstacles that are all within the same one or two solar systems.
  + The maps are in two dimensions. This simplification offers the advantage of making it easier to keep track of the spacecraft’s trajectory as well as making piloting easier because confusing directions is much less likely. Orbits and maneuvers are not affected by the absence of the third dimension; they are only limited to one plane, which is a possible scenario in three dimensional space. The only problem with the third dimension’s absence is not being able to dodge obstacles so easily.
  + Celestial bodies’ densities or the gravitational constant will be higher than normal in order to reduce the otherwise astronomical distances while retaining strong gravitational forces.
  + No procedural generation is used to build maps.
  + The entire map of a level does not have to fit in the screen; the viewpoint of the player is fixed on the spacecraft like a camera following it from afar.
  + Spacecraft are able to cross over the edges of the maps, although they are then considered lost in space and uncontrollable.
  + The levels have various different objectives such as space exploration, cargo delivery, satellite positioning, racing other rockets, rescue missions, or searching for resources.
  + Locations involved in mission objectives are marked with beacons.
  + The levels are connected by a story line.
  + Some levels are long enough to have checkpoints in them.
  + Each level should take between one and fifteen minutes to complete if the player is successful in their first try.
  + At the starts of levels, information windows will appear in order to teach about the game mechanics and the physics and mathematics involved:
    - Topics covered 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.
    - Additional topics that may be covered are Newton’s laws of motion, Kepler’s laws of planetary motion, the delta-v concept, and uniform circular motion.

### 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 is modeled with forces changing by tiny increments with distance. As an exception the value of the gravitational constant or densities of planets may change, in order to reduce map sizes.
  + The decrease in spacecrafts’ masses as fuel is burned is modeled momentum changes will be appropriately modeled.
  + Collisions: The rocket will be damaged or destroyed if it collides with anything.

### Take-Offs and Landings:

* + Take-offs and landings are shown as basic cut scenes that the players do not control.
  + If a spacecraft approaches a large body with too much speed, it will crash instead of landing.

### Other Core Features

* + Finite rocket fuel is provided to make players learn more about fuel management and efficient travel.
  + Some levels may involve time constraints, making fast and well-planned travel necessary.
  + A toggled in-game display option is to be implemented to show information like the forces acting on the rocket and possibly live-updated equations of ongoing phenomena such as forces or orbit eccentricity.
  + Trajectory estimations: The position of the rocket after X minutes can be calculated by the program using patched conic approximation, possibly while displaying the relevant equations.
  + The game has a menu system to access different levels and other locations like a credits page.
  + Score is kept depending on factors like a player’s remaining fuel or time at the end of a mission, the completion of optional objectives.
  + Progress is saved automatically by the game in terms of (any) scores, levels completed and checkpoints reached.
  + A pause menu is offered in-game to provide players a chance to pause and change game options (see User Interface).

## Desirable Features

### Pre-Launch Window

* + Take-offs and landings shown as basic cut scenes depending on players’ pre-launch configurations of the rocket.
  + 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 (See User Interface\*\*) where the player will configure the rocket according to suggestions made by the game and what the player thinks would do.
  + Some of these decisions will influence the behavior of the spacecraft in-game (eg. the mass is conserved).

### Sound

* + Sounds for thrusters and for collisions/crashes may be played.
  + Open-source music can be provided with proper citation and licensing.
  + Interface-related sound effects.
  + A muting option will also be provided.

### Other Desirable Features

* + Time-warping: A toggled feature to speed up time to quicken the travel over long distances can be implemented.
  + Pop-up boxes with the “Did you know?” sort of interesting facts about space travel, like space travel history.
  + Multiple game window sizes and a full-screen mode option can be provided.
  + Score evaluation: The player’s score can affect various aspects of the game like there being a score-requirement to unlock a level.

## Optional Features & Possible Future Additions

* + Easter eggs may be included for enriching the player experience.
  + A branching story line for enriching the player experience.
  + Multi-stage rockets may be included.
  + Power-ups affecting game mechanics can be added.
  + Tracking of the rocket’s location across the map (like a trail) can be done.
  + Some planets that have atmospheres with basically-modeled atmospheric drag or realistic atmospheric drag can be added.
  + Rocket customization/upgrading options (in terms of aesthetics or engineering) can be added.
  + A level editor where players design their own maps and objectives can be added.
  + Multiplayer features can be added, introducing more possibilities such as racing or sharing player-designed levels.
  + Achievements & stat keeping (eg. number of crashes) can be done.
  + An in-game shop for rocket upgrades can be added.
  + Wormholes / black holes can be added to spice up game mechanics.
  + Adjustable difficulty (eg. different rocket engine efficiencies) can be provided to address players with different skill levels.
  + Electrical power and solar panel management can be another limitation.

