

Introduction:

This is a post mortem for *Escape from Planet Doom*, a game created by Team SuPeRLiCiOuS during the course "Game Programming and Design" at Rutgers University New Brunswick, NJ USA. The primary goal of the course was to spend the beginning of the semester rapidly prototyping as individuals, then forming into groups to develop the strongest and/or most popular prototypes into full fledged games.

The prototype chosen for development which was the basis for Planet Doom involved two views. One was a very typical example of a simple horizontal scrolling shooter, while the second showed what was going on inside the ship focusing specifically on the emotional states of the characters and also how damage done to the player in the external shooter stage of the game impacted (often through death) the characters on the inside of the ship.

The initial concept of the game as shown by the prototype was to stress the connection between an avatar (ship, boat, etc...), whose existence the player typically abstracts away as simply being a thing which shoots and avoids bullets, and what actually happens inside of that ship. Specific focus was put on the emotional state of characters within the ship and the final game became a manic shooter external game with a narrative driven internal game where the player's choices of which character to put where would influence the story as well as the balance of abilities in the external

game. The internal game consisted of three rooms representing weapons, shields and navigation on the ship and each of the three story characters in the game could be moved to any of the three stations to upgrade that station which would also impact the story line of that character for that level.

Both the prototype and the final game were based on the Flixel game engine developed by Adam 'Atomic' Saltsman which was designed for use with 2D sprite-sheet style graphics and animations.

What Went Right:

Immediate Infrastructure

One of the most important items which we took care of the very first day of the first week of production is something which if it exists, will never be mentioned, but if it doesn't undermines the entire process. In establishing the coding infrastructure by setting up our SVN server, providing IDE setup instructions and setting up Dropbox for binary files we made our code integration and sharing process completely seamless. By establishing this as a first priority we made sure that everyone was able to work and contribute from the very start without any technological hiccups getting in our way.

Segmentation of Roles

In managing everyone's contribution to the project, the most important thing we took care of early on was to identify and segment each team member's abilities and time allowances. By making sure that every team member's focus area played to their strength and that the time availability of each team member matched the requirements of their assigned role we were able to balance the work load in a way which was as smooth as possible given the diversity skills and size of the team. Dedication in terms of time commitment was especially important to accommodate as it allowed the project to progress by fully utilizing every team member's resources to their fullest.

Communication

In terms of management, design, implementation and almost every aspect of the game one of the most critical components was creative interplay and commentary between team members. By meeting twice a week (three including seminar times), and almost always in person, we created an atmosphere which was focused on the task at hand and also allowed for much more discussion and commentary than digital communication would have given us. The dynamics of in-person interplay between the team members was crucial to the development of many key features of the game.

Flexibility and Fearlessness in Cutting

In multiple points during the development of the game, it was absolutely necessary to cut ideas, features, art styles, content and sometimes even fully developed code. Without the ability to do so fearlessly and to be flexible in changing what we realized needed to be changed this project would never have been completed. Any stalling or hesitation in removing what should have been removed would have led to a product that simply could not live up to any reasonable set of standards.

(See: Appendix A)

Choice of Platform

Finally, although also often overlooked due to the “obviousness” of the decision in many cases, the choice of platform formed the basis for the eventual success of the project. Choosing to use the Flixel engine as our platform allowed for a level of rapid development that would have been nearly impossible to achieve with any other system. Despite the limitations of Flixel and of using Flash there were many instances where the flexibility of the system gave us the ability to safely rethink many portions of the game without worrying nearly as much about implementation time. Familiarity with the tool was also important since three of our four group members already had some familiarity with Flixel in the prototyping stage. And last but not least, the web-based deployment of the game allowed for a level of widespread testing which would not have been achievable had we gone with a more desktop or console oriented platform.



What Went Wrong:

Vague Prototype

Due to the structure of the course, the first version of every project made was a prototype game often completed in a week or two. The base prototype for the development of Escape from Planet Doom was essentially more of demo rather than the basis for a full fledged game. The idea being expressed in the prototype was not one of a completed game, but instead of a manifest idea. Specifically the idea of an emotional connection between the player's avatar (space ship) and what actually

happened inside of that avatar. Due to the initial vagueness inherent in this idea, and the prototype being designed without a larger game already in mind many of the other problems mentioned in this section arose.

Scope Creep and Content Creation

In trying to focus specifically on an emotional connection to the player, we ran into issues of scope and feature creep as well as a content creation nightmare in trying to craft the narrative basis for the player's involvement with the story. In trying to create a narrative, many ideas which were essentially gimmicky were posed which caused the initial feature list to bloat with ideas without a clear focus. Even after narrowing down to a single focused idea, we still had on our hands the problem of creating enough content, both narrative and visual, which would properly portray the idea we were trying to convey.

