How King uses AI to test Candy Crush Saga

Alexander Andelkovic

King

- Founded 2003
- +200 Games
- 2000 employees
- 11 Locations (Stockholm, London, Barcelona, Berlin, Malmö, Bucharest, San Francisco, Chicago, Los Angeles, New York, Malta)
- 258 million monthly active users (Q2 2019)
- Acquired by Activision/Blizzard (2016-02-23)

Major Franchises

Candy Crush

Pet Rescue

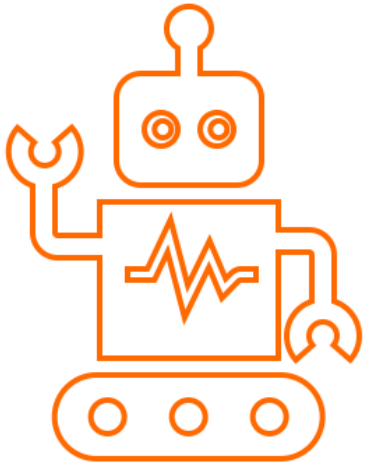
Bubble Witch 3

Farm Heroes

King QA roles

- ATL – Agile Testing Lead
- QA Analyst – Exploratory tester
- QRT – Quick Regression Team
- TAE – Test Automation Engineer
- DS – Data Scientist
- Developer – (Frontend/Backend)

BAIT – Bot for AI-based Testing



BAIT is a test automation bot powered by AI

How does BAIT work?

Bait explores game by

- Taking screenshots
- Uses AI to find button coordinates
- Click detected buttons
- Performs AI powered verifications

BAIT workflow

Different Tests Capabilities

Automato
(Automation framework)

Automato takes screenshot

1

Compare with former screens (hash similarity + text similarity)