## Not Planned Features

* + 3D graphics (see 3.1.1) will not be implemented.
  + Thermodynamics will not be implemented, although getting too close to a star would have the consequence of exploding.
  + Realistic lighting will not be implemented.
  + Orbital decay other than atmospheric drag will not be implemented.
  + Extravehicular activities will not be implemented.
  + Piloting planes in atmospheres will not be implemented.
  + Changing engine ISP’s will not be implemented.
  + Telemetry issues such as signal delay or occlusion will not be implemented.
  + Random equipment failure will not be implemented.

## Stylistic Choices

* + Game title; slogan: *Rocket Fool; “Fly you fools!”*
  + 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 (if implemented): Medium/high-quality sound effects are preferred. Possibly, sound effects for menus (eg. clicking buttons) will also be included. Music that fits the setting and attitude will be sought, for enriching the player’s experience.
  + Setting: The game is set in a not-so-distant future where space travel is uncommon but not as rare as today. Planets are loosely based on our solar system in order to give information about them and fictional planets are introduced as well.
  + 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.
  + Story outline: \*\**\*story\*\*\**

# User Interface

The interface is designed to have conventional menus and intuitive controls. Figure 4.0 displays the connections between the various windows of the game.

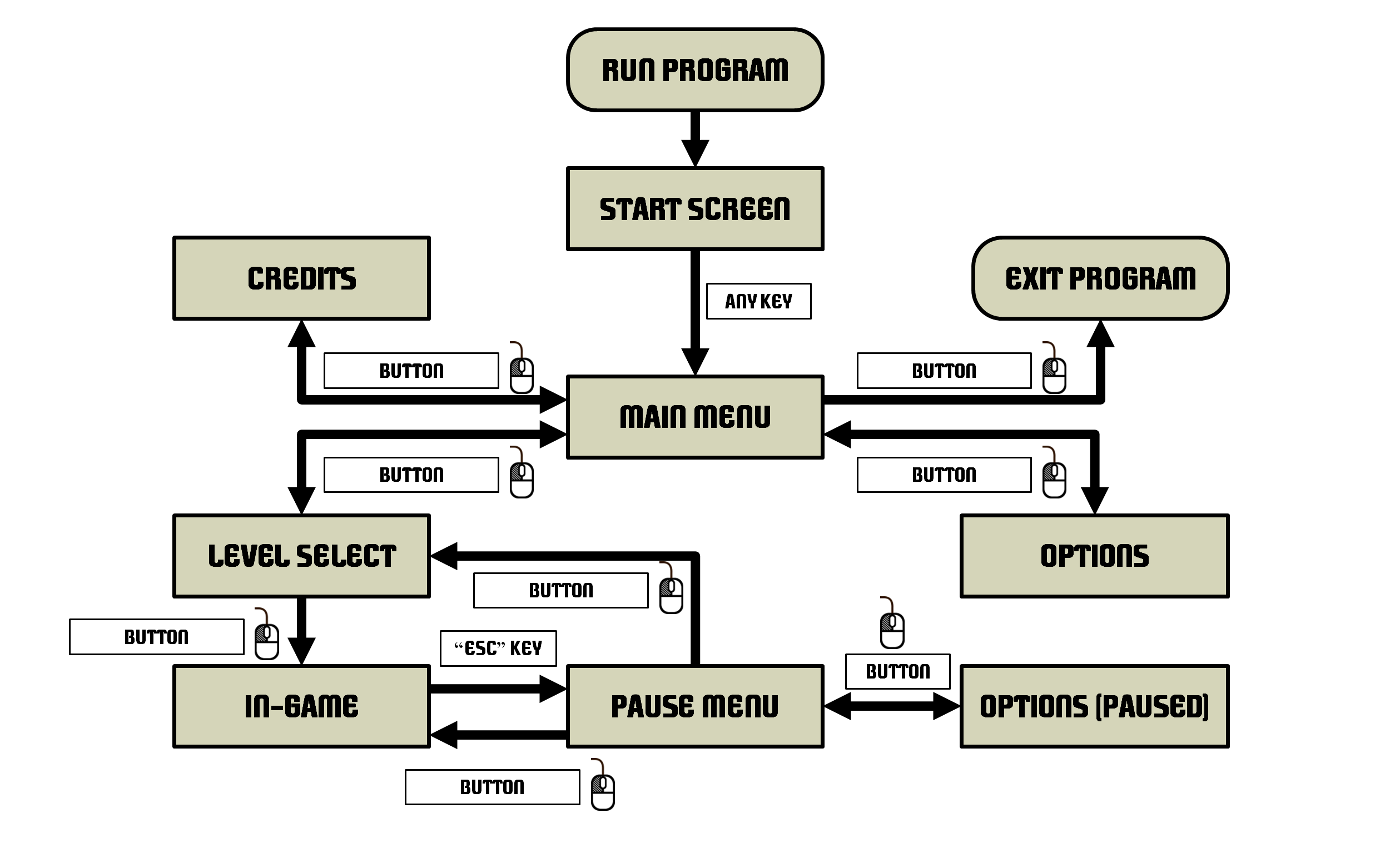


Figure 4.0: Storyboard of the Game (Made on *Microsoft PowerPoint*)

## Starting Screen

* + Once it is run, the game will not require any inputs until the Starting Screen appears.
  + It will serve as a quick one-picture introduction to the game (see Figure 4.1).
  + Clicking/pressing a key will proceed to the Main Menu.



Figure 4.1: Starting Screen (Made on *Adobe After Effects*)

## Main Menu

* + The menu will have buttons to each of these screens: Level Selection Screen (play button), credits, quit, options (see Figure 4.2).
  + 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.



Figure 4.2: Main Menu Screen (Made on *Microsoft PowerPoint*)

## Options Menu Screen

* + The options provided will be single-left-click buttons: toggle screen size (if implemented), erase save file, music on/off (if implemented), sounds on/off (if implemented), 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 (see Figure 4.5).

