

# You Raise Me Up (Working Title)

## Overview

Theme / Setting / Genre

The Elevator Pitch

Influences (Brief)

<Influence #1>

<Influence #2>

<Influence #3>

Core Gameplay Mechanics Brief

Targeted platform

## What sets this project apart?

## Player Interaction Patterns and Modes

<Player Interaction Pattern>

<How many players / how it works / etc.>

<Player Modes>

<Player Mode #1>

<Player Mode #2>

## Objectives

<Primary Objective #1>

<Secondary Objective #1>

## Procedures

## Rules

## Resources

## Core Gameplay Mechanics (Detailed)

<Core Gameplay Mechanic #1>

<Core Gameplay Mechanic #2>

<Core Gameplay Mechanic #3>

<Core Gameplay Mechanic #4>

## Story and Gameplay

Story (Brief)

Story (Detailed)

Gameplay (Brief)

Gameplay (Detailed)

## Assets Needed

2D

3D

Sound

- Code

- Animation

**Remark:** The original game pitch included a story set in the 19th century based loosely on the time of pioneers making their way to the west of America. For the sake of scope, we decided to focus on the core gameplay mechanics and put story and setting aside for now. If time allows it, we might try to reintegrate the story-elements back into the game.

## Overview

### Theme / Setting / Genre

- **platformer with classic elements and minimalist/abstract art style**

### The Elevator Pitch

While most commercial platformer games demand the player to beat the level by jumping, running and using various other skills their character, in “You Raise Me Up” the player **becomes the level**.

They will help a computer-controlled character beat a level by positioning and transforming platforming objects intelligently, unveiling new paths to finish the level and try to protect the characters as good as possible. There will be no major story that distracts the player from the original gameplay mechanics.

### Influences (Brief)

- **Super Mario/Sonic/any platformer..**
  - Game
    - “You Raise Me Up” itself is supposed to be a platformer
    - But has the intention to ‘break the rules’ of conventional platformer games and force the players to take a new and interesting role
    - Still, the character they will watch and help/obstruct and the rest of the world will follow the rules of commercial platformers (i.e. 2D style, in fact 3D, walking & jumping, climbing, potentially shooting)
- **Super Mario Maker**

- Game
  - In Super Mario Maker, the player designs platforming levels using the system and assets of the Mario games. “You Raise Me Up” is similar in that the player has control over many of the platforming options that the character will do.
  - The key difference here is that the player will be constructing platforms for the character to surpass obstacles in real time
- **The Oregon Trail (see remark at the top)**
  - Game/actual Oregon Trail + the period of time
    - One of the first games I played -> fond memories
    - It is brutal and rewarding at the same time -> influences difficulty
    - Interesting setting that is not used too much, I wanted to pick that up

### Core Gameplay Mechanics (High-Level)

- Player **controls multiple elements of the level**. The player takes items out of an inventory and places them on the screen in real time.
- **Physics** can be manipulated to make new areas accessible
- There is an **implicit time limit** as the computer guides the character(s) through the world - if the world isn't set up correctly in time, the character will most likely not beat the level

### Targeted platform

- Windows/Mac/Linux. Desktop game

### What sets this project apart?

- **Unique** Gameplay - the player becomes the level - the game breaks with the traditional concept of platformers
- **Engaging** Difficulty - this is not meant to be a Candy Crush-style pastime game, to finish this, you'd have to have some old school video game skills

## Player Interaction Patterns and Modes

- **Single Player vs. Game**
  - One player vs the game. Even when the player is “helping” the game-controlled characters, he is in fact playing against the game as it will just keep on trying to beat the level, putting pressure on the player to design the path accordingly
- **One Play Mode (if there is time potentially a second mode)**
  - Raise Me Up
    - The player has to help the computer-controlled characters complete the level by getting rid of obstacles and building new paths like bridges, etc.. The initial game pitch included another mode called “Push ‘Em Down” in which the player would have to prevent NPCs from completing the level by creating obstacles and removing all paths that lead to the finish. We will focus on building this game mode because it seems more natural to help the NPCs actually achieve something and also for the sake of scope. If time allows it and adjustments to be made are not too major, we will consider implementing “Push ‘Em Down” as well.

