

Design Document for:

**SPACE RACE**

**A game about racing… IN SPACE!**

All work Copyright ©2013 by Clam Slam productions

Version #1.00

Tuesday, April 23, 2013

Table of Contents

* Name
* Design
	+ Design History
	+ Design Team
		- Clam Slammers
		- The Team
* Game Overview
	+ Philosophy
* Features
* Game World
	+ Overview
	+ SPAAAACE
	+ Physical description of the setting
		- Locations
		- Travel
		- Scale
		- Passage of Time
	+ Camera
		- Overview
		- Cockpit Camera
		- Minimap Camera
		- Hanger Camera
		- Menu Camera
	+ Game Engine
		- Overview
		- Collision Detection
		- Other physics
	+ Lighting Models
		- Overview
* Levels
	+ Tutorial
		- Overview
		- Story
		- Objective
	+ Fleet
		- Overview
		- Story
		- Objective
	+ Graveyard
		- Overview
		- Story
		- Objective
	+ Asteroid Field
		- Overview
		- Story
		- Objective
* Game Characters
	+ Overview
	+ Player
	+ Enemy Racers
		- Enemy #1
		- Enemy #2
		- Enemy #3
	+ NPC ships
* Music/Sound
	+ Overview
	+ Musical scores
	+ Sound Effects
* 1v1 Race Game Mode
	+ Overview/Objective
	+ Open Level Design
	+ Waypoint System
	+ Victory Conditions
	+ Shields/Health
		- Overview
		- Player Death
	+ Enemies
		- Overview
		- AI Waypoints

**Design**

**Design History**

 This section is meant to keep track of any major revisions to the design document. Versions of the design document are ordered sequentially starting with Version #1.00 as the first completed draft. Each update after Version #1.00 includes a list of changes made since the previous version.

**Design Team**

**Clam Slammers**

 The Clam Slammers (also known as Clam Slam Productions) are a silly group of individuals put together for the singular purpose of utilizing their unique skillsets in order to make this game absolutely awesome.

**Team Members**

* Evan deLaubenfels
	+ Designed the waypoint system, two of the levels, and acquired most of the art assets.
* Troy Sullivan
	+ Troy is a senior in the Computer Science & Engineering department at the Ohio State University. He worked on various aspects of the project including ship physics, level design, and menu design.
* Clayton Mallory
	+ Clayton did some initial work on the AI, but focused on other aspects of the game when Justin began work on the intelligence. Following the AI work, he assisted with the menu design and interface, some of the scripting, and level design. Additionally, he created the Technical Poster for the presentation.
* Doug Pittenger
	+ David (Doug) Pittenger is a senior at OSU who will be graduating in December. He was in charge of creating the sounds and music for the game.
* Justin Smith-Eble
	+ Justin Smith-Eble is a graduating senior in the Computer Science & Engineering department at the Ohio State University. He was primarily responsible for project documentation and team leadership. Additionally, Justin was responsible for the implementation of the game’s artificial intelligence

**Game Overview**

**Philosophy**

**Why?**

 While this game was not meant to be a life-changing experience, but rather an expression of our group’s creativity and skills as Computer Science & Engineering students at the Ohio State University. As a capstone project, this game is meant to represent the completeness of our degrees and our preparedness to apply this information in the professional world.

**Features**

* **General Features**
	+ Massive open space environments
	+ Three unique enemies for the player to defeat
	+ Visually stunning environments
	+ 3D interactive scene for selecting levels
	+ Tutorial level
* **Gameplay**
	+ Realistic space physics/controls
	+ Challenging AI opponents
	+ Movement and rotation on all three axes

**Game World**

**Overview**

 It is the year 3465.

A new planet, Eden, has been discovered, containing enough clean, crystalized fuel to sustain a planet for millennia.

Current space technologies make travel to Eden infeasible. So the galactic federation Galaxy United Indivisibly started the Space Race to spur scientific advances.

The planet represented by the winner receives a massive science grant from GUI and also doesn't die in a fiery space wreck.

This year Commander Hawk hopes to lead his home world, New Earth, to victory after years of failure to the inhabitants of Angra Mainyu.

However in this race, Doctor Nefarious, Commander Hawk’s arch enemy, is determined to control Eden in order to progress his mad science and take over the galaxy. He contracted Nova Jackal and Chef Meister to make sure he gets to Eden first.

