'Forms* really *are relative in this game!' - Kasper Dissing Bargsteen

This page: Table of Contents and Team Member Listing

Table of Contents

1 Game Overview
2 High Concept
3 Unique Selling Points
4 Platform Minimum Requirements
5 Competitors / Similar Titles
6 Synopsis
7 Game Objectives
8 Game Rules
9 Game Structure
10 Game Play
  10.1 Game Controls
  10.2 Game Camera
    10.2.1 HUD
    10.2.2 Maps
11 Players
  11.1 Characters
  11.2 Metrics
  11.3 States
  11.4 Weapons
12 Player Line-up
13 NPC
  13.1 Enemies
    13.1.1 Enemy States
    13.1.2 Enemy Spawn Points
  13.2 Allies / Companions
    13.2.1 Ally States
    13.2.2 Ally Spawn Points
14 Art
  14.1 Setting
  14.2 Level Design
  14.3 Audio
15 Procedurally Generated Content
  15.1 Environment
  15.2 Levels
  15.3 Artificial Intelligence NPC
  15.4 Visual Arts
  15.5 Audio
  15.6 Minimum Viable Product (MPV)
16 Wish List

Game Development Team Members

PRODUCER
Kasper Dissing Bargsteen

PRODUCTION MANAGER
Kasper Dissing Bargsteen

PRODUCTION COORDINATOR
Kasper Dissing Bargsteen

GAME DESIGNERS
Kasper Dissing Bargsteen

SYSTEMS/IT COORDINATOR
Kasper Dissing Bargsteen

PROGRAMMERS
Kasper Dissing Bargsteen

TECHNICAL ARTISTS
Kasper Dissing Bargsteen

AUDIO ENGINEERS
Kasper Dissing Bargsteen

UX TESTERS
Kasper Dissing Bargsteen

1 Game Overview

Title: Forms Be Relative
Platform: MacOS
Genre: Puzzle platformer
Rating: Everyone ESRB
Target: Casual gamer (aging from 8 - 30)
Release date: November, 2019
Publisher: Bargsteen inc.

Description: Forms Be Relative(FBR) brings the concept popularized by the children's game Shape Sorter into the 21st century in the form of a 2D puzzle platformer-game. Each level in the game presents you with a set of shapes such as squares and circles, which fall downwards through a number of obstacles eventually landing in a CollectorBox marked with a specific shape. If the falling shape matches the type depicted on the CollectorBox, a point is earned in the level. The player controls a character on screen which can move around and interact with buttons in the level which have different effects. The first button you encounter is a SwapButton, which will change all the falling circles into squares and vice versa. To complete a level, the player has to activate the correct buttons at the correct times in order for all the falling shapes to land in their corresponding CollectorBox.

2 High Concept

FBR sets the player in a world of shapes and forms. Some shapes fall downwards through various obstacles when each level starts, and the player has to make them fall into CollectorBoxes depicted with the correct shape. Circles should, therefore, fall into boxes marked with a circle. The player moves in an independent part of the world and interacts with buttons that have different effects; fx changing all circles into squares and vice versa. Timing and tactics are key elements of the game.

3 Unique Selling Points

- Challenging gameplay
- Quick levels; good for a quick mental break
- Retro graphics

4 Platform Minimum Requirements

MacOS Standalone
OS: MacOS 10.8+
System: Generally every Mac made after 2004

5 Competitors / Similar Titles

Braid by Number None
PuzzlesCave by MamoriStudio

6 Synopsis

In a retro-world of shapes falling into mismatched boxes for collection, you are the only one able to change the outcome and satisfy the game rules *and* your OCD.

