



Chronix Pulse Mobile Project #1

Title of Game: GemFall Catcher



Version: 1.0.0

Last Update: 06.08.2024

Table of contents

Overview	3
Story and Narrative	3
Gameplay Mechanics	4
User Interface (UI) and User Experience (UX)	4
Game Art and Visuals	5
Audio Design	5
Technical Details	6
Monetization	6
Team and Timeline	7
Legal and Compliance	7
Conclusion	8

◆ Overview

◆ Brief summary of the game

A dexterity arcade mobile game where the player controls a small character at the bottom of the screen. The screen fills up with different coloured shapes that spawn on random and falling from the top of the screen. As the game progresses, the falling shapes get faster and more complex, making it more challenging for the player to catch the right shapes.

◆ Inspired by

Wolf & Eggs! (a.k.a “Nu, pogodi!”)
Tetris

◆ Target audience

A wide range of mobile users with no particular age range.

◆ Platform(s)

Primary: Android
Secondary: iOS

◆ Story and Narrative

◆ Premise and setting

Fantasy setting - gems and power up runes are magical entities. Nature theme - the background environment, visuals and particles, main character and user interface should resemble and include elements from nature like trees, water, grass, and sky.

Narrative: Woodland Hermit that lives peacefully but then gems start falling from the sky

◆ Main character (Mascot)

Main character helps players navigate through the tutorial
Example: Tutorial from Project Zomboid where Spiffo shows you around the user interface and core gameplay loop.

◆ **Gameplay Mechanics**

◆ **Objective of the game**

The player's objective is to catch only the shapes that match the colour of the character by moving it left and right. If the player catches a shape that doesn't match, they lose a life. The main game mode is endless, and the player's score is recorded and displayed at the end on a leaderboard to compare with other players.

Future update?: Level mode has a goal objective to collect specific shapes shown on the screen.

◆ **Core gameplay loop**

1. The needed shape and colour of the gem is shown at the top of the screen
2. Various coloured shapes spawn and slowly fall from the sky (top of the screen)
3. Player moves the character around to avoid wrong gems
4. Collecting correct gems increases the score
5. Repeat
6. One life is subtracted from total of 3 when a wrong one is collected
7. When life counter hits 0 the game is over

◆ **Controls**

One input - dragging the character. Drag: Dragging involves pressing and holding on the screen while moving a finger or stylus.

◆ **Progression**

In endless mode the speed of gems falling progresses after a score milestone is reached

Future update?: When player completes the set goal of coloured shapes, they progress to the next level

◆ **Player customization options**

None yet, but simple character and map environment skin changes are in mind

◆ **Win/Loss conditions**

When a player collects a wrong coloured shape and has no lives left the game is over

♦ Difficulty curve

Endless mode increases difficulty over time. Speed of falling coloured shapes, objective rotation on the character itself and number of obstacles / deboosters

Future update?:-The difficulty of levels increases progressively by growth of the goal objective. It includes collecting rare probability coloured shapes, a larger number of them or specific ones like finding correct letters for a word.

♦ User Interface (UI) and User Experience (UX)

Moodboard: https://miro.com/app/board/uXjVMl6szpo=

Wireframe: <https://www.figma.com/design/untuDfJnaShroS6UmPBwRw/Catcher-UI?node-id=0-1&t=MgZ20cki34qPYMS5-0>

♦ Menu screens

Main Menu: Slightly animated and volumetric art scene with a main character sitting at a campsite and fishing at a pond or a running river (Reference to the game environment)

Style: 2D art/pixel art or 3D animated scene

Example: [Main Menu - dark-souls-bonfire.gif](#)
[pixel art fishing](#)
[a cat fishing :\) pixel artwork by me : r/drawing \(reddit.com\)](#)

Settings: Includes buttons and sliders

Example [game-ui-vector-set-leaderboard-260nw-2246728681.jpg \(462×280\)](#)
[\(shutterstock.com\)](#)

Leaderboard: Has pictures, rectangular frame with name and score, position number

Example [1000_F_568727881_0Xn2fWEIU63pJbvadVfzNnCh92Ao2nCc.jpg \(667×1000\)](#)
[\(ftcdn.net\)](#)
[pngtree-game-leaderboard-ui-mobile-app-design-interface-png-image_2995946.png \(640×640\)](#)

♦ In-game HUD (Heads-Up Display)

Live Points — Heart icon
Score points — Gems or coins icon
Pause Button — Visible and non-intrusive UI icon
Gem objective — Main UI screen with a border to catch some attention

Primary Icon Button: 60x60 dp with 16 dp padding
Secondary Icon Button: 48x48 dp with 12 dp padding
In-Game Icon Button: 40x40 dp with 8 dp padding

(Just for guidance, for now one general icon size is fine)