2

BAIT brain

Updates Game graph

3

AI detects all the buttons in the screenshot

4

Choose a button and send the coordinates

Click the coordinates

5

Button detection – AI model training

Data labeling

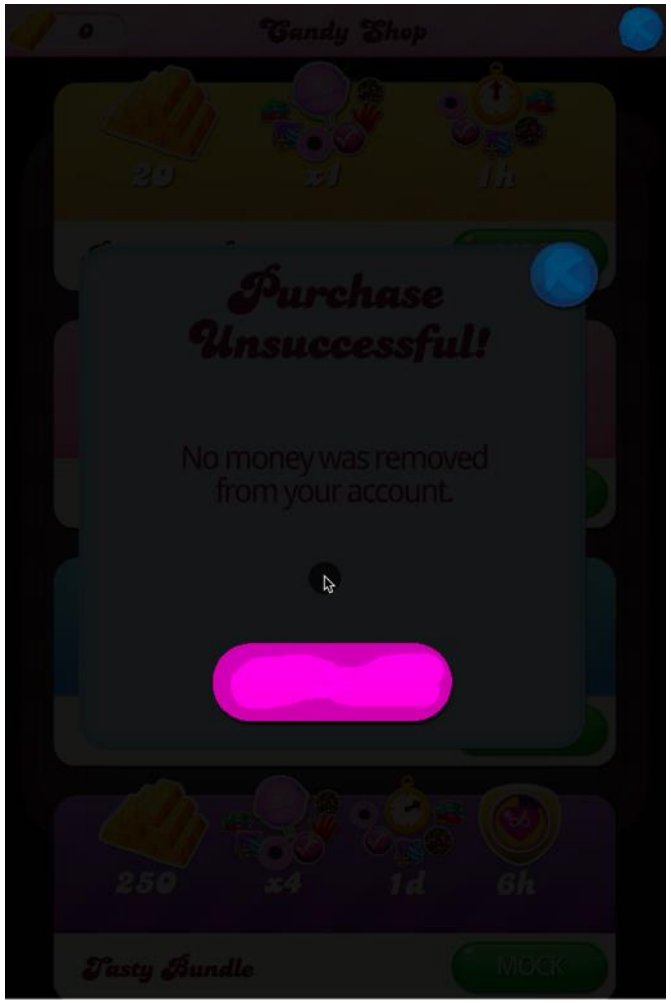
Data augmentation

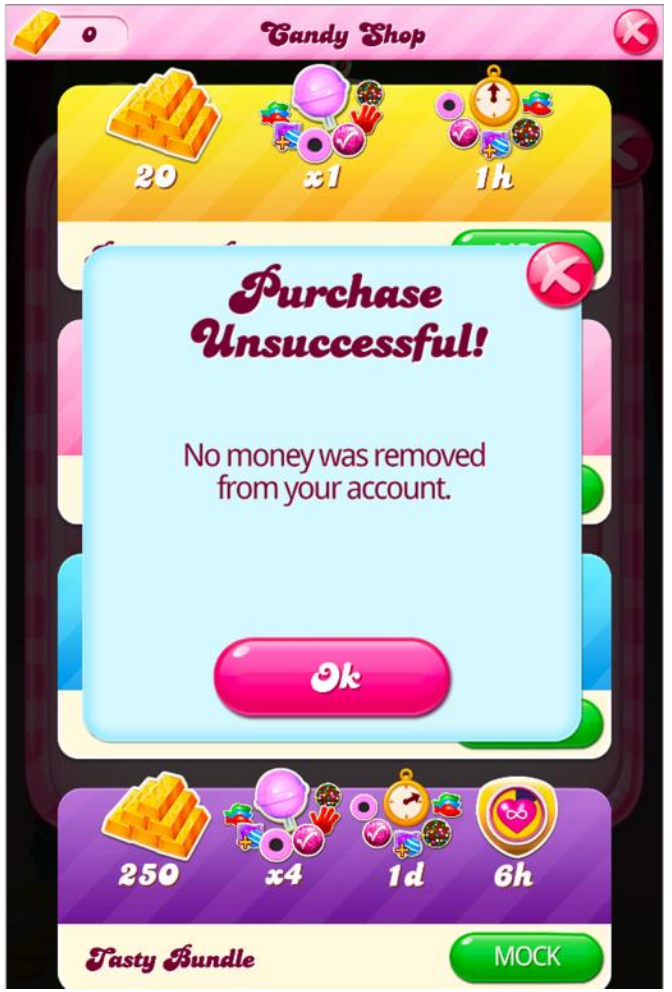
DNN training

Button detection – Mask R-CNN

Mask models

- Close button
- Other button



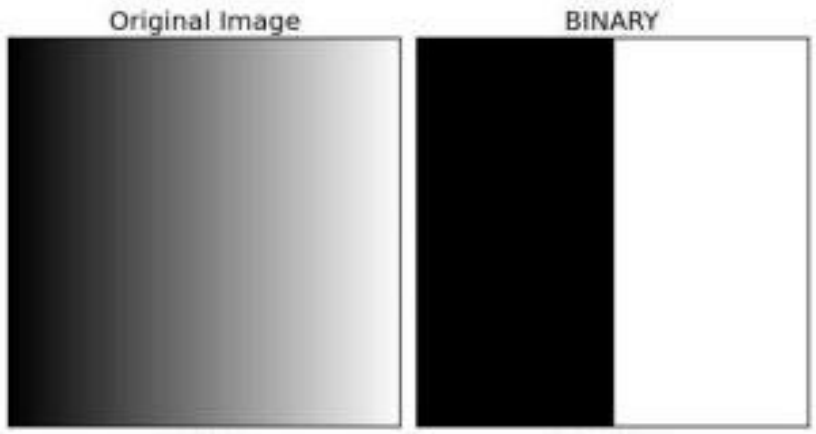


Instance Segmentation

Button detection – image segmentation and thresholding

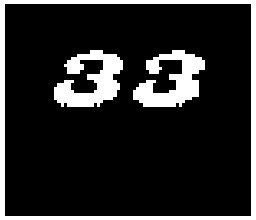
Color shareholding

- Associate each pixel with black or white according to a threshold(rgb value)



Otsu's Binarization

- Filter to remove noise in an image



Button detection - Result

Similarity detection – Image hash similarity

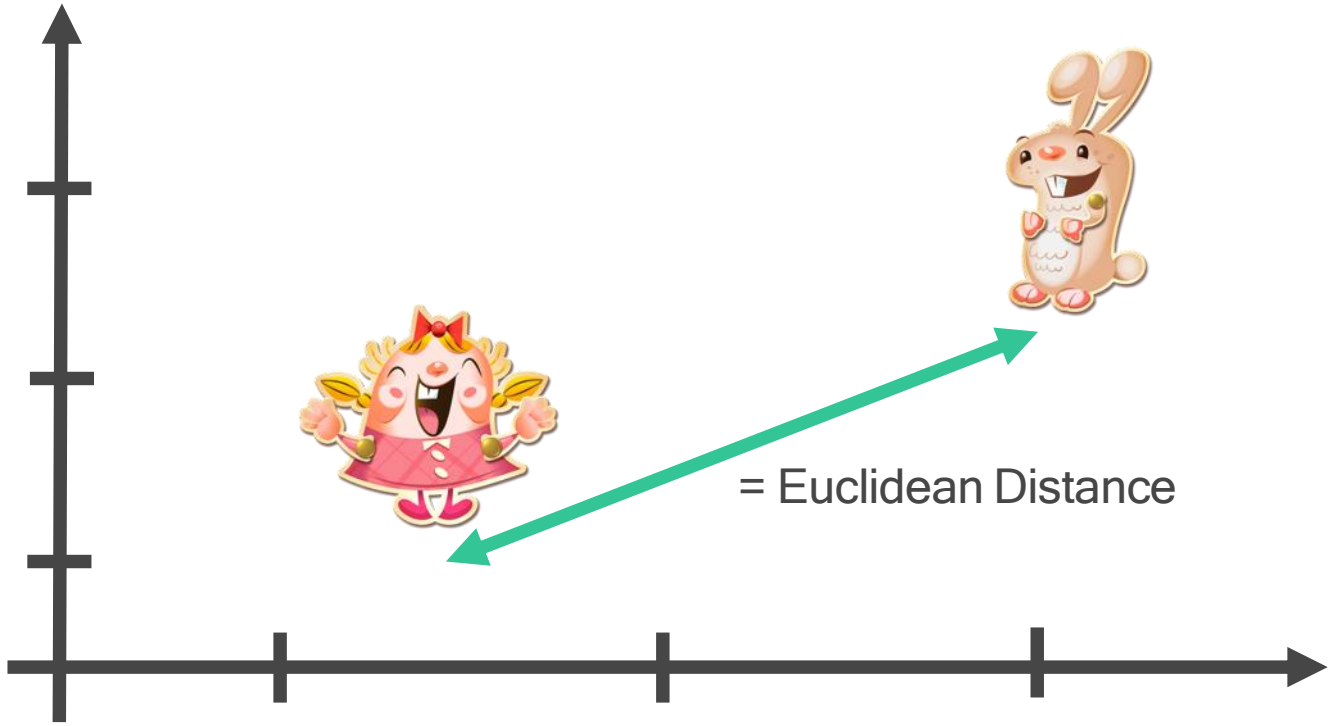
1. Generate image fingerprint with perceptual hash algorithm (p-hash)