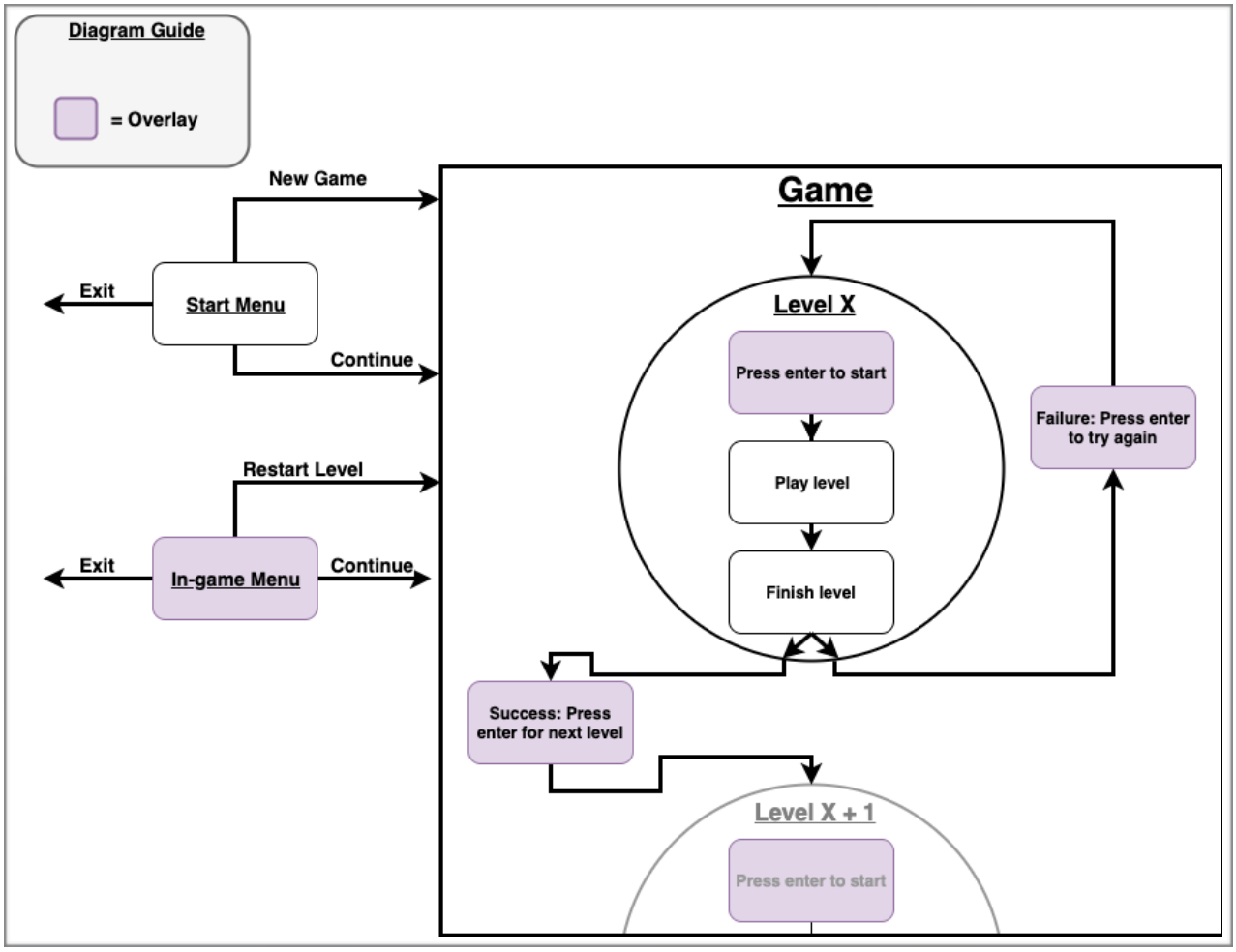
7 Game Objectives

The objective of the game is to transform all the falling shapes at the correct time in order for them to match CollectorBox they are falling into.

8 Game Rules

Each game level consists of two closed areas; the ShapeArea and the PlayerArea. The ShapeArea contains the falling shapes, static obstacles, and CollectorBoxes. The PlayerArea contains the player and various buttons that the player can interact with. The buttons are the only way that the player can interact with the world residing in the ShapeArea. When a level starts, the shapes in the ShapeArea begin to fall down through the obstacles toward a CollectorBox. Each CollectorBox collects a specific type of shape, for example a circle, which is depicted on the outside of the box. If a shape falls into a matching CollectorBox, the player earns one point. A level is completed when the player gets as many points as there are shapes in the level. Partial completion is therefore not possible.

9 Game Structure



I tried to encapsulate all the state details in the diagram. The arrows with New Game and Continue go to level 1 or the highest level previously achieved, respectively.

10 Game Play

10.1 Game Controls

A / Left Arrow: Move left
D / Right Arrow: Move right
Space: Jump
Escape: Show/hide in-game menu

10.2 Game Camera

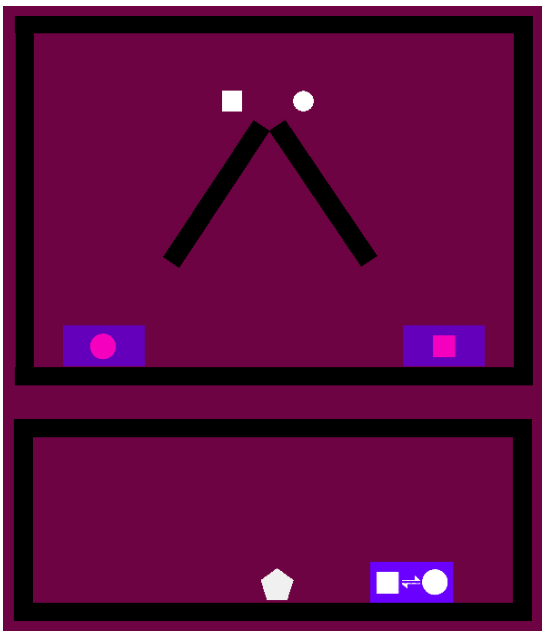
The game camera is static throughout the game and levels.

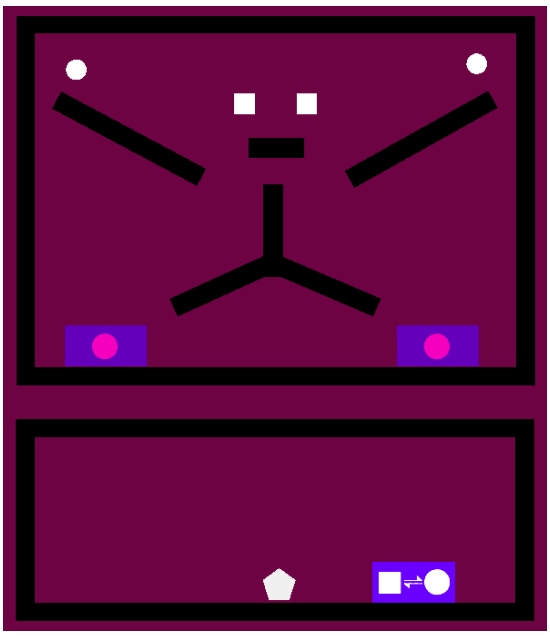
10.2.1 HUD

The score in the current level is displayed in the top right corner. An example can be seen in the figure underneath.

10.2.2 Maps

There are currently two different maps, as shown below. Several levels are built on top of each map in which the amount, type, and placement of the falling shapes and CollectorBoxes vary. The two levels shown in the figures are level 1 (left) and level 4 (right).





11 Players

11.1 Characters

The sole playable character of FBR is the pentagon seen in the PlayerArea. The pentagon can move from side to side and jump. In-air-movement is possible.

11.2 Metrics

Speed: 100
Jump Force: 650
Movement Smoothing: 0.05

11.3 States

Idle: Player stands still.
Move: The player will move according to input from the keyboard.
Level Completed/ Failed: The player is immovable while the result screen overlay is being displayed.