## Objectives

- **Help the Characters complete each level**
  - Finishing each level, the characters make their way along all the levels
- **Discover creative ways to surpass obstacles**
  - Any individual obstacle will have multiple solutions on how to get past them. Part of the game will be learning which ways are best in a given situation.

## Procedures/Actions

- The player can choose from a “toolbox” of items per level, these may be platforms, ladders, springs/trampolines or small interactive objects such as explosives. These can be dragged and dropped on the map. Items will not be reusable, i.e. once placed on the screen they cannot be moved by the player or put back into the inventory.

- The options in the toolbox will, despite varying in amount and tool type per level, enable the player to find multiple different ways to complete a level. (This is not really a procedure)
- The player can pause the game (but not move/transform anything while doing so) to inspect and get to know the level

## Rules

- The player has a fixed number of lives at the start of the game. If the player loses one of those lives during a level, this carries over to the next level. If the player loses a life and has none left, they lose the game.
- The characters also have a certain amount of health which will be reduced when they get hit by obstacles along the level. If the health drops to 0, the character will die.
- A character's health will instantly be reduced to zero if he falls off the level at some point
- The character's health is restored to max upon completion of a level
- For each level, it will take the characters some time to arrive in the level. During this time, the player can set up platforms and manipulate the level. He can also do all of the above after the players have already entered the level.
- A level will be unlocked after completing the one before.
- Characters can use objects (explained below)
- The character will simply walk straight to the right of the screen no matter what it sees in front of it. Walking upon a usable object will trigger the AI to use it (e.g. climb a ladder, use a spring for jumping). When colliding with a harmful object, the character will take damage. If the character hits an obstacle that cannot be passed, they will bounce back slightly allowing the player to make adjustments to eventually pass the obstacle. Bouncing back from an obstacle will reduce the character's health.

## Resources

- Limited amount of lives
- Limited amount of tools in toolbox/inventory
- Implicit time limit (characters keep walking regardless of state of level)
- Limited amount of interactable objects in a level (they are all affected by physics and could be used to help/harm the characters)

## Objects/Entities

- Level objects/entities (some might possibly involve scripting if they are destroyable/otherwise manipulatable):
  - Bridges, Ladders, Blocks, Water, Barrels, Boards/Planks, ...
- “Harmful”, moving Entities: Fire, Enemies, Falling Blocks
- Player Entity: Cursor to drag and drop items
- The character: not directly played

## Core Gameplay Mechanics (Detailed)

- Player **can drag and drop multiple objects to manipulate the level**
  - Drag & Drop
    - The player has access to a toolbox with a limited amount of different platforms and items which they can use to manipulate the level. The levels are initially designed such that when the computer-controlled character passes it, they will fail by running into some obstacle (i.e. fire, enemy, ...) or some kind of abyss. It is the player’s task to place the platforms and items so that some of these obstacles can be avoided when the characters enter the level. During the course of the game, the player will unlock more items types for his toolbox such as as ladders, slides, trampolines, springs etc. ...
  - How it works: Drag & drop item from toolbox + Collision detection or other action
    - The platform’s position can be altered by dragging and dropping. Once dropped, an item cannot be picked up an moved again by the player. As it is just part of the level like any other level object/entity, collision detection and other actions like climbing (ladder) or sliding (slide) will be triggered upon contact. Some elements also have special functions that are triggered upon placing them in the object (e.g. TNT exploding after a few seconds)
- **Implicit Time limit**
  - Characters run through the level relentlessly
    - The characters are not really smart - they will just keep walking through the level, even straight into an abyss if the player does not

stop them from doing so. That being said, the time between a character starting the level, and arriving at certain obstacles is more or less set and static.

- How it works: Easy, there is basically no 'intelligent' AI
  - For each character and level, their route and the actions they take when colliding with an object or have to interact with an object have to be defined. The interactions with objects will be the same regardless of the level. That means, that whenever a character encounters a ladder, he will try to climb it.

## Story and Gameplay

Story of the game has been drastically reduced/removed in order to reduce the scope of the project. Since narrative is not a primary aesthetic of the game, this change allows us to improve other systems with greater focus. The original story idea for the game is below.

### Story (Brief)

Pioneers in the mid-19th century hoping to make their fortune in far away Oregon, having to go the infamous Oregon Trail with all its obstacles and adversities.

### Story (Detailed)

Between the 1840s and 1860s, ten thousands of pioneers and settlers made their way from the settlements surrounding Missouri River through more than 2000 miles of uninhabited land to the fertile lands of the pacific northwest. The route led through the Great Plains and by first small, later torrential rivers up to the Rocky Mountains and beyond.

The game's story is closely based on its original ancestor's story: There are four party members, a family of farmers consisting of two parents and two children (which basically represents the hardest difficulty in the original game - the names can be chosen by the user) makes its way along the Oregon Trail. Along the way, they'll pass settlements, forts and native Americans who and, depending on how well the player does during the course of the game, the family will either have a great time on their way, or the worst time of their life, possibly losing their children or parents.

Different from the original story though, there will be other pioneers, mainly bankers (representing the easiest difficulty in the original game) who also want to make their

way up the trail - the player will try to change their fortune and make prevent them from ever reaching the promised land of Oregon.

### **Gameplay (Brief)**

The player has to beat a number of levels to help the character make it to end of the game. Each level consists of classic platformer elements but instead of controlling the character beating the level, the player controls platforms and manipulates the level in real time to create paths for the computer-controlled character. He can do so by placing objects from a set inventory of items including platforms, springs, and explosives into the level. Each of these items can be used in a variety of circumstances to create multiple solutions to overcoming obstacles in the level. When the character passes the final obstacle of a level, he will walk off-screen and continue on the the next level.

### **Gameplay (Detailed)**

Once a level is entered, the player must to familiarize himself with layout of the level and ask questions like: what is a possible path from start to finish in this level? What do I have to do in order to allow the characters to reach the end? Where will I have to place the platforms and other items? Which terrain can manipulate using items? Where can the characters go? How do I make sure I don't use all of my items too early in the level? While the player is making his choices, manipulating items and maybe causing physics to create new paths or destroy existing ones, he will see a character progressing through the level. Characters could progress through the level safely, or can take damage and can potentially lose a life, triggering a restart at the beginning of the level (or a checkpoint).

Characters will show the same kind of behavior for similar objects, which the player will get to know during the first or first two levels. That is, whenever they approach a climbable object, they will climb it. Some levels might contain pickups like additional health. It might be a strategic choice and a question of timing, when to lead a character with potentially low health on a path with a additional health to increase their chances of making it through the level.

Some items that the player can place will not be permanent, such as TNT. The player can place TNT in order to change the level so that the character can take a different path. For example, the player can blast a hole in the ceiling of a tunnel and then place a spring to allow the character to escape. Similarly, the player can use TNT to remove enemies that may hurt the player.



The player can use planks to cover up holes in the ground and can use planks as slopes to go up or down high obstacles. The player can use springs to jump over obstacles or to reach a higher path.

The player seeks to avoid letting the character die by running into spikes, which requires the player to get the character to avoid these spikes by going either over or under them, depending on the situation.

The player can use trampolines, should they have them in their inventory, to allow the character to survive higher falls without taking fall damage.

The character will pick up items such as health potions that lie on the ground.

There will some enemies that go from the right to the left on the screen, or stay put.

They can be killed by falling, explosives, or being crushed by rocks.

## Assets Needed

- **2D**

- Textures for all kinds of world objects
- Environment Textures
- Images for Pop up dialogues
- Explosives (TNT)
- Planks
- Springs
- Life icon (i.e. if you have two lives left, shows two of these in the upper left corner)
- Health Bar
- Inventory Box
- Pickups like health potions
- **Characters List**
  - The AI character
- **Environmental Art Lists**
  - Enemies
  - Spikes
  - Fire
  - Boulders
  - Unstable platforms

- **Sound**

- Music to be played during each level
- Sound List (Player)
  - Player TNT explosions
  - Feedback sounds for dragging and dropping items
- Character Hit / Collision Sound list

- Character has non-lethal fall (maybe sound during a fall)
- Character eats food (gains health)
- Character takes damage
- Character bumps into wall
- Character Death sound list
  - Character falls into abyss/water
  - Character hits spikes or falls (OUCH!)

Template based on work by: Benjamin Stanley and Alec Markarian