P-hash( ) = 8a0303f6df3ec8cd

P-hash( ) = 8c011456ef9eadcc

2. **Euclidean distance algorithm** used to compute the similarity(distance) of the two hash values.

Similarity detection – Euclidean Distance



$$\sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

3. Similarity range 0 = Similar, <6 minor difference, >6 different

4.Compare two images fingerprint, if they are similar by a threshold(<6) then considered same image

Similarity detection – Text similarity

1. Screen parsed by Google vision API for text

2. **tfidf vectorizer(word count, !word frequency)**used to generate matrix of text input

Text from image 1: “Buy the Gold Bars and give the Piggy Bank”

Text from image 2 : “Piggy Bank!”

Text mining results(tfidf matrixes):

TextTerm	Buy	Gold	Bars	give	Piggy	Bank
Text 1	0.3160	0.2248	0.3160	0.3160	0.2248	0.6320
Text2	0	0	0	0	0.7071	0.7071

Cosine similarity = 0.3

Similarity detection – Cosine Similarity

3.**Cosine similarity algorithm** used to calculate distance(0-1) between two vectors and in our case if distance >0.6 text considered similar

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Similarity detection – Hash & Text similarity

1	2	3
		
Image Hash similar = similar	Image Hash not similar but text similar = similar	Image Hash not similar and text not similar = not similar

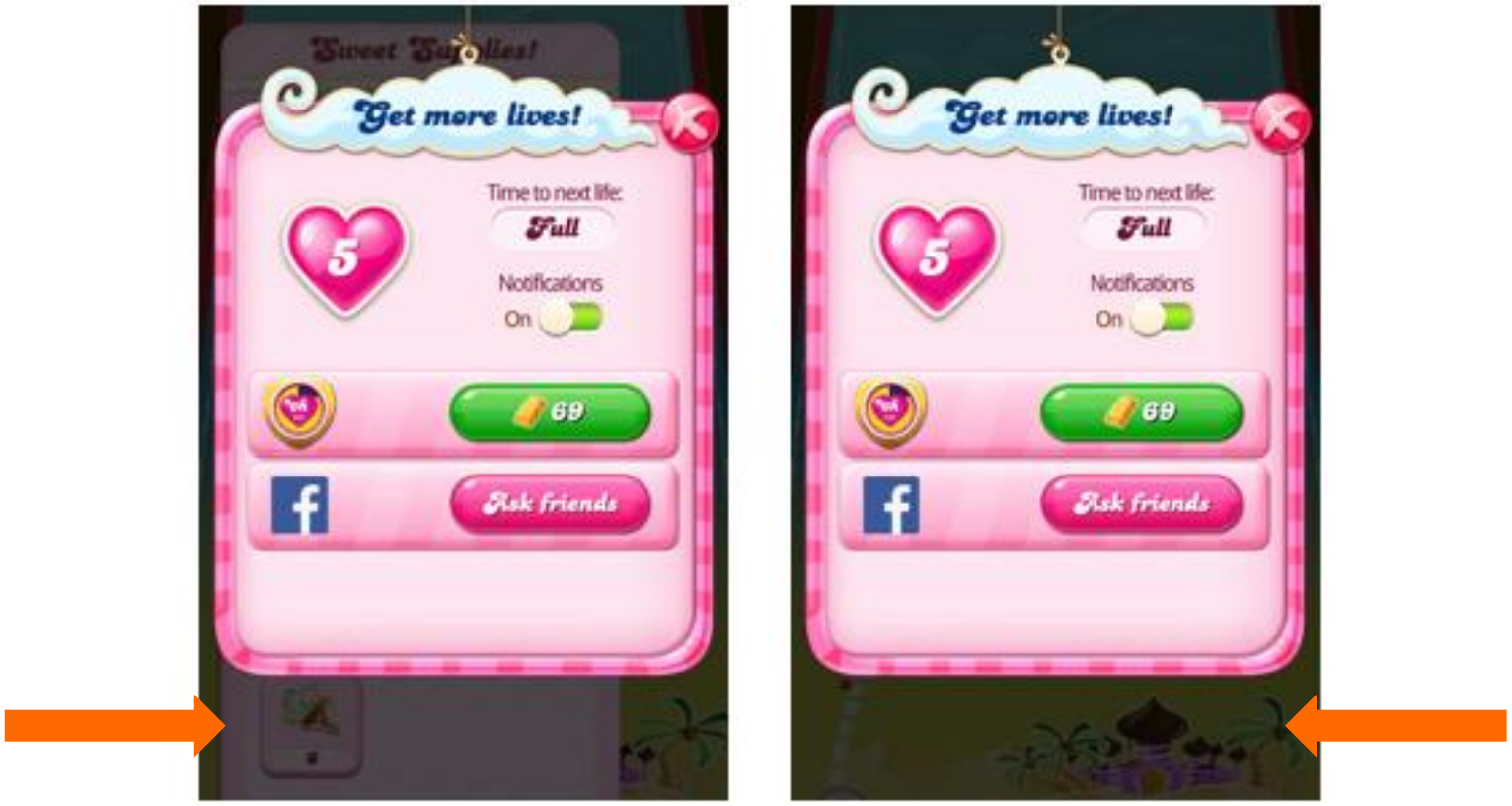
Similarity detection results

Moving features(animation glitches)

Similarity detection results

Miss-aligned features(bug)

