**Attack on Blob: Divide and Conquer**

Developed By:

Renegadeware

1.0.0

# 

# Table of Contents

Contents

[Table of Contents 1](#_Toc192187125)

[Game Overview 3](#_Toc192187126)

[Target Learning Objective (LO) 3](#_Toc192187127)

[Demographics - Target Audience 3](#_Toc192187128)

[Genre / Theme / Setting 3](#_Toc192187129)

[Core Gameplay Summary 3](#_Toc192187130)

[Look and Feel 3](#_Toc192187131)

[Target Platform(s) 4](#_Toc192187132)

[Game Flow 5](#_Toc192187133)

[Blob Battle 5](#_Toc192187134)

[Summary 5](#_Toc192187135)

[Mechanics 5](#_Toc192187136)

[Losing Gameplay / Incorrect Concept Understanding 6](#_Toc192187137)

[Mockups 7](#_Toc192187138)

[Part 1 – Large Division Lesson 13](#_Toc192187139)

[L.O. Concept(s) Covered 13](#_Toc192187140)

[Summary 13](#_Toc192187141)

[Part 2 – 3 digits Dividends and 1 digits Divisors 13](#_Toc192187142)

[Summary 13](#_Toc192187143)

[Part 3 – 4 digits Dividends and 1 digits Divisors 14](#_Toc192187144)

[Summary 14](#_Toc192187145)

[Part 4 – Large Division Lesson #2 14](#_Toc192187146)

[L.O. Concept(s) Covered 14](#_Toc192187147)

[Summary 14](#_Toc192187148)

[Part 5 – 3-4 digits Dividends and 1-2 digits Divisors 14](#_Toc192187149)

[Summary 14](#_Toc192187150)

[Part 6 – 3-4 digits Dividends and 2-digits Divisors 14](#_Toc192187151)

[Summary 15](#_Toc192187152)

[Part 7 – End 15](#_Toc192187153)

[Summary 15](#_Toc192187154)

[LO Concept Coverage 15](#_Toc192187155)

[Academic Concepts 15](#_Toc192187156)

[Legends of Learning Required Content Practices 15](#_Toc192187157)

[Checklist Overview 15](#_Toc192187158)

[Connection Between Gameplay and Learning 17](#_Toc192187159)

[Role of Text in Learning 17](#_Toc192187160)

[Characters - Diversity 18](#_Toc192187161)

[Technical 18](#_Toc192187162)

[Development Hardware/Software 18](#_Toc192187163)

[Asset Summary 18](#_Toc192187164)

[Music and Sounds 18](#_Toc192187165)

[Art Style 19](#_Toc192187166)

[Mockups 19](#_Toc192187167)

[Schedule for Development + Delivery 19](#_Toc192187168)

[Story / Narrative 20](#_Toc192187169)

[Back Story 20](#_Toc192187170)

[Plot Elements 20](#_Toc192187171)

# Game Overview

### Target Learning Objective (LO)

* [5.NBT.B.6: Divide Whole Numbers with 4 Digit Dividends and 2 Digit Divisors](https://drive.google.com/file/d/1xpJdZCLiwQyad0TlFfQQ8N3fJn_qnZkj/view)

### Demographics - Target Audience

* Ages 8-11 (Grade School)

### Genre / Theme / Setting

* Puzzle game focused on math skills with a scoring system on how efficient the player performs. The metrics being the number of mistakes and how many steps the player needed to beat the level.
* Earth is once again in grave danger as a blob from a lab grew out of control till it covered an entire city. It is up to the player to quell its growth by using division techniques with large numbers.
* The game takes place in a city where the player confronts various blob titans. The visuals will vary based on each blob, with organic abstract shapes and patterns.
* These blobs are neutral creatures that simply grew out of control. There will be blobs within the blob that represents numbers, and the blobs the player generate will have a distinct look.

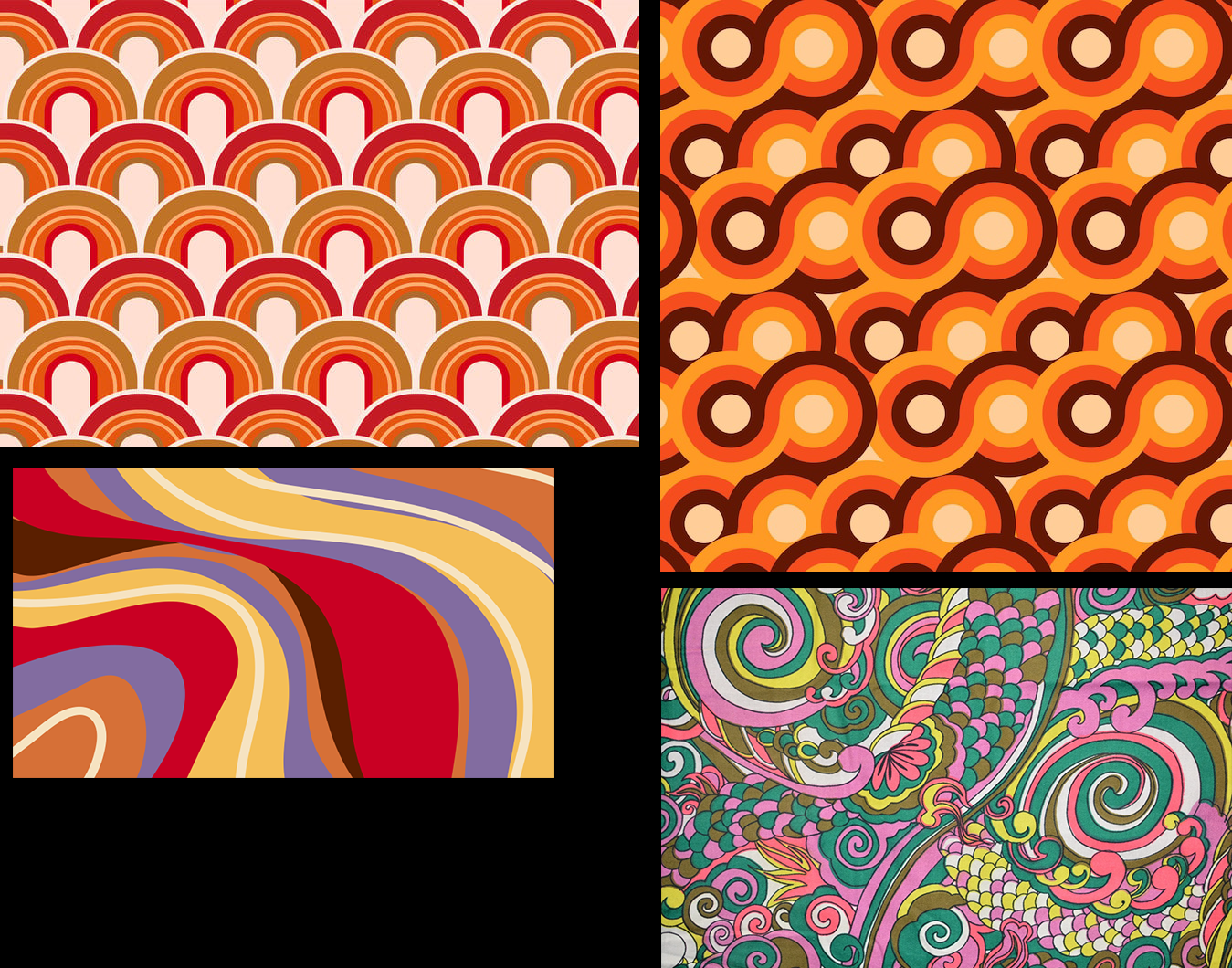
### Core Gameplay Summary

The game is broken up into two parts: lesson and blob battle. There will be at least two lessons on how to break up the division equation. The following battle would enforce the lesson. After each battle, an interlude will be displayed to convey a progress in the game, and some sense of accomplishment.

During the blob battle, the player is tasked to eliminate a mega blob. The gameplay takes place within the mega blob, where two numeric blobs appear for the player to solve. Solving the division of the numeric blobs will damage the mega blob. The damage amount will be based on how well the player performs by making as little to no mistake, and using as few steps as possible. The score will then be based on the amount of damage done, with bonus to having no mistake. Once the mega blob’s health is reduced to zero from damage, the battle is finished.

### Look and Feel

* Colorful vibrant abstract shapes with a city backdrop under a starry night sky.
* Nonchalant blobs with simple expression, and gelatinous circular body.
* Blobs have a distinct look to represent if they are the divisor, dividend, or solved numbers.



Examples of shapes and patterns used as a background in the game.

### Target Platform(s)

* WebGL with iPad support – Since the game is completely mouse driven, there should be no issue with playing the game in any platform that supports mouse or touch input.

