# Risk Assessment

The Oregon Trail with Python(s) – Red Team

## Introduction

This is a linear path game where you try to make it to the next checkpoint before running out of food. Each decision you make only affects how much time it takes to get to the next checkpoint or your food rations. If you take too long, you’ll run out of food and die. There are also special scenarios where you’ll get more food, lose food, or randomly die. It all is related to the food countdown or randomized death.

There are several risks that can arise when developing this kind of application. This document aims to identify those risks and how to mitigate them. Please see the below table for this analysis.

## Risk Assessment Table

|  |  |  |  |
| --- | --- | --- | --- |
| RISK | DESCRIPTION | RISK PRIORITY | RISK PLANNING |
| Schedule Risk  (The project cannot be completed and tested in the allocated time) | * Coding the game to full functionality will require a great deal that might not be able to be completed during the allotted time. | High Risk | * The game will be simplified to adhere to schedule. * A detailed schedule will be made with set checkpoints to keep development on track. |
| Skills Resource Risk  (The group does not have the proper skills to complete the project) | * Group coding is still a new skill for most members of the group, and we will be working in a short timeframe. | High Risk | * Our group has 2 or more people who have skills in each of the areas of development, allowing for shared responsibility as well as backfill if necessary. * The team will develop and adhere to a schedule to ensure there is time to complete each part. |
| Cost Risk  (The project will exceed the allotted budget) | * Our investor has agreed to fund a fixed budget, so it is important to manage costs effectively to stay within the budget. * Cost benefit analysis is important when considering paid resources that would save time in development. | Moderate Risk | * Free or low-cost options will be used as much as possible. * Thorough cost benefit analysis will be performed on high budget resources. |
| Scope Creep Risk  (Changes will be made to the project scope during development without proper planning) | * The project is a parody of an existing software. The intent is to be similar in style and not a faithful reproduction. * The project needs to be scaled down to a project that can be completed in a fixed timeframe of development. | Moderate Risk | * The flow of the program will be mapped out prior to coding. This will maintain the structure without the temptation to overly rely on the source materials. * The same flow will help in keeping the content to an achievable level. |
| **Operational Risk**  (Work and people processes may be insufficient or degraded) | * The team has to set up and figure out a workflow. * Some team members are unfamiliar with group coding. * Unsure of group availability and final size of group. | Moderate Risk | * Several communication and workflow management tools have already been set up (Trello, Discord) to keep everyone connected and on the same page. GitHub and Google Drive will aid collaboration. * Project planning will further connect team members. * Group members have overlapping skills and can cover for others as necessary. * Cloud backups help build resilience in case of natural disaster or other emergencies. |
| **Performance Risk**  (The product does not meet User Requirements) | * Project may not meet client requirements. * Project may be buggy or otherwise non-functional. | Low Risk | * Should the project be properly completed, risk of not meeting client requirements is low due to having the project approved. * Conversing with clients can ensure requirements are met. * Proper planning can reduce bug risks and ensure requirements are met. * Functionality can be confirmed and improved through rigorous testing throughout the project lifecycle. |
| **Technology Risk**  (The potential for any technological failure) | * Technology used to develop the project becomes outdated or faulty thereby triggering additional problems. * Technology usually always carries a certain amount of risk, like malware. * Technological risks can result in interruptions, which could slow down our project’s operations and processes. | Low Risk | * Team members should be aware of the root causes of the technological risks, the effects of those risks that have been discovered, and the likelihood that the risk will materialize. * Members of the team should keep an eye on each risk factor and be prepared to transition to a different strategy if the technology employed is shown to be ineffective. * Include time to test software and anticipate lags or unexpected delays. * Using up-to-date applications can prevent lags or unexpected delays. |
| **Communication Risk**  **(The group fails to**  **communicate**  effectively, which leads to mistakes and a project that is headed in the wrong path) | * Establishing a communication vehicle. Communication vehicle is a way we convey information to team members. * Availability of team members. | Low Risk | * We must communicate the project plan, timelines, and requirements plainly and clearly from the team leader down to every member of the team, leaving no space for misunderstanding or error. * Everyone working on the project should be able to access the most recent information by ensuring that the same information is consistently given and kept in the same manner across communication tools (Trello, Discord, Google Drive). * Team members should be informed beforehand of the agendas and the meeting’s main objective. * Team members must communicate information, facts, questions, and concerns to all team members in an open and straightforward manner. Each message that needs to be conveyed must be understood completely. |