Similarity detection results

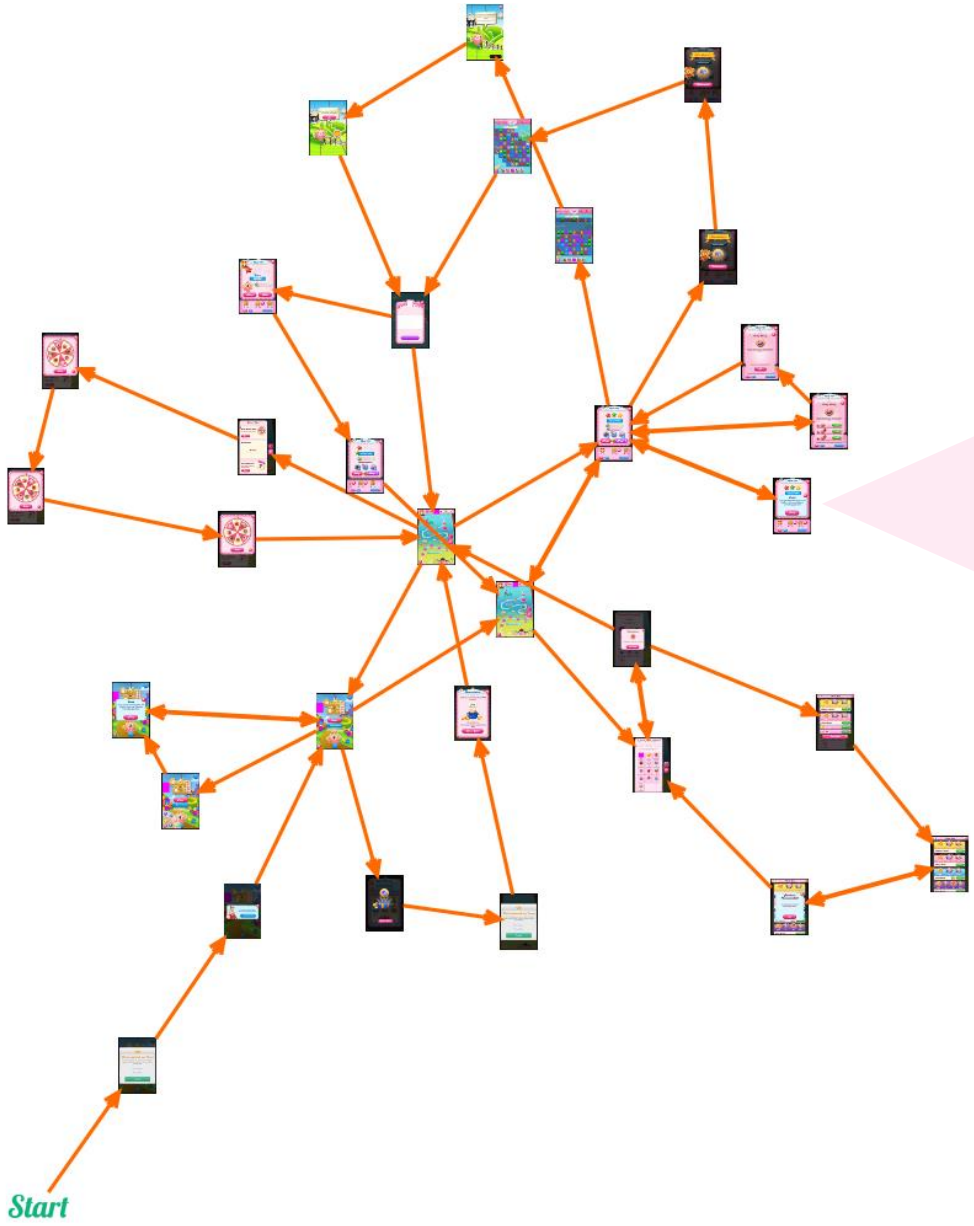


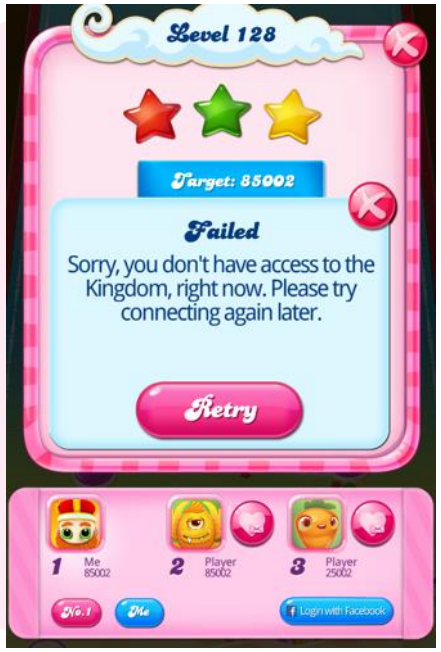
Same screen, different backgrounds(threshold)

Similarity detection results

Overlapping menus(new state)

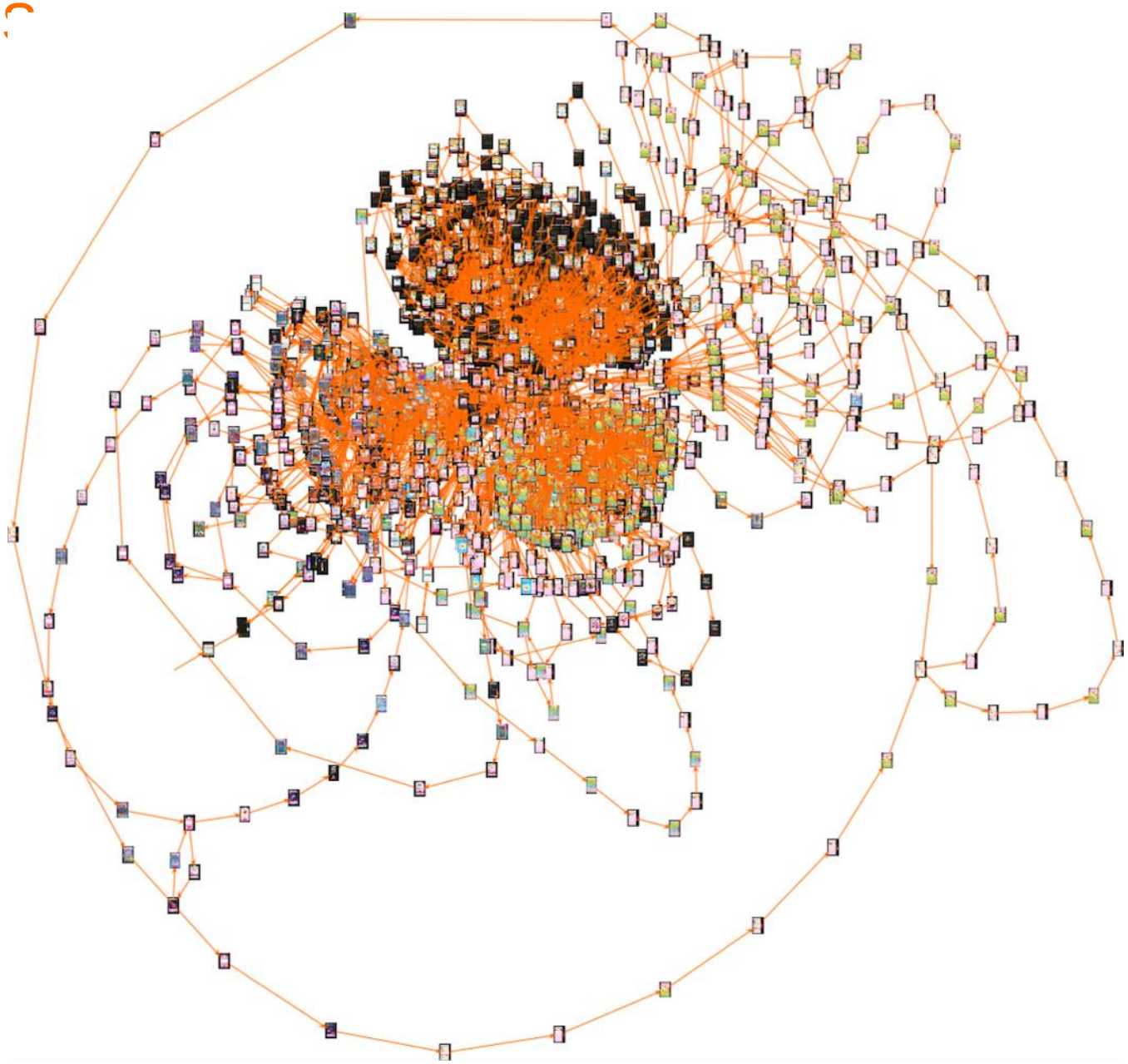
Game Graph





Cloud Vision API + OpenCV

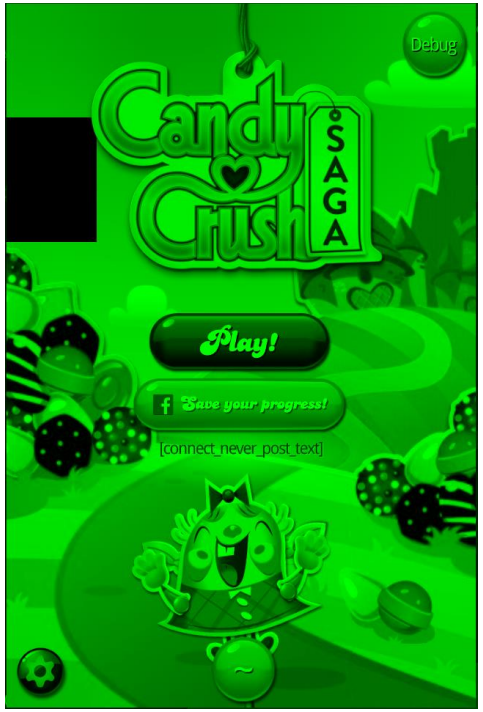
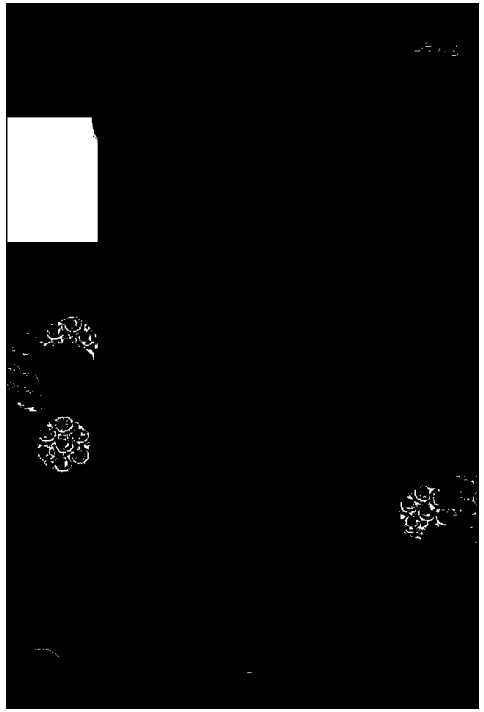
Game Graph +4500 levels



BAIT - Game console commands

- Win/Loose <level>
- Progress <level>
- Unlock <feature>
- Access <facebook_user_token>

BAIT – Missing texture detection

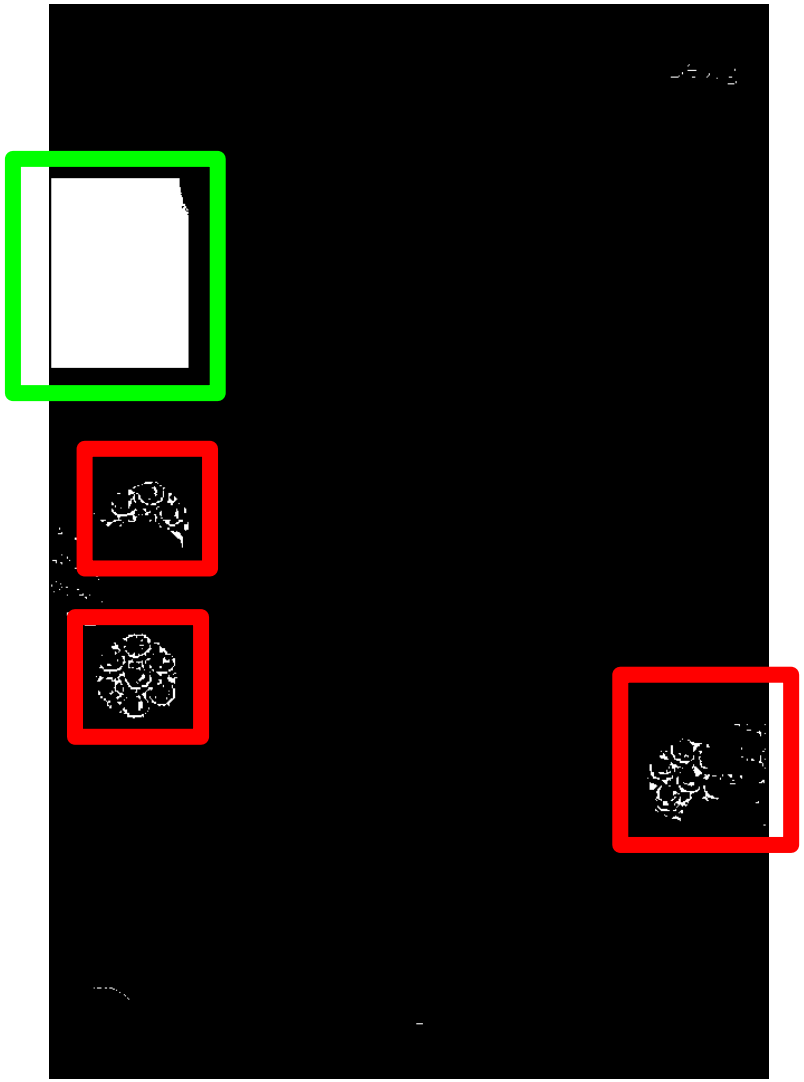
1	2	3
		
Convert image RGB to only keep Green channel	Thresholding - Pixels close to black set to white rest set to black	Contour finding algorithm used to find contours