♦ Icons and buttons

Main Menu Button - Will be reused and recoloured (Rectangular)
No ads badge - TV/portal saying “no ads” with a cancel sign on top (Squared)
Promotion badge - A treasure chest with coins a small “play” icon (Squared)
Coin Shop - Market stand with goods (Squared)

Button Height: 56 dp
Button Width: 240 dp
Padding: 12 dp on all sides
Margin Between Buttons: 16 dp
Text Size: 18 sp

(Just for guidance, for now one general button size is fine)

♦ Tutorial and help screens

Simple tutorial of 5-6 steps that shows main gameplay loop to the player
Game slows stops when a mechanic is introduced text bubble with character appears
A hand or a highlight appears to help navigate
Main steps: Movement, shape spawn, wrong shapes, collection, level objectives, booster

♦ Sound and music

Intense, magic and fantasy-like music to promote “danger” in the arcade style game

♦ Visual effects

Minimum number of VFXs: collecting runes, booster and dying

◆ **Game Art and Visuals**

◆ **Art style and direction**

Stylized and cartoony graphics.

◆ **Character design**

Animal (Fox/Cat/Squirrel) in a robe with a bycoket (or a wizard hat or a hood)

◆ **Environment design**

Parallax effect could be present slightly when moving left and right

◆ **Animations & Particles**

Running animation

Losing life animation

Collecting a correct gem (non-intrusive for gameplay)

◆ **Audio Design**

◆ **Sound effects**

Collecting rune

Switch of the colour/shape objective to notify user clearly

Button pressing

Booster usage

Taking damage

Coins?

◆ **Music**

Idle, calm music in the main menu

Fast paced action music during the gameplay

◆ Technical Details

◆ Game engine and programming language

Unity as a

◆ Platform-specific requirements

- Developer accounts (Google Developers and Apple Connect)
- App Annie/Sensor Tower/Adjust/IronSource (CPI)
- Unity ads/Google ads
- Amplitude Analytics/GameAnalytics/Appmetrica
- Firebase
- Applovin
- PlayFab

◆ Performance targets

The game must be able to run on old or less powerful Android devices, as well as all Apple products with different iOS after 14.3. It also must not be battery demanding.

◆ Monetization

◆ Business model (f2p, premium, in-apps, ads, etc.)

Game is free-to-play and does not rely on heavy ads or in-app purchases. A few places for advertisements will be allocated and it will have in-app purchases for consumable items such as premium currency or boosters.

◆ Virtual goods and currency

Gold is earned for free by playing, used for refilling lives or customization unlocks
Premium currency (like gems) is found in some shapes, achievements, bought with RLC

◆ Pricing strategy

Simple pricing strategy of 6 in-app purchase tiers going from 0.99 to 99.99

◆ Marketing and promotion plans

Promote the game mainly through word of mouth and different social media channels

◆ Team and Timeline

◆ Roles and responsibilities

Ivan Khlyzov - Producer, UX & Game Designer
Sam Ravnushkin - Front & Back-end Developer
Shang - User Interface Artist

◆ Development schedule

Begin in August - September
Develop core mechanics and gameplay loop by mid-end of September
Have a first playable prototype and start fixing bugs by mid October
Do the polishing and art wrapper by end of October

◆ Milestones and deliverables

September, 2 weeks - planning and final design phase
September-October, 4 weeks - prototyping and early development
October-November, 8 weeks - full development and testing phase
November, 2 weeks - launch and post-launch support phase

◆ Testing and quality assurance plan

Game must be tested on each stage. If an added part of the game or a mechanic has a bug it must be fixed first. For quality assurance a separate game or company email must be used.

◆ Legal and Compliance

◆ Intellectual property rights

- Cookies Policy
 - Return Policy
 - Disclaimer
 - EULA
- [Legal Requirements for Mobile Games - TermsFeed](#)

◆ Licensing agreements

- Privacy Policy
- Terms & Conditions

- ♦ **Privacy policy and data protection**

-

- ♦ **Content rating and classification**

-

- ♦ **Accessibility standards**

-

- ♦ **Conclusion**

- ♦ **Summary of the game design document**

This document mentions the full potential of the project. However, the priority of the development process is to deliver a playable prototype that can be released with a minimum outcome. That is the core mechanics and the main gameplay loop, excluding most of the back-end engineering, monetisation methods and artistic wrapping

- ♦ **Goals and objectives**

The goal of this project is to learn the essentials of game development, game industry business and back-end engineering through mobile games in order to get a better understanding on how large online games work.

The main objective of this project is to release a final game and list it in a portfolio, potentially turning it into a small source of passive income

- ♦ **Expected outcomes and metrics**

By following agile metrics it is expected to reach certain outcomes. These include learning and building things that will be used in the future projects.

- ♦ **Next steps for development**

Next step is to move to a slightly more difficult mobile project, before moving to the PC or console type of games.