# Game Flow

## Blob Battle

### Summary

* The player is tasked to defeat the mega blob by reducing its health to zero, which can be done by solving the division equation of blobs spawned.
* Two blobs are spawned at a time in an enclosed circular board (represents the mega blob), a number for the divisor and another for the dividend.
* The player can split the dividend blob by clicking on it. Upon clicking, a prompt will be shown to give the player an option on how to split the dividend:
  + Tenths place value
  + Partial quotient: X amount multiplication of the divisor
  + Cancel
* Upon picking a split method, a prompt will appear where the player will input a number to
* Once the dividend blob is split, a new one will appear with the value reduced from the original blob. A replication of the divisor blob will also appear when applicable.
* The player can drag a divisor blob to a dividend blob, and vice-versa to merge the blobs into a quotient blob.
* The player can drag two quotient blobs to merge into one blob by solving the addition operation.
* Once there’s only one quotient blob, the attack on blob will commence. The amount of damage on the mega blob will depend on how efficient the player solved the entire equation.
* When the mega blob’s health reaches zero, the player wins the level.

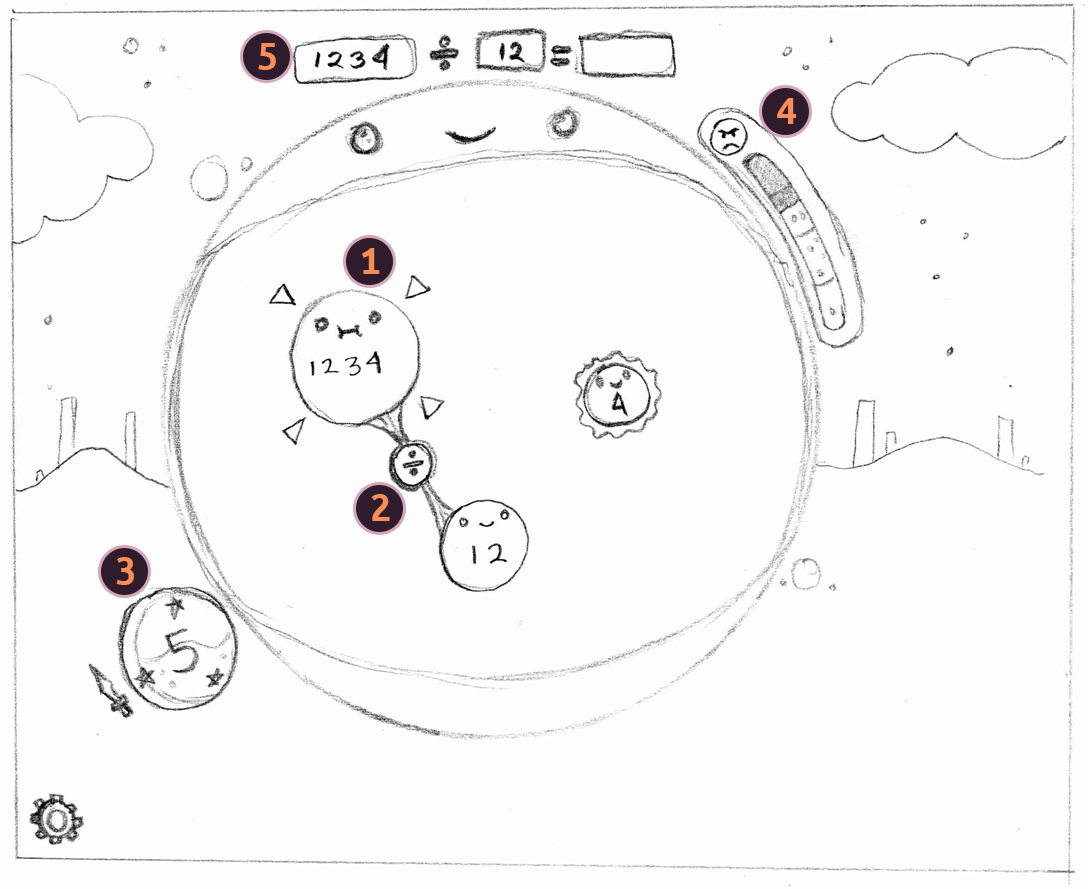
### Mechanics

* Clicking on the dividend blob will initiate the distributive operation (blob splitting).
* Dragging one blob into another will perform the relevant operator for the player to solve. A dividend and a divisor blob will perform division. Two quotient blobs will perform addition.
* Whenever there is a task to input a numeric value, a numpad will appear for the player to use.
* The puzzle aspect of the game is simply how well the player knows their division/multiplication arithmetic. The better they are at solving big divisions, the more effective they will be at defeating the mega blob.
* The effectiveness of the attack on blob will be based on how little the player needs to split the divisor blob. They are even allowed to solve the quotient right away for maximum efficiency.
* There will be an initial attack value for each time two blobs are spawned. The value gets reduced each time the player makes an action: splitting, merging blobs. An attack value is wasted if the player makes enough mistake when performing an action:
  + When attempting to split the blob in tenths place, the value must be divisible.
  + When attempting to do a division operation.
  + When attempting to do an addition operation.
* The less the attack value is, the more rounds there will be for the player to perform, thus allowing for more practice with division.