Lack of Unifying Vision

Once again, stemming from the vague prototype, one key problem was a lack of a single unifying vision from the beginning of the project. Many of the issues in the design of the game could have been avoided by planning out the entire game down to the finer details from the very start. Although this may not have been entirely possible for this particular project, there were still many cases where a single vision was either missing or left open to interpretation causing scatter in the development process.

Too Reactive in Design

Combining many of the previously mentioned problems a new problem arose in that we became too reactive in designing the game. The problem of feature creep and lack of vision led to a lot of trial and error, which was often simply in reaction to comments made by testers. Single comments would often lead to entire developed features, which although often successful, were difficult to keep up with when this process happened over a period of a few weeks. A unifying vision would have likely fixed this issue, but nonetheless it was important for us often to take a step back and look at what we were doing because it was what we intended and what we were doing simply in response to comments made by players.

Lack of Easy to Produce and Well-Established Art Style

As a subset of the previously mentioned problem of lack of vision, there was a significant lack of vision in the design of the art style. Instead of sitting down and developing a single style all together which would be set and provide guidelines for all future art, much of the art was often iterated off of itself without care to notice

inconsistencies between art pieces. Until we settled with the final pixel style, the iteration and production of the art proved too time-consuming to be feasible.

(See: Appendix A)

Conclusion:

In conclusion, the final product produced was one that we were very happy with given the circumstances. Although there are still many aspects which could be improved, the game that we developed is one which had positive reactions from many testers and has been submitted to multiple flash portals for the public at large to view. Comments from those portals have also been positive except for areas where we already knew needed improvement.

The development process itself, although often difficult, was exceptionally fun and the experience we gained especially from seeing brand new players testing something we created was invaluable.

Specs:

Name: Escape from Planet Doom

Developer: Team SuPeRLiCiOuS, Rutgers University, NJ, USA

Team Members: Yori Kvitchko, Dan Gopstein, Tarek El-Gaaly, Ming Jin

Number of Developers: 4 Students

Release Date: 4/27/10

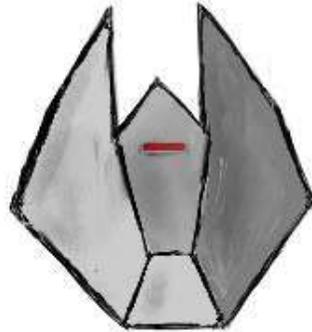
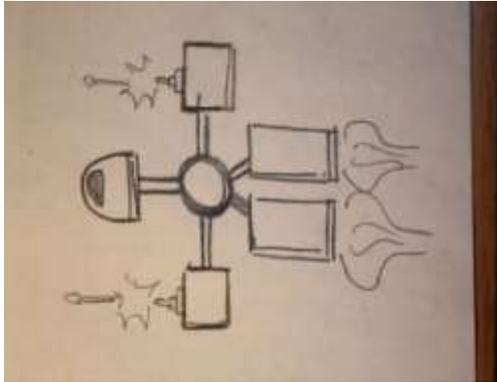
Platform: Adobe Flash Browser Plugin

Used Software: FlexBuilder, Photoshop, GIMP, Audacity, Transistor Synth 404

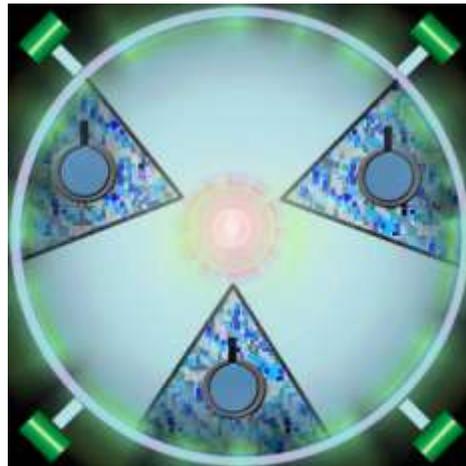
Used Hardware: Four, occasionally very old and clunky, laptops

Appendix A: Evolution of Art Style

Initial Sketches:



First Concept:



Ship Iterations:



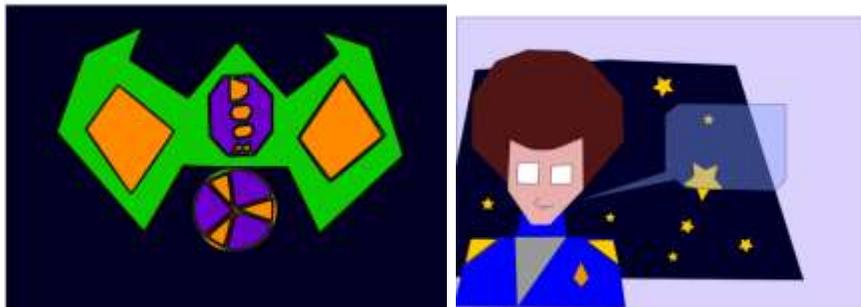
Character Iterations:



Pixel Style Testing:



Vector Style Testing:



Finalized Style:

