

Game Name:

Architect of Goo

Genre:

2d educational puzzle game

Game Elements:

* Math
* Puzzles

Player:

It is a single player game.

# TECHNICAL SPECS

Technical Form:

2D Graphics (flat)

View:

The camera view from which the player will experience the game is a side-view

Platform:

WebGL

Language:

C#

Device:

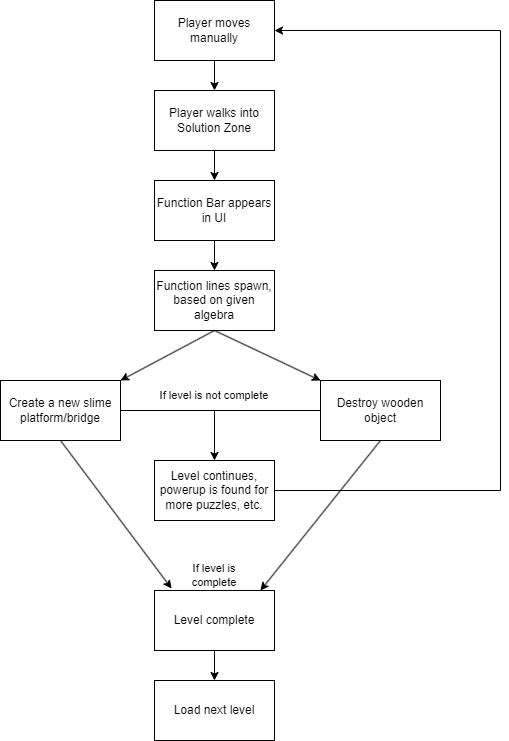
PC

# GAME PLAY

“Architect of Goo” is a math-based puzzling game that tasks the player, as the slime, to escape from the factory they find themselves in. They can use their slime based powers to create platforms, bridges and other contraptions in order to safely find their way out of the factory.

In order to place down these slime contraptions, the slime needs to carefully calculate where the contraptions need to be placed down. It is up to the player to use different algebra to help the slime place their contraptions.

Gameplay Loop



Gameplay Phases

These are the different gameplay phases a player will go through per puzzle

* Exploration phase
* Experimentation phase
* Execution phase

Gameplay Elements

* Slime
  + The player
* Solution zone
  + The area where the player is able to make a slime bridge or attack wooden structures with acid. A solution zone is displayed differently based on what power the slime is able to use (create a bridge or destroy something with acid).
  + The player is able to rewind their created structure within the solution, gaining back their used items and allowing them to create a new contraption. Using a rewind depletes the player’s health partially.
* Formula lines
  + The lines that get created when formulas are entered in the solution bar that aid in the making of bridges and attacking
* Barrier
  + An area where the player can’t move through or make bridges
* Raster
  + An area where the player can move through and make bridges but formula lines cant go through
* Pickups
  + Lives and variables the player can pick up. These variables are one time use.
* Boxes
  + Interactable physics objects that can be used to press buttons and other things
* Wooden structures
  + Breakable wooden structures that prohibit movement through the levels or hide pickups, boxes, and/or solution zones.

Gameplay Outline

The game outline for this project looks as follows:.

* Opening the game application
  + The user is greeted with a main menu that has buttons to start the game and settings.
* Settings
  + A basic settings menu with different options.
* Story synopsis
  + Before the first level there will be a short story synopsis to get the player invested about the slime’s journey.
* Game elements
  + The gameplay is a blend of math and regular gameplay, aiming to get the player to learn more about math by simply playing the game.
* Game levels
  + The game sports a number of levels, all based on different solutions in mind in order to handle several different algebra questions.
* Player’s controls
  + The slime moves manually, the player can control the slime’s powers while it is in a “Solution Zone”.
* Winning
  + The player wins when the slime enters a suction pipe, leading to the next level.
* Losing
  + The player loses if the slime dies.
* End
  + The player successfully ends the game if the slime manages to escape the factory by completing enough levels.
* Why is all this fun?
  + The game is fun because it takes the rewarding gameplay of a puzzle game and combines it with the challenges of algebra.

Key Features

Key features are a list of game elements that are attractive to the player.

* Placing platforms based on algebra functions
* Interacting with items in the inventory in order to make functions
* Sticking to slime platforms regardless of gravity
* Game Objects like walls and objects that interact differently with the functions

# 

# DESIGN DOCUMENT

Design Guidelines

This project is a serious game requested by our client Mr. van Crasbeek. The given requirements are pretty open, with the few requirements being:

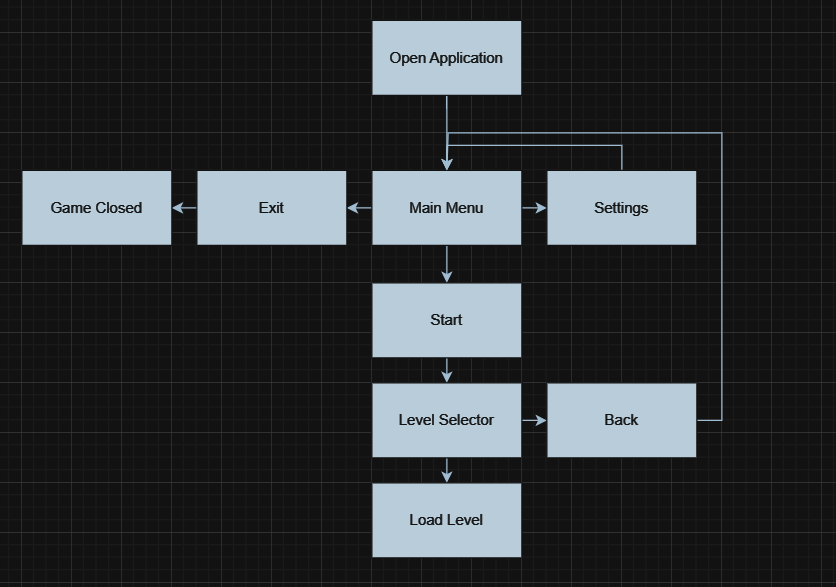
* The gameplay must be a blend of regular gameplay and math
* The algebra needs to be on a MAVO/HAVO level

Game Design Definitions

* The main focus of the gameplay is how the player creates different algebra functions in “Solution Zones” in order to solve the puzzles in the levels.
* The player loses if they touch lava or any other damaging obstacles in the level. After which the level restarts.
* At the end of the level there is a pipe that the player enters that sends them to the next level.
* The player wins by reaching the end point of the level.

Game Flowcharts

Menu

Player Definition

* Use the Player Properties section (below) to define the properties for each player. Player Properties can be affected by the player’s action or interaction with other game elements. Define the properties and how they affect the player’s current game.
* Use the Player Rewards section to make a list of all objects that affect the player in a positive way. Define these objects by describing what effect they cause and how the player can use the object.

Player Definitions

In “Architect of Goo”, the player has:

* Lives
* Inventory with different variables
* A function bar for solving puzzles

Player Properties

* The player can interact with “Solution Zones”
  + On entering a “Solution Zone”, the player can make functions using the items in their inventory.
* The player can create platforms
  + The player can, upon entering a “Solution Zone”, use functions to see if an intersection is at the right grapple point.
* The player can destroy wooden obstacles
  + The player can, upon entering a “Solution Zone”, use functions to see if a function intersects with a wooden obstacle, allowing the player to destroy it.

Player Rewards (power-ups and pick-ups)

Players can:

* Pick up lives
* Pick up variables that augment formulas

User Interface (UI)

Below is an early concept of how the UI will look in “Architect of Goo”: 