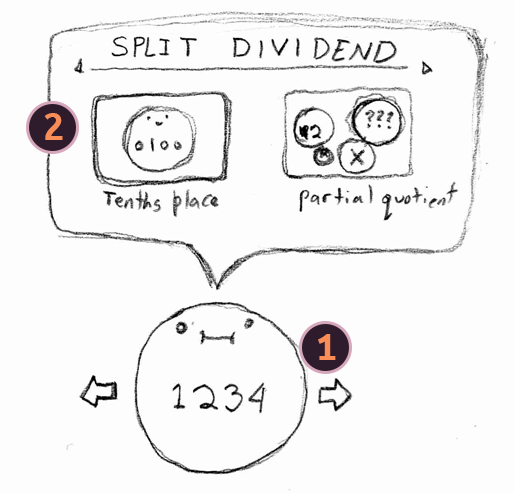
### Losing Gameplay / Incorrect Concept Understanding

* The player is given 3 tries whenever they are attempting to split a dividend blob, or solving an equation. Failing 3 times will cancel out the action, wasting an attack value.
* During tenth place splitting, the game will inform the player if the split value is not divisible.
* When splitting using the partial quotient method, if the value is greater than the dividend, then the game will inform the player to try lowering the multiplication value.
* The game will generally inform the player if the answer is incorrect.
* If possible, the game will give helpful tips if the player keeps failing.
* We can also have blobs spawn with easier division values as the rounds increase.
* The summary UI in the end will show how efficient the player is on each round they performed. A score and a grade will be displayed to summarize the player’s overall performance.

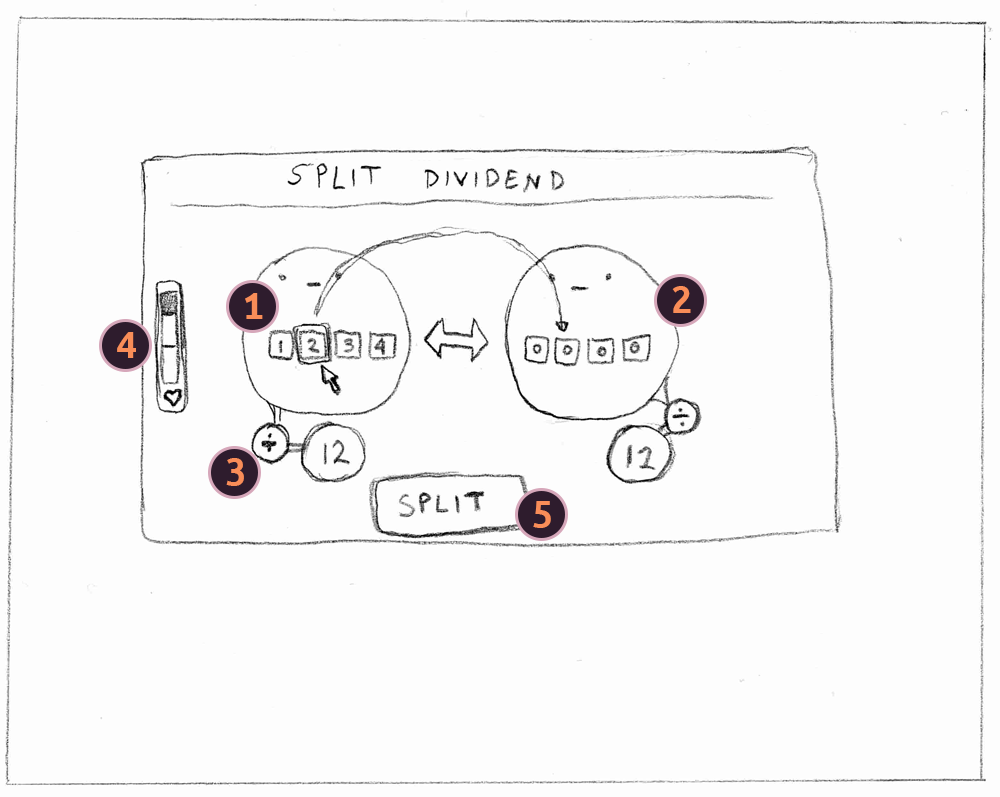
### Mockups



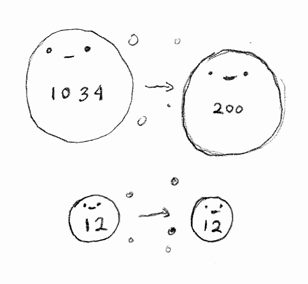
|  |  |  |
| --- | --- | --- |
| 1 – Blobs spawned of different types: dividend, divisor, partial quotient. | 2 – Contextual operator depending on the drag from one blob to another. | 3 – Power level indicator. This is reduced for each action made with blobs. Once the quotient blob is solved, this power value is used to reduce the mega blob’s hitpoints. |
| 4 – Mega blob’s hitpoints indicator. Once this reaches zero, the level is complete. | 5 – Display of the equation the player is trying to solve. (Possibly can be changed depending on the splitting blobs if doable.) |  |