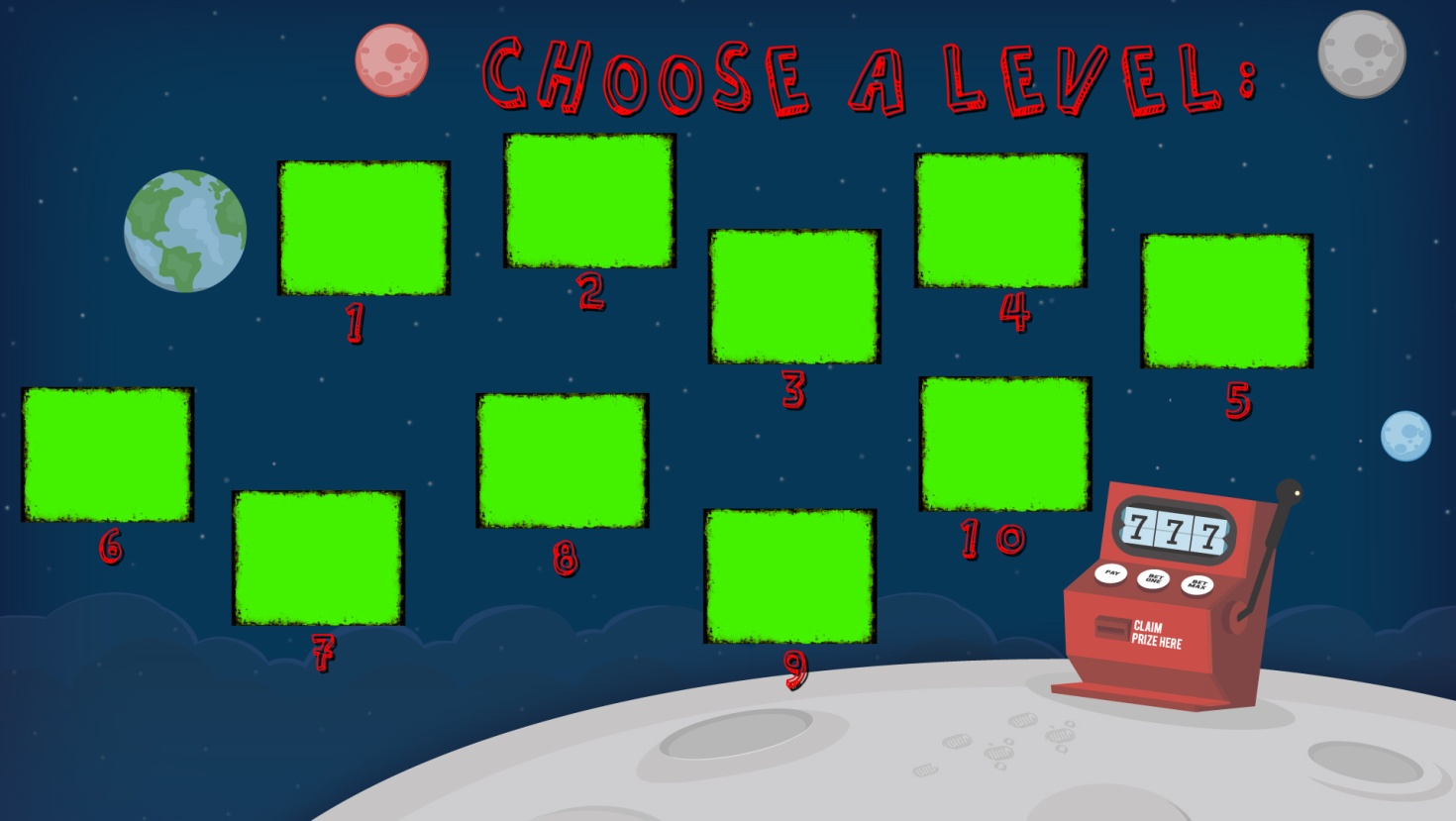


Figure 4.5: Level Selection Screen (Made on *Adobe After Effects*)

## 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 shown on the upper left corner of the screen (see Figure 4.6).
  + Mission objective locations will be highlighted by beacons/crosshairs. If the marker is not on the screen, an arrow on the screen will point to it.
  + Stats, like the fuel remaining will be displayed on the upper right corner (see Figure 4.6).
  + The upper center of the screen will have a toolbar for various single-left-click buttons (see Figure 4.6).
  + The window will be a viewpoint centered on the rocket on all times, like a camera filming it from afar, while the entire level does not have to fit on the screen.
  + The viewpoint will have 2 or 3 zoom settings, each centered on the rocket. The [+] and [-] keys, or scrolling the mouse wheel up or down would zoom in or out, respectively.
  + The button to toggle the overlay to display forces (right), as well as the one to calculate and display the rocket’s trajectory (center) will be on the top center part of the screen (see Figure 4.6).
  + The player is able to view again the information windows at the start of a level by clicking on the “?” button on the upper center (left) part of the screen (see Figure 4.6).
  + Trajectory estimations are shown via patched conics (like the example on Figure 4.6).
  + Pressing “Esc” will open the pause menu, which offers single-left-click buttons for resuming the game, displaying an options menu, returning to the Level Selection Screen, or restarting the level.
  + The options accessed via single-left-click buttons would be: toggle screen size (if implemented), music on/off (if implemented), sounds on/off (if implemented), back to pause menu.
  + Additional text will appear as pop-up windows with single-left-click buttons on them to close them.



Figure 4.6: In-Game Screen (Made on *Microsoft PowerPoint*)

## 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 any key (see Figure 4.7).
  + Any dialogue will appear in speech bubbles.
  + The cut scenes may proceed as comic panels or basic animations.