11.4 Weapons

N/A

12 Player Line-up

13 NPC

The falling shapes can be seen as a type of NPC as they are non-static. They are falling throughout the game, and has two primary states: **falling** or **collected**. The collected state can, furthermore, be split into two: **Correctly collected** and **incorrectly collected.**
The sprites used for the shapes can be seen in the two figures underneath.

13.1 Enemies

N/A

13.1.1 Enemy States

13.1.2 Enemy Spawn Points

13.2 Allies / Companions

N/A

13.2.1 Ally States

13.2.2 Ally Spawn Points

14 Art

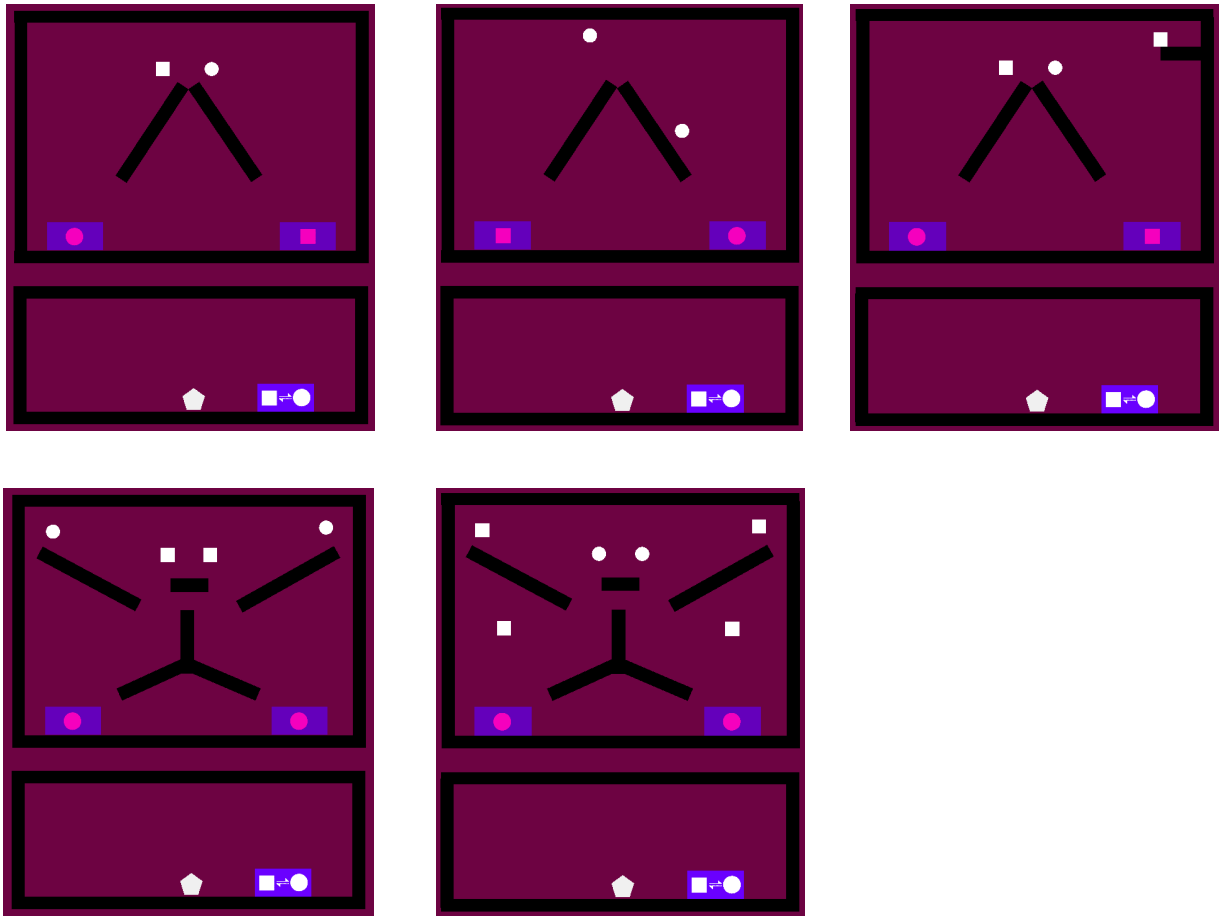
The game-art is very minimalistic and consists solely of basic colored shapes or a combination of such.