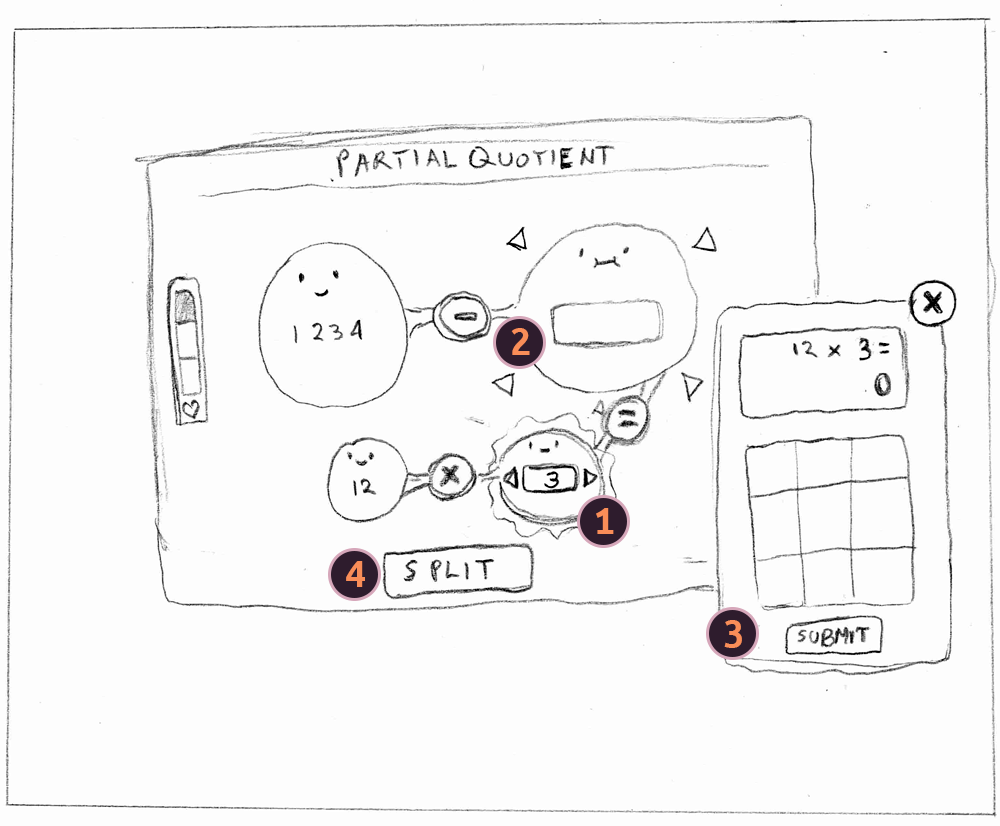
|  |  |
| --- | --- |
| 1 – Quotient split context when hovering mouse to a quotient blob if can be divided by divisor over more than one factor. Clicking on the blob will open up the split action context. | 2 – Split actions if both types of distributive method are applicable. Clicking on either one will open up their corresponding UI. |



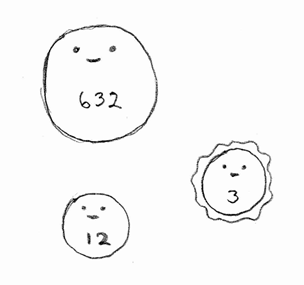
|  |  |  |
| --- | --- | --- |
| 1 – Dividend blob value with each number placed in a clickable UI. Clicking on this will transfer that number to the new blob. | 2 – Numbers clicked from the original dividend blob will move here. An animation will show how the number moves to emphasize the procedure. The player can also click the numbers here to transfer to the original blob. | 3 – Display of division operator and the divisor value. This is here to allow the player to check if the dividend is still divisible as a whole number. |
| 4 – Player hit points that shows how many more mistakes the player can make. If it reaches zero, the split operation is cancelled. | 5 – Clicking on this will evaluate the original and new dividend blob to see if they are still divisible as a whole number. An error animation plays, otherwise the game will proceed will splitting the blobs. |  |



