

**Sandstorm Games**

SubTerra

Request for Proposal  
 Version 1.0

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | When | Who | What |
| 1.0 | 9/18/2025 | Connor Wolfe  Luke  Mather Brisset  Arjun R  Jack  Mikayla  Tyler Gehring | Initial Drafting |
|  |  |  |  |

Table of Contents

1. Problem description
2. Project Objectives
3. Current System(s) – if any or similar systems
4. Intended users and their interaction with the system
5. Known interactions with other systems inside/outside the client organization
6. Known constraints to development
7. Project Schedule
8. How to Submit Proposals
9. Dates
10. Glossary of terms

1. Problem description / opportunity / expression of need

There is a market gap for cave crawler games. Most existing cave crawlers are 3D and have fixed maps with no procedural generation. This makes for stale gameplay, as the replay ability is quite low. The opportunity for this project comes from combining procedural elements such as destructible, randomly generated cave systems. This is an element that has been implemented in games such as Terraria and Minecraft, but they have yet to be implemented in a classic cave-crawler format.

1. Project Objectives

1. For this game, we will have a new randomly generated cave for each level. This gives each player a unique and ever-changing gaming experience. This is very important for our game because it provides the building blocks for how the game functions.
2. The second element we will be implementing for this game is to put a relic in each cave level, which is what you are looking for to get to the next level. This is important because it serves as a trigger for entering the next level and eventually winning the game.
3. The third objective is to implement sound effects and smooth player controls. This is important to our group because it gives the game a polished finish.
4. The forth key objective for this game that we will implement, items like T&T and pick axes that the player can pick up and use to win the game. This element is important because it makes the game more dynamic.
5. The fifth objective is to implement enemies into the game. This is also important to our team because we see it to make the game more challenging, and it adds another dimension to it.
6. And the last critical objective of our game is to make it available on at least 2 platforms. The reason for this is to reach a larger community of customers.
7. Current system(s) – if any / similar systems

As of writing this document Sandstorm games has no existing prototypes or systems built for Subterra. So to provide context for this proposal for development we have listed some games that have systems similar to what is planned to be in the final release of Subterra.

**Spelunky (2008, Mossmouth)** is a cave diving roguelike\* similar to Subterra. Spelunky also features autogenerated caves with destructible terrain. Spelunky also has the player explore its map under a time limit and make use of limited resources to gather as much treasure as the player can before the time runs out. Runs of Spelunky can take about 5-10 minutes.

**Terraria (2011, Re-Logic)** is a 2d survival game. The similarity goes from how Terraria also makes use of auto generated maps with destructible terrain. But, has a focus on long term planning, exploration, and development; which makes it different from what Subterra will be like.

Though Subterra has similarities to these above games, we will take these similar systems in a new direction. With a focus on intelligently using the items you collect to explore the map, fight a variety of different enemies, and collect as many treasures as possible before collecting the relic or your player dies or time runs out.

1. Intended users and their basic interaction with the system

The intended users of SubTerra are:

* Gamers (people who enjoy adventure games)
* Cave exploration enthusiasts
* Puzzle and strategy game fans

Users will interact with the SubTerra system in the following ways:

Exploration:

* + Move through dark, sprawling caves
  + Discover hidden passages
  + Push deeper into the cave system

Resource Management:

* + Gather and manage scavenged resources (like TNT)
  + Make strategic decisions about resource usage
  + Balance risk vs. reward when using limited supplies
  + Use TNT to blast through walls
  + Avoid dangers like collapsing tunnels
  + Uncover secrets and treasures

1. Known interactions with other systems within or outside of the client organization.

The development and delivery of the Unity-based cave crawler game will involve interactions with several internal and external systems to support collaboration, development, and final presentation.

1. **Internal System:**

* Git/GitHub: Used for version control, code management, and team collaboration on the project’s source files.
* Presentation Tools (PowerPoint/Google Slides): Used during the in-class demonstration to explain objectives, gameplay, and technical details.
* Microsoft Word: Used by the team to create and share design documents, RFP, and supporting written materials.

1. **External Systems:**

* Unity Editor: The primary development environment used to design, build, and test the game.
* Unity Asset Store: Provides third-party assets and plug-ins to support game mechanics, environments, and visual design.

3. **Data and Platform Interactions**

* Local Storage: Player data such as progress, preferences, or items will be stored on local devices.
* Target Platforms: The game will be deployed on at least two platforms (e.g., Windows PC and WebGL) for accessibility and classroom demonstration.

1. Known constraints to development

To streamline development, it is important to understand the known and possible constraints that our team will experience during this designing process. Not all constraints listed will cause delays, some of which simply require extra group communication in order to solve.

* Cave Generation
  + The utilization of the ‘marching squares’ algorithm will determine procedural cave generation, which can be difficult for beginners and requires heightened knowledge to successfully implement.
  + Procedural generation needs to be consistent for users and free of major bugs that would impact overall functionality.
* Connectivity requirements
  + All developers must have to a stable internet connection to access GitHub, sync resources, and to share documents which may not always be available to developers.
* Hardware requirements
  + Team members will be required to use Unity which may not function properly on all systems.
* Unity Versioning
  + All developers must use the same Unity version during the entire development process to avoid compatibility issues.
* Asset Licensing
  + Any assets used within this game will either need to be directly created from a member of our group or from third-party assets that fall under the Creative Commons License.
* Function integration
  + One of the mandatory requirements for this project includes each team member adding their own function into the source code.
  + Some developers of this project are not as proficient as other which could lead to delays in the process.

1. Project Schedule

Below is the General project schedule for the SubTerra project. For specific Issue completion dates, refer to issues endpoint of the github directory: https://github.com/tylergehring/SubTerra/issues

**SubTerra Schedule:**

**MVP | Milestone:**

1. Cave Generation by September 21
2. Player, tools, UI components by September 28
3. Completion of MVP by October 11

**Final Product | Milestone:**  DemoDecember 11th

1. How to Submit Proposals

For this project, Sandstorm Games will submit all project proposals, and appropriate documents, into Canvas for comment and approval. Proposals will be submitted by no later than September 19th, 2025, at 11:59 PM PST. Copies of submitted documents can be found on the game development GitHub for SubTerra under the ‘Docs’ tab or by clicking the direct hyperlink provided below.

[SubTerra Project Proposals and Other Relevant Documents](https://github.com/tylergehring/SubTerra/tree/main/Docs)

1. Dates

September 19th: RFP submission

September 20th: Individual Champion documents outlining our tasks and contributions

September 25th: TL1 Presentation over git, SA presentation, and storyboard

September 27th: Class Diagram and Sequence Diagram

September 30th: TL1 Weekly Presentation for SubTerra project

October 7th: TL2 Weekly Presentation for SubTerra Project

October 9th: TL3 Unit Test Engine

October 15th:TL4 video presentation in git

October 18th:TL5 video presentation in git

October 21st: TL4 presentation in class

October 24th: TL3 individual deliverable

October 25th: TL5 AI specialist presentation

October 28th: TL6 presentation due in git

November 4th: TL6 presentation

**December 11th: Final Demo**

1. Glossary of Terms

**roguelike:** is a game genre named after rogue (1980, Epyx). Specific features that make a game a roguelike are: auto or procedurally generated maps, little to no save progression between runs, and an emphasis on resource management.