BAIT – Missing texture detection

OpenCV functions used

1. findContours() used to find all contours

2. contourArea() used to calculate contour area and if below certain threshold it's filtered out

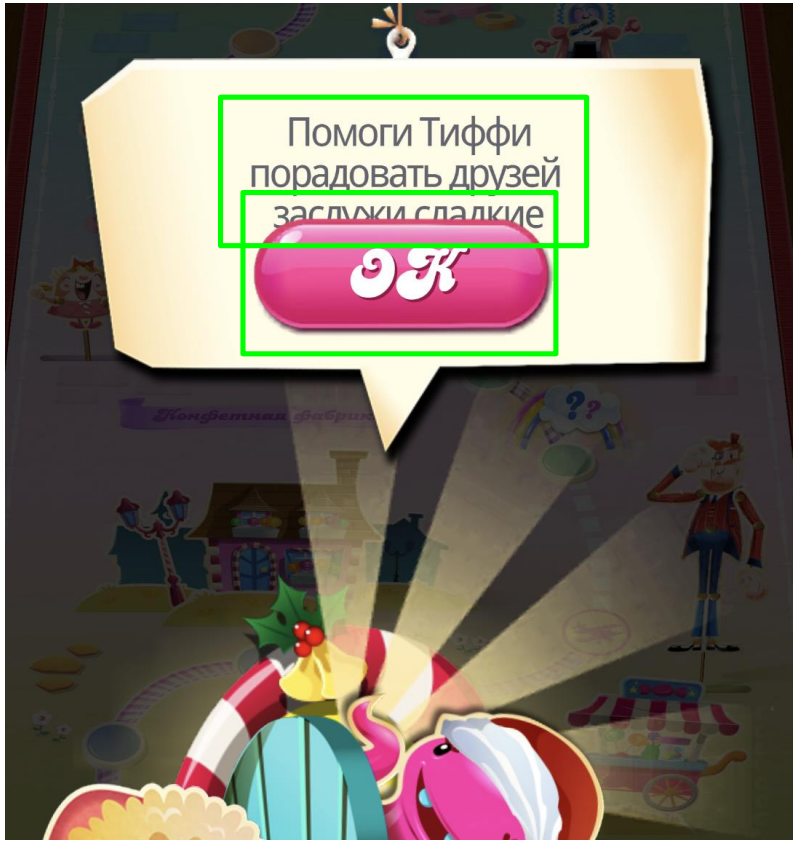


Example - Missing textures

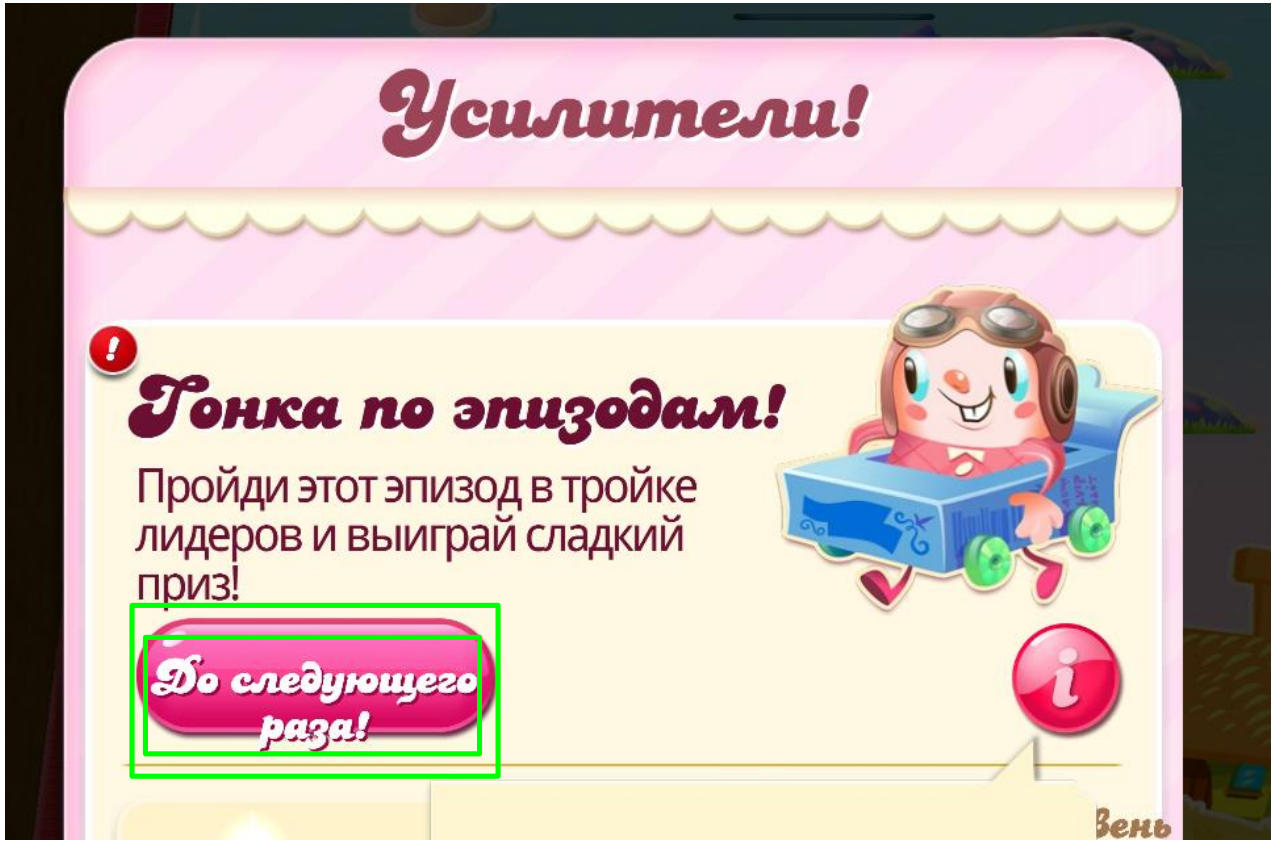
Example - Missing texts

Cloud Vision API

Overlapping objects