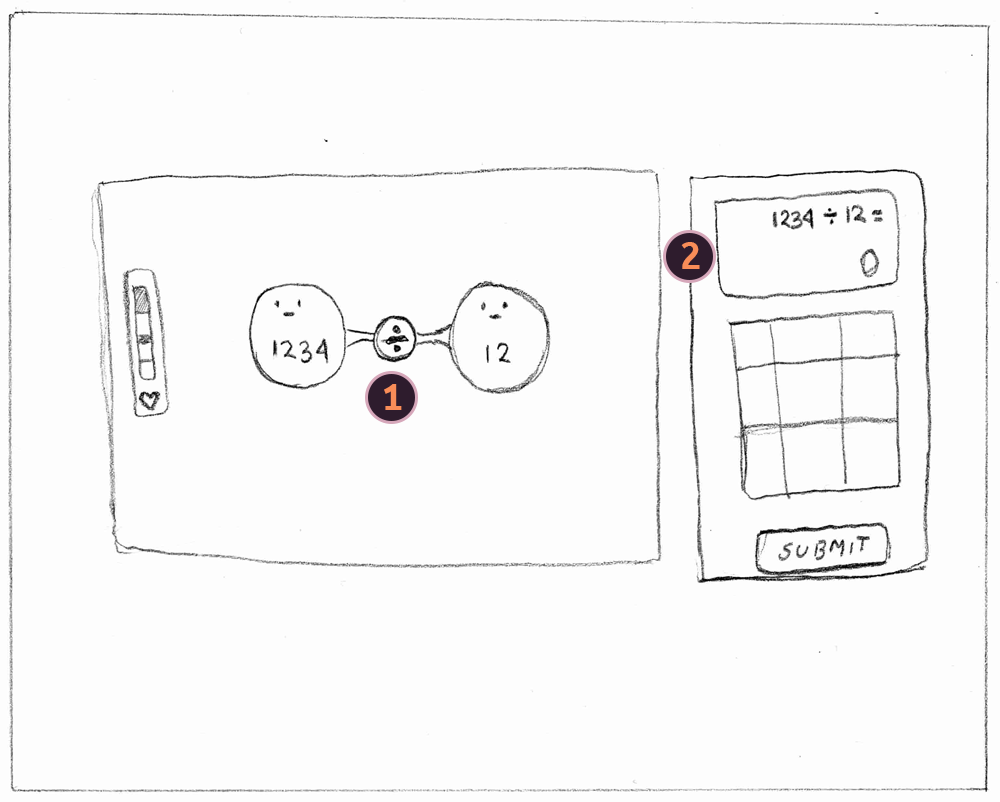
Once the dividend blob is split, a new one will spawn. A new divisor blob will also spawn.



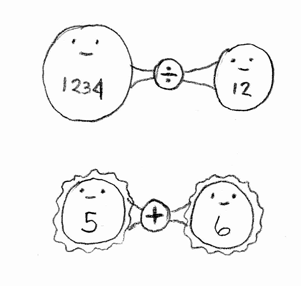
|  |  |
| --- | --- |
| 1 – Interface for changing the number of the partial quotient. | 2 – Displays the value of divisor times the partial quotient. |
| 3 – Input for determining the display on 2. | 4 – Once there’s a value in 2, the player can click on this to evaluate the whole equation. An error will occur if the value on 2 is wrong or greater than the original dividend. Note: some iterations will be needed here to determine the best possible gameflow. |



Once the partial quotient split has been executed, the blob dividend value is updated, and a quotient blob is spawned.



|  |  |
| --- | --- |
| 1 – Display of the equation after the player drags one blob to another compatible blob (dividend to divisor, quotient to quotient) | 2 – input for player to answer the equation. Upon clicking submit, if the answer is correct then the two blobs will merge into a new quotient blob. If wrong, an error animation will play, and the player hitpoint is reduced by one. If the player hitpoint reaches zero, then the operation is cancelled. |



The two possible operations when merging blobs. Blob merging is restricted to their type: dividend to divisor, quotient to quotient.

## Part 1 – Large Division Lesson

### L.O. Concept(s) Covered

* Using distributive property to solve large number division.
* Area model to visualize the distributive property.
* Using placement value to split the dividend.

### Summary

* This part of the game will have a quick summary of the distributive property, and a visualization of division by area model.
* The lesson will show a way to split up the number by using tenths place.

## Part 2 – 3 digits Dividends and 1 digits Divisors

### Summary

* The game will start with a simple 2 digits dividend, and 1 digit divisor. This part is mostly to show how the game fundamentally works.
* The next set of numbers will then be a 3 digits dividend, and 1 digit divisor. From here, the player is asked to split the dividend using the tenths place method. The function here is restricted to interacting with the dividend for the purpose of showing the mechanics of splitting the blob.
* The rest of the game will proceed as normal, telling the player that the objective is to ultimately reduce the mega blob’s health to win.

