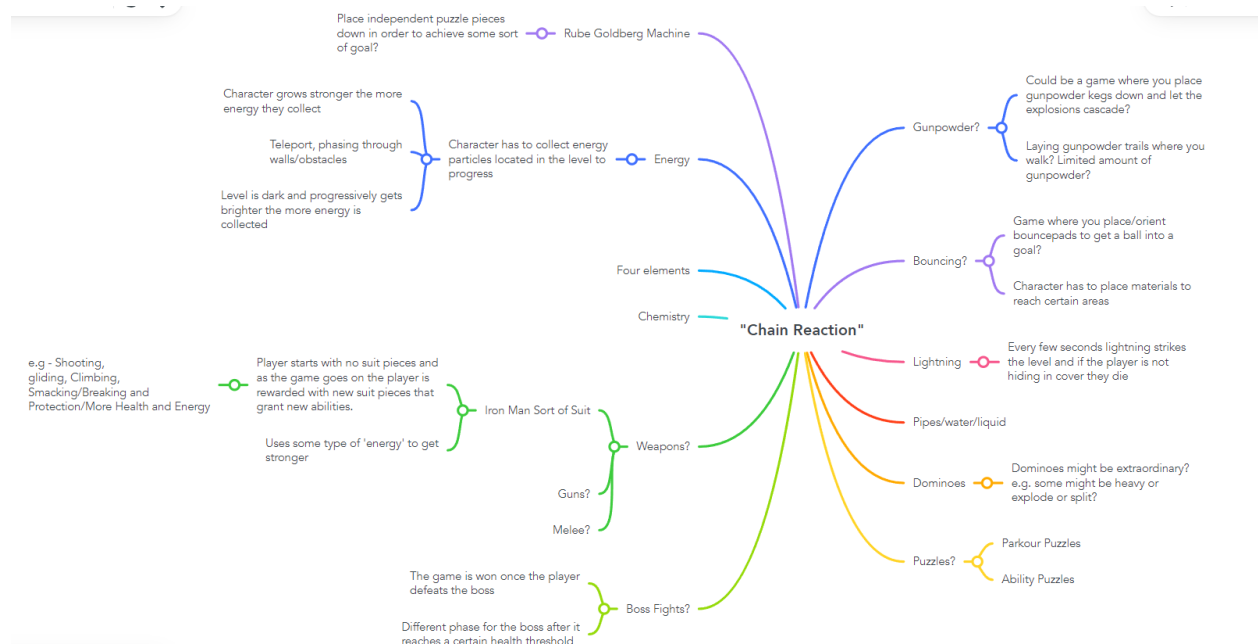


# Assignment 2 Documentation

## Introduction

Our game is based on the old school arcade-style Bomberman games where the gameplay revolves around strategically placing bombs to traverse around a grid-like maze and collect points along the way.

We came up with the idea for this game through a generated prompt "Chain Reaction" and expanded on that concept with similar ideas. The mind map for these ideas is below.



## Gameplay Mechanics

In our game, the player controls a character that can be moved in the four cardinal directions using either the arrow keys or WASD. The game world is set as a matrix grid where each cell can hold various objects such as bombs, explosions, the character, and more.

The game's main objective is for the player to traverse the grid strategically. This is done by planting bombs to blow up rocks obstructing the player from getting to the

sweet loot scattered around the grid. The player gains points whenever a piece of loot is collected.



# Game Design Decisions

## The Grid

Similar to the classic style of the Bomberman games we decided to go for a matrix-like grid that makes the game objects easily distinguishable. By limiting each cell in the grid to hold just one object (two temporarily after the player places a bomb), the game becomes easier to follow what is happening, which we believe is important for this style of game.

## Player Movement

We based the movement of the character on the cardinal directions as opposed to having fixed x and y velocities. This allows the player to have easier control over how they want to navigate in the game.

## Object Types

We included several different objects including bombs, gems, explosion effects, rocks, and tiles. These different object types have their own respective classes which include concepts like breakable, collectable etc. As the code is set up in this way, it allows for new unique objects to be added to the game with relative ease.

## Scoring System

The player earns points based on the loot that they have collected throughout the game. Similarly to the object types, the code is designed to be extensible and can be used to modify the score based on certain conditions. Currently the score tracker updates only when loot is collected, and the player is caught in an explosion. However, the code could be easily modified to work if new additions were added to the game.

# Contributions

We shared the code in github and the link for the project can be found here:

<https://github.com/JamesCoburn246/Games-Programming-A2>

Below is a brief summary of those contributions made (from README):

Reece Bonnington (15357959)

- Main menu.
- Sprite sheets/animation sheets.
- Player controls/movement.
- Collectable pickup logic
- Scoretracker logic
- Script
- Documentation

Pierce Grant (21018347)

- Explosive placement logic
- Explosive detonation logic
- Picking up collectables code (gems, detonator crates, bomb crates)
- Score display
- Player controls/movement
- Recording gameplay

Josh Cressey (21016238)

- Editing

James Coburn (19044568)

- Parent classes and functionality for props
- Game grid logic
- Level manager
- Explosion logic
- Abstract code and inheritance functions.
- Planning and organization.
- Voice over