**Blackout Bingo**

## Overview

# Concept

An online competitive bingo which adds **reflexes** and **planning** as the keys to winning.

# Features

As in bingo, each player is handled a board of **5X5** filled with numbers within a certain range **Y** of integers. After every **X** number of seconds, a random new number from the range is pulled out and displayed. After a certain **T** time passes, the game ends.

Each player is able to select any of the numbers he has on the board at any time. If the number is matching the last number pulled from the **Y** range, he scores an amount of points based on the time it took him to do so (faster is better). In addition, a streak if correct clicks help fill a meter which allows some special powerups, who would help the player score higher. Scores are also scored by having a diagonal, horizontal or vertical full rows.

# Environments and Tools

* Unity based game.
* Github (Git) extension for version control.
* PUN2 (Photon) for online functionality.
* TMP (Text Mesh Pro) for better looking UI elements (official Unity package).

# Test Plan

Checked locally and offline the core game mechanics.

Check with a server plus one player.

Check with a server plus two simultaneous players.

Out of Scope -   
Server / Player Crush / Log off.  
Expected result – should not crush the game for the other players. At most it could be a momentary delay in the gameplay.

## Gameplay

# Initial

# Scoring

Points are earned in two ways:

* Clicking on rightfully pulled balls scores points, according to the next (generic) ledger:
  + Very Fast (less than 0.5 second) = 5\*scoreValue, where scoreValue=100
  + Fast (less than 1 second) = 3\*scoreValue
  + Ok (less than 2 seconds) = 2\*scoreValue
  + Slow (less than 3 seconds) = 1\*scoreValue.
  + Any slower than that would still mark the tile, but would not add additional points.
* Scoring a bingo (a full row of rightfully clicked tiles, horizontally, vertically or diagonally). All new bingos are calculated only after pressing the bingo button, and add points according to the next formula (if the number of bingos is 1 and above):
  + (2^(number of bingos + 1))\* bingoValue, where bingoValue=1000

For example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of new Bingos | 0 | 1 | 2 | 3 | 4 |
| Points Added | 0 | 4000 | 8000 | 16000 | 32000 |