## Part 3 – 4 digits Dividends and 1 digits Divisors

### Summary

* This is more of a continuation of the game now with 4 digits dividends.
* The splitting mechanic is restricted to using the tenths place method.

## Part 4 – Large Division Lesson #2

### L.O. Concept(s) Covered

* Using multiplication of a number with the divisor to reduce the dividend (partial quotient).

### Summary

* This part of the game will show another way of splitting the dividend by figuring out a partial quotient by multiplying the divisor with a number.
* The lesson will demonstrate this step-by-step using an area model, as well as tying it with the distributive property.

## Part 5 – 3-4 digits Dividends and 1-2 digits Divisors

### Summary

* The game will restrict the player into splitting the dividend using the partial quotient method. From there, the game will guide the player through the steps.
* Once the entire splitting process is complete, the game will proceed as normal.
* The splitting method will be restricted to just using partial quotient for this level.

## Part 6 – 3-4 digits Dividends and 2-digits Divisors

### Summary

* The player will pretty much have free reign here to solve the divisions. All splitting methods are available, as well a being able to solve divisions directly.
* Once the level is complete, the game will proceed to the end scene.

## Part 7 – End

### Summary

* Ending cinematic where all the blobs are subdued.
* A summary of the player’s performance from all the levels.

# LO Concept Coverage

### Academic Concepts

* Solve whole number division problems with whole-number quotients with up to four-digits and two-digit divisors, using strategies based on place value.
* Solve whole number division problems with whole-number quotients with up to four-digits and two-digit divisors, using strategies based on the properties of operations.
* Solve whole number division problems with whole-number quotients with up to four-digits and two-digit divisors, using strategies based on the relationship between multiplication and division.

# Legends of Learning Required Content Practices

### Checklist Overview

[Google Doc Reference](https://docs.google.com/document/d/10yED8ZwFXOWjwvroqZxaHn1A6utMDncaFwmyc8dqc-g/edit?usp=sharing)

|  |  |
| --- | --- |
| **ITEM** | **COVERED** |
| Players should learn and be held accountable through gameplay-based problem solving and experience. Players should not be learning primarily through text-based instruction or assessment items. |  |
| Game does not include multiple choice assessment items. |  |
| All instruction is scientifically and mathematically correct. |  |
| Confirm that the game is linked to 2/3 or 5 main concepts of the total, whichever is greater. Confirm that the linked main concepts are correctly covered in the game. |  |
| All on-screen words spelled correctly and grammatically correct. |  |
| Vocabulary and reading level appropriate for the lowest grade level within the target audience and grade band. |  |
| Game does not include material that is inappropriate for school. This includes, but is not limited to: violence, firearms, bombs, knives, daggers, blood, gore, smoking, vaping, drug use, any mind-altering substances, alcohol, harm to human-looking characters, harm to animals, insinuating killing or death, ideally they’re always chased away rather than eliminated. If there is conflict with an enemy in game, they are chased away rather than eliminated or killed. (There can be death if it is in the context of the learning objective – ex. The food chain) If you have any questions about this policy and your game, please ask us. |  |
| Game avoids any stereotypic presentation of gender, race, region, or culture. |  |
| Characters are diverse in gender, race, culture, and ability. |  |
| Players cannot simply click through and complete the game without learning. Players should be prompted to re-learn and re-do portions of the game where they had poor results due to less understanding of the academic material. Avoid the word “FAIL” if the student incorrectly understands academic material. |  |
| Academic problems are not consistently repeated. Players are presented with different problems to solve. |  |
| Gameplay mechanic reinforces the academic material, rather than being completely separate from instruction. I.e, there is a focus on academic reasoning rather than concept / question repetition. |  |
| Gameplay is intuitive and a player in the target age range can navigate the game and beat it with enough effort. |  |
| Games should be fun and interesting, designed as non-educational games are designed, with design to encourage players to keep playing. |  |
| Game is between 5 and 25 minutes in duration. |  |
| All text must be large, clear and concise with font sizes that can be read on a small Chromebook screen. |  |

### Connection Between Gameplay and Learning

* The player is directly involved in using the distributive property when needed to solve the division of two large numbers.
* During the first two levels, the player splits up the dividend strictly by tenths place. This helps to understand the process of division by subtracting iteratively. The visuals of how the numbers move from one place to another will help reinforce the process.
* On the later levels, the player can reduce the dividend by multiplying the divisor by a given number. This process will help the player understand the correlation between multiplication and division.
* The player is also encouraged to be efficient in solving the division as little steps as possible to score a bigger attack on the mega blob.