14.1 Setting

The game takes place in two enclosed areas: the ShapeArea (top) and the PlayerArea (bottom). The borders of the areas along with the black obstacles inside the ShapeArea are static and unmovable.

14.2 Level Design

Each level varies in terms of amount and placement of obstacles along with the amount, placement and type of the falling objects and CollectorBoxes. The five levels that currently exist in the game can be seen underneath going from level 1 on the top left to level 5 on the bottom center.



14.3 Audio

ItsOkayCaptainMastered3.wav - The background music played throughout the game.
CorrectShapeSound.wav - The sound being played when a falling shapes hits a matching collectorBox.
WrongShapeSound.wav - The sound being played when a falling shape hits the wrong CollectorBox.

15 Procedurally Generated Content

15.1 Environment

N/A

15.2 Levels

N/A

15.3 Artificial Intelligence NPC

N/A

15.4 Visual Arts

N/A

15.5 Audio

The sound effects **WrongShapeSound.wav** and **CorrectShapeSound.wav** are played dynamically based on the gameplay.

15.6 Minimum Viable Product (MVP)

- One level created
- Built for MacOS

16 Wish List

Add more levels
Simply adding more levels using the current mechanics would increase the value and playability of the game.

Add different mechanics
- Add more types of falling shapes.
- Add different types of SwapButtons: [square <-> triangle], [triangle <-> circle] etc.
    - This would enable levels where the player has to press a particular sequence of buttons in order to win (turn circles into squares, then squares into triangles).
- Increase the PlayerArea
    - For example by adding a U-shaped PlayerArea around the ShapeArea, where the player would have to jump on platforms in order to move upwards to reach a particular SwapButton.
- Add different kinds of buttons, for example:

    - Buttons that slow down the speed of the game
    - Buttons that changes the direction of gravity (sideways / upwards)

Improve graphics
- Replace the white shapes with colored 3D-looking shapes that glow.
- Replace the SwapButtons with some kind of a portal.
- Replace the player sprite with an animated shape.

Add particle effects
- When activating the SwapButton:
    - The SwapButton should show that you have entered it
    - The falling shapes should more visibly show that they change their form
- When a falling shape enters a CollectorBox:
    - One color for correctly collected.
    - Another for incorrectly collected.

Genre

Platform games, or **platformers**, are characterised by the player controlled character's ability and necessity to jump or climb onto floating platforms and subsequently along them to achieve some goal. Often enemies or obstacles have to be avoided, for example evil mushrooms or lava, respectively.
A **puzzle-platformer** is a subgenre of platformers, where the goal of the game is achieved by solving a kind of puzzle, often via some gimmick.

While FBR currently doesn't have any platforms for the player to jump on, it is part of the items listed in the Wish List and platforms exist in the ShapeArea. In FBR the gimmick is both the matching of shapes and boxes along with the ability to change the shapes using the SwapButton.

Bibliography

- **Audio:**
    - I created the background music and the sound effects using Ableton Live.
- **Graphics:**
    - I created all graphics using Krita.
- **Scripts**:
    - My S*ingleton* implementation was heavily inspired by the following: http://wiki.unity3d.com/index.php?title=Singleton&oldid=20231
    - My *CharacterController* was inspired by the following: https://github.com/Brackeys/2D-Character-Controller/blob/master/CharacterController2D.cs