PROJECT: ROGUE SKIES

Spring 2018 CSCI 470 Capstone Project Proposal

By: Andrew Clink

Advisor: Scott Kast

What is the Project?

Project: Rogue Skies is a project designed to provide practical Software Engineering experience, develop better Programming skills, and foster Computer Science knowledge. To achieve that, this Project will be to plan, detail, document, and develop a Rogue-like computer game program. This rogue-like will attempt to translate the precarious and dangerous Beyond Visual Range (BVR) and Within Visual Range (WVR) combat areas of Fighter Pilots in 3rd-Generation fighter jet aircraft. Project: Rogue Skies will seek to emulate the design process, decisions, and development of a computer game, starting with idea inception and ending at post-release support.

* Plan
  + Timeline Development
  + AGILE development process creation and modification
  + Software Lifetime Planning
  + Support Software and Documentation (Trello, OneNote, BitBucket)
* Detail
  + Feature development
  + Mechanical Sketching
  + Plain English explanations of workflow
* Document
  + Development Logs
  + AGILE process meetings
  + Tutorial writing and development
  + Encyclopedia entry writing
* Develop
  + Program coding
  + Program testing
  + Library Editing
  + Prototyping
  + Alpha/Beat/Gamma Testing and Implementation
  + Release and Post-Release

Who will be working on the Project, and who is the customer?

I (Andrew Clink) will serve as the project manager and lead developer on Rogue Skies. Other students, friends, family, co-workers, etc. may also be involved with the Testing phases. Outside help will be kept strictly to a minimum, involving brainstorming and bug-hunting.

The aim of this game is to appeal to rogue-like veterans, as the premise of the game hasn’t been done before in a rogue-like setting. Casual and Beginner players will also be coaxed into the game with several, quality-of-life features that many (even modern) rogue-likes lack, such as full documentation in-game, clarified tooltips, and tutorials.

When will the project be completed?

Project completion is a tricky matter when it comes to video games, as Post-Release updates are now the standard for nearly every single program in some form or fashion. Unfortunately, while there are many who would enjoy working on a game for extended periods of time, the Semester only has four months. For this reason, project ‘completion’ will be at the Beta 1 Phase of development—the first major update to the game that is targeting only bugs, unintended issues, and minor features and content that don’t disrupt the previously laid out features.

Where will the Project be shown when completed?

The project will be displayed in its most basic form during the Midterm Presentation. After Final Presentation, the Project will continue to be available from the school’s website, along with Documentation and Presentation Recordings for those interested in the process as a whole.

How does the Project define Progress?

This Project is being developed in Phases. These Phases each represent a significant idea or development step, starting from an idea on a napkin to a final install package. These Phases are as follows, in chronological order with an abstract idea of how much time it would take to complete:

* Research Phase
  + What does the Final Product look like?
  + How does it run/play?
  + What are the fun parts?
* Prototyping
  + What is the Primary Game Loop?
  + What is the Secondary Game Loop?
  + What is the Minimally Viable Product?
* Alpha
  + What Features need to be added to the Prototype to be considered a Release-worthy Game?
* Beta
  + What are the unintended events that need to be addressed before moving to Release?
* Gamma
  + What can be done to smooth the game experience over?
  + How can we best implement the In-Game documentation and tutorials?
* Release
  + Get feedback from players; what do they want to see changed/added?
* Post Release
  + What were features originally cut that can be added back into the game?

Why is the project being done in this way?

To be a good Software Developer, a person needs three things: Programming skills, Software Engineering experience, and Computer Science knowledge. Project: Rogue Skies is an attempt to emulate these experiences in a scholastic environment, while simultaneously fostering and improving these personal aspects. These three facets of development are incredibly important to the development cycle of software:

* Software Engineering experience to ensure the process is done in an orderly fashion, able to react to quickly shifting or changing demands in a volatile marketplace
* Programming skills to actually write the code to have the intended effect of what was planned, not simply hack it together until it compiles
* Computer Science knowledge to make sure the code works quickly and safely, without exposing the environment the program runs in to malicious users or resource