**SPAAAACE**

 As the name “Space Race” probably implies, this game takes place almost entirely in the open void of space. However, since most of space is actually quite dull, each race will take place on a course representing important locations and events in this otherwise empty void of space.

**Physical Description of the Settings**

**Locations**

 Each race takes place in a specific location throughout the Milky Way galaxy. In general, the race location is chosen by the player’s opponent for that race and represents a location special or important to that enemy racer. Each stage will be covered in more detail later on in the design document.

**Travel**

 After each race, the survivors have their ships loaded into the hangar of the great GUI carrier U.S.S Not-Enterprise. Although there are many warp-capable ships throughout the GUI, this method of transport is mandated to prevent outside interference or fleeing contestants.

**Scale**

 As previously mentioned, the great Space Race takes place throughout the entirety of the GUI. Although the scale of the individual races may vary, most take place in an area comparable to the far orbit of a New Earth-sized planet.

**Passage of Time**

 With the advent of a new generation in warp technology, the U.S.S. Not-Enterprise is now able to transport candidates nearly anywhere in the galaxy within a week. This means that the entirety of the great Space Race can now usually fit within a month’s time, barring unforeseen circumstances.

**Camera**

**Overview**

The camera is very typical for a first-person game. It is simply attached to the front of the player’s ship and made a child of the Player object transform in Unity. This forces it to stay oriented with the player at all times.

**Cockpit Camera**

The cockpit model is attached to the camera in the same way the camera is attached to the player. This makes it appear that you are looking out the window of a spaceship cockpit.

**Minimap Camera**

The minimap camera is a scene camera that remains a fixed height above the camera at all times, pointing  downward at an angle. In addition to seeing the level geometry, this camera can also see objects that are culled on other cameras, allowing the minimap grid and player/enemy/waypoint indicators to be attached to the scene, but only be visible on the minimap.

**Hangar Camera**

The Hangar camera utilizes a package called iTween that allows for smooth path following from one area to the next. There are only four places the camera can stop at, so using a simple state diagram, you can determine which path the camera needs to follow to get to the next area.

**Menu Camera**

The menu camera is the main scene camera, but with GUI elements drawn on top of it. The main menu is like any other scene, but with a pre-set path, thus the menu camera is the same as the ingame camera.

**Game Engine**

**Overview**

We used the Unity game engine because it allowed us to avoid reinventing the wheel and work more closely with the game assets.

**Collision Detection**

 The collision detection is handled by the Unity engine.

**Other Physics**

 The player ship uses a script for its physics in order to simulate anti-gravity. The effect, however, could have been produced by Unity’s built-in Rigidbody component with a script that applies forces to it.

**Lighting**

**Overview**

 Most scenes contain ambient lighting, and therefore don’t have a lighting model. The only scene that doesn’t use ambient lighting is the Graveyard scene.

**Levels**

**Tutorial Level**

**Overview**

 The tutorial level is a quick level designed to introduce the player to the controls of the game.

**Story**

 Commander Hawk has decided to train for the upcoming Space Race event using one of his ship’s augmented reality missions.

**Unique Elements**

 The Tutorial level makes heavy use of popup elements in order to instruct the player in the basics of the game such as how to control the ship.

**Fleet Level**

**Overview**

 The player must race against Admiral Bōntapίk through a live-fire training exercise being put on by the admiral’s massive fleet.

**Story**

 Admiral Bōntapίk has chosen his fleet as a racetrack. Angra Mainyu has one of the largest military forces in the galaxy and from here his ships will be able to provide supporting fire.

**Unique Elements**

 The player must weave through numerous ships engaged in deadly combat, all while avoiding the Admiral’s fighters and stray asteroids from the local system.

**Graveyard Level**

**Overview**

 In the second leg of the Space Race, Commander Hawk must navigate through a mysterious and dangerous graveyard.

**Story**

 Nova Jackal has chosen his planet’s graveyard asteroid as a racetrack. This is the place where all his people’s dead are buried, and he believes that his ancestors will help him to win.

**Unique Elements**

 The course for this race runs straight through the narrow heart of a sacred asteroid, with danger at every bend.

**Asteroids Level**

**Overview**

 In the final stage of the great Space Race, the player will have to navigate through one of the galaxy’s most dangerous asteroid fields.