## Art

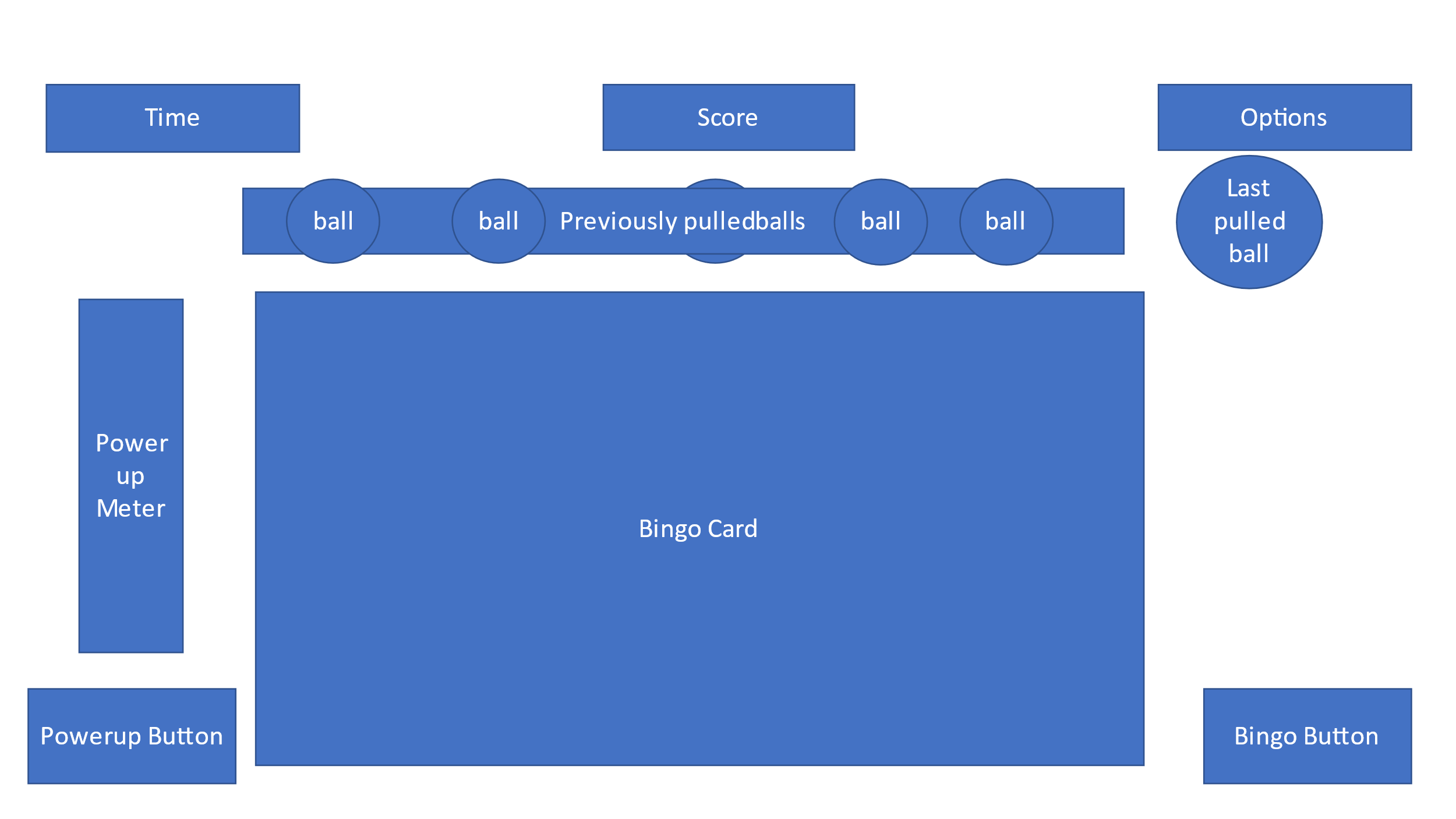
# Art Style

As in the original game, I used the combination of light cyan and light magenta, which are calming colors.

# UI

Most of the UI is created on runtime, which allows the most dynamic game changes. Could be done in advance to look nicer.  
The position of the buttons could ease transferring into mobile, as the majority of the action is in the bottom center of the screen, with the two buttons being on opposite sides fitting two thumbs.

# Layout



# Out of Scope

Security/anti cheaters measures.

Offline / connectivity issues handling

Limited to 20 users at most on the application (free Photon limitation)  
  
The center number should be free from the beginning, it is right now a standard bingo tile.

# Actions

Pregame Server –

Initiate the game when enough players have joined

Game manager / Server –  
Generate random game boards

Pull a new number and display it  
Display scores by the end of the game

Pregame player –

Allow name change

Join game

Player –

Click to earn points

Click on possible bingo numbers

Declare a Bingo

Fill a progress bar

Use a powerup  
Announce a game over if all numbers were picked up, or if local time ran out  
  
Bingo Number –

Be clicked – if is valid score points, fill meter etc.

Indicate status – not used, clicked correctly, clicked incorrectly (and return to not used)  
  
Timer -  
Show the current player’s time left for the game  
  
Score Display –

Format points and display them  
  
Ball –

Holds value and time pulled

# Screens

Server Main menu

Player Main menu

Player in game board

Player special power up screen

End game scores

# Controls

## Mouse based only. Audio

# Sound Effects

Asda

# Music

Asdasd

## Gameplay Diagram

## WorkPlan

# Basic Player interaction – Have a timer with a popup. A score is based on the speed of the click. (Limited UI)

Timer -  
Show the current player’s time left for the game

Score Display –

Format points and display them

Ball –

Holds value and time pulled

Player –

Click to earn points

# Bingo board creator, with pulling new numbers and displaying them

Player in game board

Game manager / Server –  
Generate random game boards

Pull a new number and display it

# Combining the two and scoring only when the desired number is clicked.

End game scores

Bingo Number –

Indicate status – not used, clicked correctly, clicked incorrectly (and return to not used)

Player –

Click on possible bingo numbers

Announce a game over if all numbers were picked up, or if local time ran out

Fill a progress bar

Bingo Number –

Be clicked

Game manager / Server –  
Display scores by the end of the game

# Adding Bingo declaration functionality.

Player –

Declare a Bingo  
Add Visual Cues on Bingos

# Separation into Server and Clients

Server Main menu

Player Main menu

Pregame Server –

Initiate the game when enough players have joined

Pregame player –

Allow name change

Join game

# Allow score progression

# Add Power ups

Player special power up screen

Player –

Meter and filling it

Use a powerup

Improve final look, including ui and audio.  
Added audio effects and background music