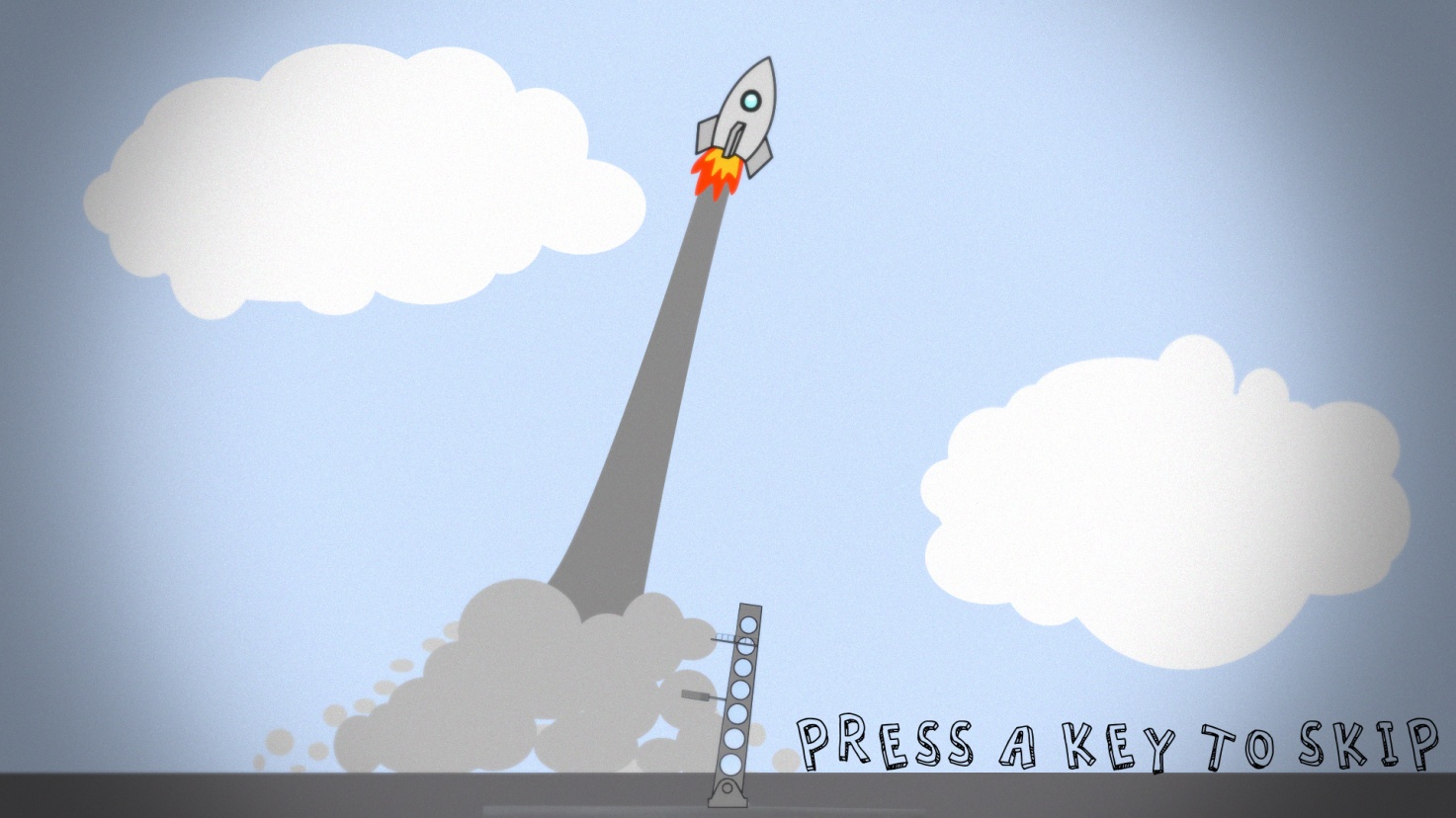


Figure 4.7: Sample Cut Scene (Made on *Adobe After Effects*)

## Pre-Launch Window (if implemented)

* + Pressing “Esc” will open the pause menu, which offers single-left-click buttons for resuming the game, displaying an options menu or returning to the Level Selection Screen.
  + The options accessed via single-left-click buttons would be: toggle screen size (if implemented), music on/off (if implemented), sounds on/off (if implemented), back to pause menu.
  + 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 life support resources to pack, or fuel, or a choice of engine (see Figure 4.8).
  + Other horizontal sliders arranged in a vertical list will be on the lower left side of the screen, with which the player adjusts values for launch thrust and angle (see Figure 4.8).
  + A large single-left-click launch button, titled “Ready”, in the lower right part of the screen, will start a launch cut scene that (if the launch is successful) links straight to the In-Game Screen. Otherwise, an explanation will be displayed about the failed launch and the player will try to configure the rocket again (see Figure 4.8).
  + A picture of the rocket equipped with the current configuration will be displayed on the right part of the screen (see Figure 4.8, currently labeled “<IMAGE>”).