### Role of Text in Learning

* The first lesson scene will do a quick recap on how the distributive property works, and how it can help with solving large divisions. A visual aid will help with this on how numbers are split, as well as an area model.
* The second lesson scene will help with showing the relation between division and multiplication with the help of the area model.
* Other texts in the game will be used to explain how to play the game as features are introduced.

### Characters - Diversity

* A reassuring robot talks throughout the game.
* All blobs are presented in a variety of sizes and colors, with a cheerful demeanor.
* The mega blobs represent each stage, and will have distinct looks and color scheme for the level.

# Technical

### Development Hardware/Software

* All development will be done in Windows 10.
* iPad 6 for tablet testing.
* Browsers: Firefox, Edge, Chrome, and Safari (via iPad 6)
* Game Engine: Unity 2022.3+

### Asset Summary

* This game will be borrowing some assets from a previous project - Attack on Blob: Mega Multiply. There will of course be drastic changes to the gameplay logic, as well as completely new assets for each level.
* Most of the art will be done programmatically with shapes to minimize the game size.
* Game will be delivered via WebGL (targeting desktops and touchpads).

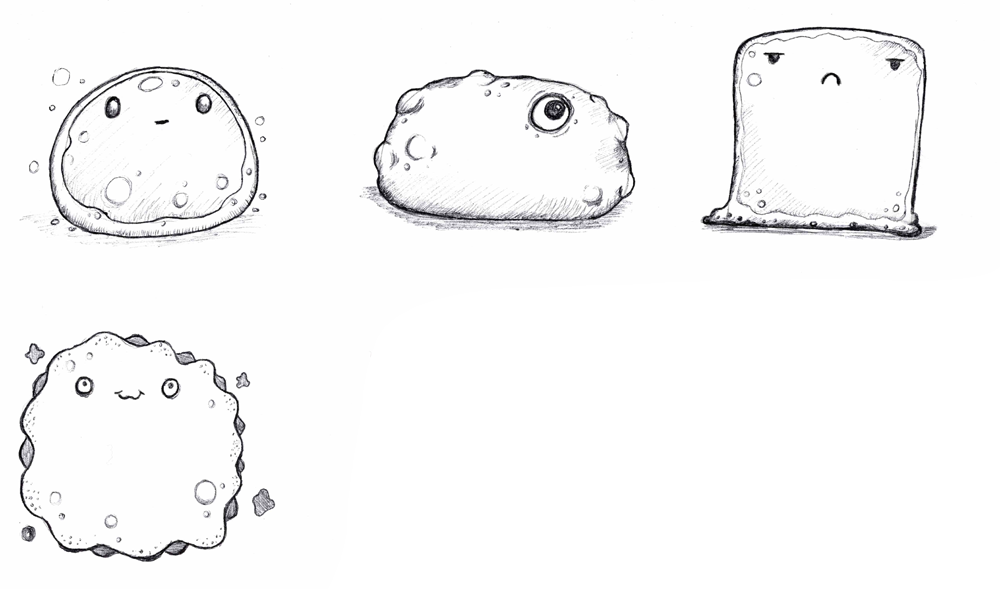
### Music and Sounds

* Music will come from various artists from the internet under the Creative Commons Attribution license.
* We have a growing library of sound effects that are comprised of public licenses, as well as purchased licenses.

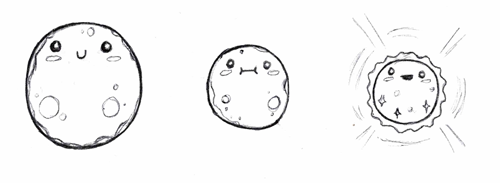
# Art Style

### Mockups

#### Mega Blobs



#### Blobs



# Schedule for Development + Delivery

* Projected coding time: 2 – 3 weeks. (Coding will also accommodate time for implementing art, sound/sfx, lessons, polish)
* Projected art asset time: 2 – 3 weeks.
* Projected lesson implementation time: 1 week.
* Projected polish time (intro, ending, etc.): 2 – 3 weeks.

# Story / Narrative

### Back Story

* Blobs in a lab container shared and ate a mysterious food. In a matter of seconds, they all grew as large as the mountains. Having found new freedom, each travelled across the world, settling in their favorite spot. The player is tasked to take care of these blobs, by reverting them back into their original size.

### Plot Elements

* Each level represents a mega blob.
* After a level is complete, there can be an animation of the blob reverting back to normal.
* The level progress indicator can be displayed as each blob resting in a container.
* The ending will be all the blobs together, but rather than putting them back in the lab, they are released into outer space.