**Story**

 Chef Meister has chosen the asteroid field near his home planet as a racetrack. He chose this place because he wouldn’t feel like a man racing anywhere less dangerous.

**Unique Elements**

 This level featured a large number of moving asteroids which can cause serious problems for the player if he/she does not think carefully about his/her course.

**Game Characters**

**Overview**

Each planet sends a representative racer to participate in the Space Race. This representative must be both the most skilled racer on their planet and the one with the highest quality ship.

**Racers**

**Commander Hawk**

 A resident of New Earth, Commander Hawk is a bounty hunter and longtime enemy of both Admiral Bōntapίk and Doctor Nefarious. He mainly takes on bounty hunting jobs so he can pay for his ship’s upkeep.

**Admiral Bōntapίk**

 The leader of Angra Mainyu, he possesses a disdain for all humanity and he has been the winner of the last few Space Races.

**Nova Jackal**

 Nova Jackal is a mercenary hired by Doctor Nefarious to kill Commander Hawk. He has been outsmarted several times by Commander Hawk before and is determined to succeed this time.

**Chef Meister**

 A genetically engineered supersoldier, his model is superior to other supersoldiers because he can prepare his own food. He has been disqualified from several previous Space Races for bringing his signature spatula gun.

**Music/Sound**

**Overview**

 The sound clips for the most part were made with Audacity sound editing software. Sounds were either borrowed from the internet or recorded and then imported into Audacity for editing. Once the sound was finalized and game ready, they sound was then exported in MP3 format to the assets for use.

**Musical Scores**

 The musical scores were scavenged from the internet using free scores that were available.

**Other Sound Effects**

 Some other significant sound effects were also made by combining multiple sounds into one through layering and volume control. These sounds were both recorded and borrowed in order to get the correct sound produced.

**1v1 Race Game Mode**

**Objective**

 In each race the player must navigate obstacles, live enemy fire, and his/her opponent in order to reach designated waypoints and be the first racer to cross the finish line. Each course is unique, but each one requires that the racer pass through each waypoint, seen as a white ball, in order before he/she can complete the race.

**Open Level Design**

 Simply put, space is very, very big and very, very open. A large challenge of this game was our need to simultaneously represent space somewhat realistically while still keeping the sense that the player is engaged in a fast-paced, high-stakes race. The waypoint system helps to solve this issue by forcing the player to make constant turns and course corrections instead of taking a single straight line to their destination. In addition, most levels contain a significant number of obstacles directly in the path of the player. These obstacles also provide incentive for the player to swerve and adjust their course in order to avoid collisions.

**Waypoints**

 Having full roam of a 3D environment is a difficult situation for racing. It is difficult to define a path. To solve this, we implemented waypoints throughout each level that the racers are required to fly through in a particular order. This is a loose way of defining a path that leaves some room for racers to choose their own path through each level. The racer who reaches the last waypoint before the others is the winner.

**Victory Conditions**

 If the player is able to successfully navigate the course faster than the opponent, he/she is declared the winner of that race. Crossing the finish line is as simple as making contact with the final waypoint in the course. At this point, the racer will receive a menu screen asking whether they want to go back to the hangar or quit the game.

**Shields/Health**

**Overview**

 The player has limited defensive measures on their ship, including rechargeable shields protecting the ship’s relatively fragile hull. Shields will recharge after a few seconds of not taking damage and completely protect the hull while active. Hull damage is permanent throughout the level and reducing the hull integrity to zero triggers player’s death.

**Player Death**

 Once the player’s hull meter is empty, a death event will trigger. This event involves the camera zooming out and showing an explosion where the player was. The player will then be prompted with a menu asking whether they want to retry the level or to move on to a different one.

**Enemies**

**Overview**

 Each enemy racer is controlled by a force-based artificial intelligence which guides them through the course. In short, the racers are drawn towards waypoints while simultaneously being repelled from objects such as ships and asteroids. This allows the use of dynamic and semi-random elements in levels, which a set-path AI may not be able to handle.

**AI Waypoints**

 The AI uses a waypoint system similar to the player, except these waypoints are not visible to the player. Additionally, the AI has more waypoints than the player does. Having these additional waypoints allows the AI to better maneuver around static obstacles and to generally make their path seem less linear.