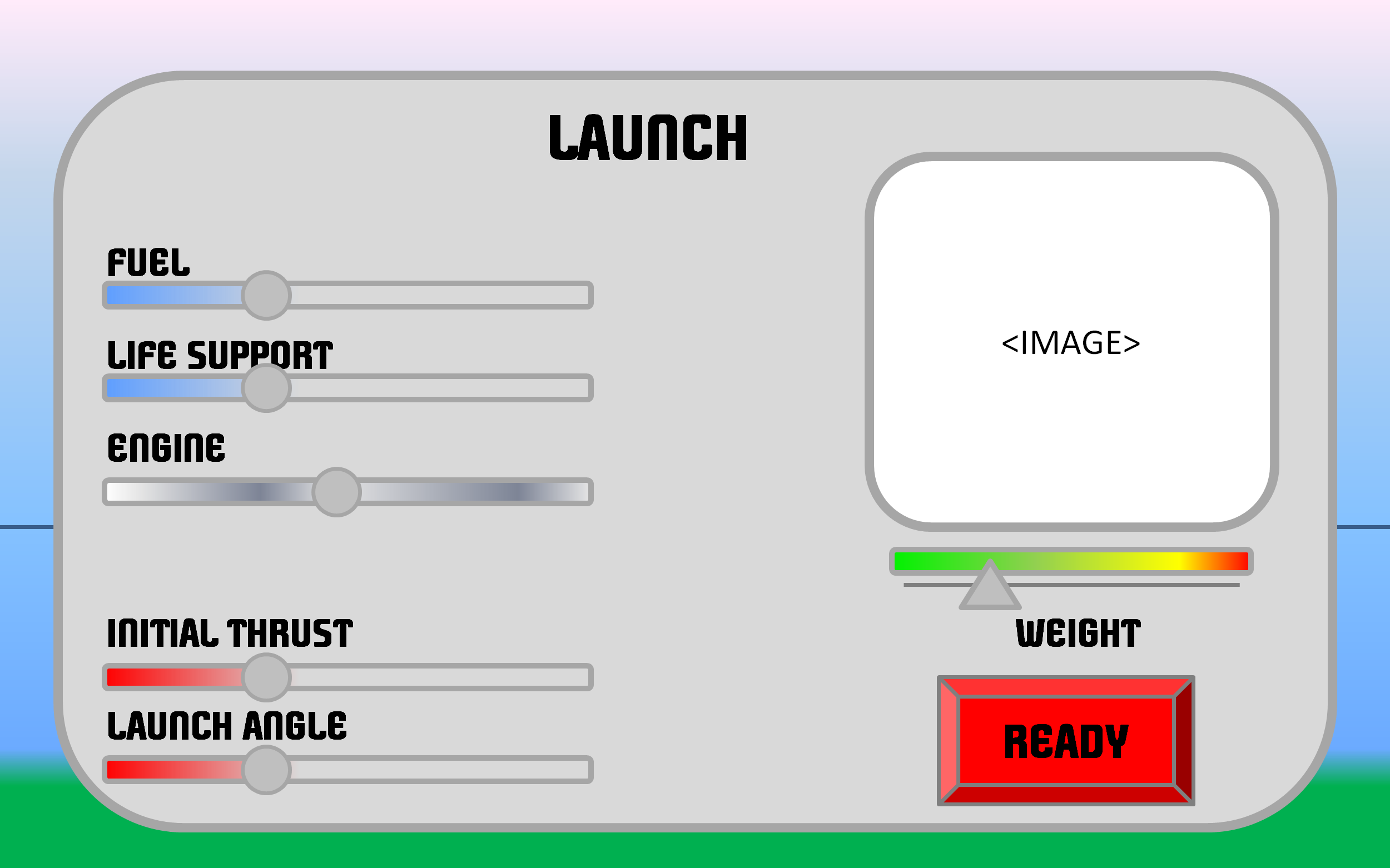


Figure 4.8: Pre-Launch Window (Possible Concept, Made on *Microsoft PowerPoint*)

# 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 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. McKelvey, C. (2015, June 18). Why this indie video game sensation caught NASA's attention. Retrieved March 12, 2016, from http://theweek.com/articles/560370/why-thisindie-video-game-sensation-caught-nasas-attention
2. Wikipedia. (2016, February 27). *Kerbal Space Program*. Retrieved February 27, 2016, from https://en.wikipedia.org/wiki/Kerbal\_Space\_Program
3. Barbarian Games. (2012, October 04). Into Space 2. Retrieved February 28, 2016, from http://www.kongregate.com/games/BarbarianGames/into-space-2
4. Wikipedia. (2016, January 23). *Angry Birds Space*. Retrieved February 27, 2016, from https://en.wikipedia.org/wiki/Angry\_Birds\_Space
5. Harbour, J. S. (2012). *Beginning Java SE 6 Game Programming* (Third Edition ed.). Boston, MA: Course Technology.
6. 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
7. 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
8. Giancoli, D. C. (2009). *Media manager Physics for scientists & engineers with modern physics, 4th ed. Giancoli.* Upper Saddle River, NJ: Pearson Education.