spaceship rpg gAME

Text Based Game

CIS\_350 Winter 2024

Professor: Dr Nandigam

Team members

Adrian F Perez-Lopez

Eric Moras

Phuc Le

Contents

[Project Information 3](#_Toc158608804)

[Features 3](#_Toc158608805)

[Screenshots 3](#_Toc158608806)

[Project Plan 4](#_Toc158608807)

[Requirements & Definition 4](#_Toc158608808)

[Development 4](#_Toc158608809)

[Verification 4](#_Toc158608810)

[Maintenance 4](#_Toc158608811)

[Umbrella Activities 4](#_Toc158608812)

[Responsibilities 4](#_Toc158608813)

[Requirements & Specification 5](#_Toc158608814)

[Use-Cases 5](#_Toc158608815)

[Natural Language Requirements 5](#_Toc158608816)

[Traceability Matrix 5](#_Toc158608817)

[Design 6](#_Toc158608818)

[Development 7](#_Toc158608819)

[Code Standards 7](#_Toc158608820)

[Static Analysis 7](#_Toc158608821)

[Code Documentation 7](#_Toc158608822)

[Configuration Management 7](#_Toc158608823)

[Verification 8](#_Toc158608824)

[Integration Tests 8](#_Toc158608825)

[Unit Tests 8](#_Toc158608826)

[Code Coverage 8](#_Toc158608827)

[Requirements Coverage 8](#_Toc158608828)

[Postmortem 9](#_Toc158608829)

[Earned Value 9](#_Toc158608830)

[Variances 9](#_Toc158608831)

[Lessons Learned 9](#_Toc158608832)

[References 10](#_Toc158608833)

# Project Information

The Role-Playing Game (RPG) project is a text-based game that will be a single character (space janitor). The janitor(character) faces different challenges while cleaning shipwrecks; therefore, he acquires different fighting techniques to help him move up in rank, such as cadet and captain, and ultimately becomes a king.

As the character moves through different maps, enemies will get stronger; thus, the player will be rewarded at the end of each challenge. These rewards will help him increase his stats by acquiring various combat tools.

## Features

* Username Input.
  + The user interface will get player’s name; otherwise, Bob will be assigned if user does not want to provide a name.
* Stats.
  + This feature will display how the points are going to be distributed while playing; in addition, the player will be awarded 5 free points to get the game going.
* Game Screen.
  + A map will be displayed showing the player current position.
  + Stat information to keep track of the game.
  + Message screen showing the interaction of the player.

## Screenshots

* A screenshot of a computer

  Description automatically generatedName input.
* **A screenshot of a computer game

  Description automatically generated**Stats display.
* A screenshot of a computer

  Description automatically generatedGame Screen.

# Project Plan

[Overview of Project Plan, including any customizations to the software process model]

## Requirements & Definition

[Plan for requirements and definition detailed here]

## Development

[Plan for development / implementation detailed here]

## Verification

[Plan for verification detailed here]

## Maintenance

[Plan for maintenance (e.g., problems found by prof or during updates for next release) detailed here]

## Umbrella Activities

[Plan for project management and status tracking / meetings detailed here]

## Responsibilities

[Describe the responsibilities of each member, how you plan to break down the work]

# Requirements & Specification

[Description of methods used (e.g., use- case diagrams, user stories, use case-descriptions)]

## Use-Cases

A diagram of a person's diagram

Description automatically generated

[Include use-case diagrams AND use-case descriptions here]

## Natural Language Requirements

[Include requirements here]

## Traceability Matrix

[Include traceability (matrix) from requirements to use cases]

# Design

[Description of methods used (e.g., class diagrams)]

[Design descriptions (e.g., descriptions of each class)]

# Development

[Description of methods used (e.g., Checkstyle, PMD, Javadoc) AND any additional libraries that you used]

## Code Standards

[Include report on code standards and justification for any variances]

## Static Analysis

[Include report on static analysis and justification for any variances]

## Code Documentation

[Include link to javadocs (likely included as separate file) and justification for any non-documented areas]

## Configuration Management

[Include link to Git repository and Git log]

[Explain / describe method for tracking releases]

# Verification

[Description of methods used (e.g., integration & systems and/or unit testing)]

## Integration Tests

[Include manual and integration test procedures]

## Unit Tests

[Include references to unit tests in code]

## Code Coverage

[Include code coverage reports, must include: coverage of automated tests, coverage of manual tests, and combined coverage]

## Requirements Coverage

[Include traceability (matrix) from requirements to test procedures]

# Postmortem

[Include a reflection on how well the project has gone thus far]

## Earned Value

[Include the earned value calculations for your current status and any explanation of over/under runs]

## Variances

[Include any additional variance (time, coverage, functionality, …) explanations necessary]

## Lessons Learned

[Include lessons learned]

# References

Pokpong145. “Daedalus Class Star-Ship.” *Wallpaper Abyss*, wall.alphacoders.com/big.php?i=1303811.