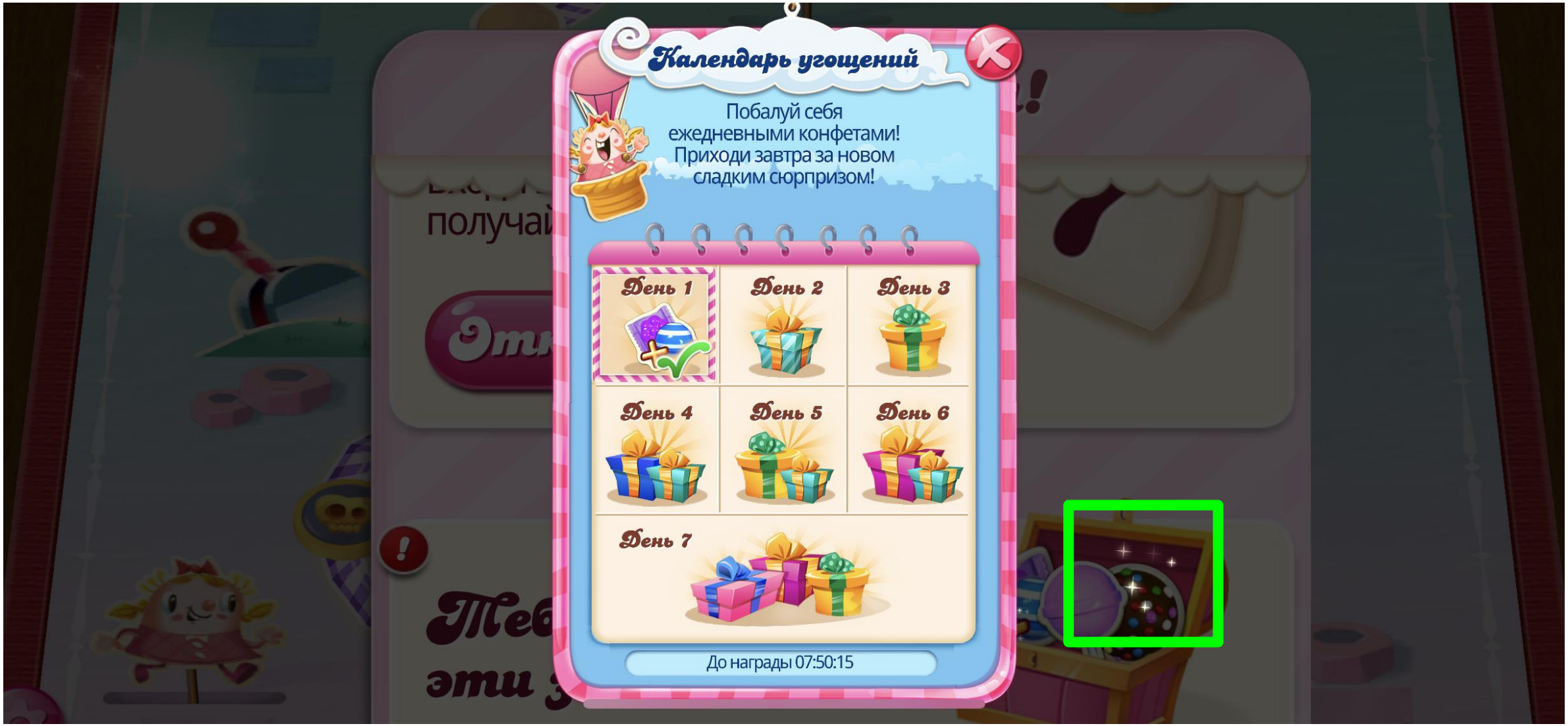
Text covered by buttons



Text does not fit button

BAIT Quiz – Can you find the bug?

Game Rendering bug



Game Rendering bug – thresholding used for detection

BAIT Requirements

Desktop

GET automation/screenshot/<filename>

GET automation/tap/<x>/<y>

GET automation/level/win

GET automation/fbuser

Mobile

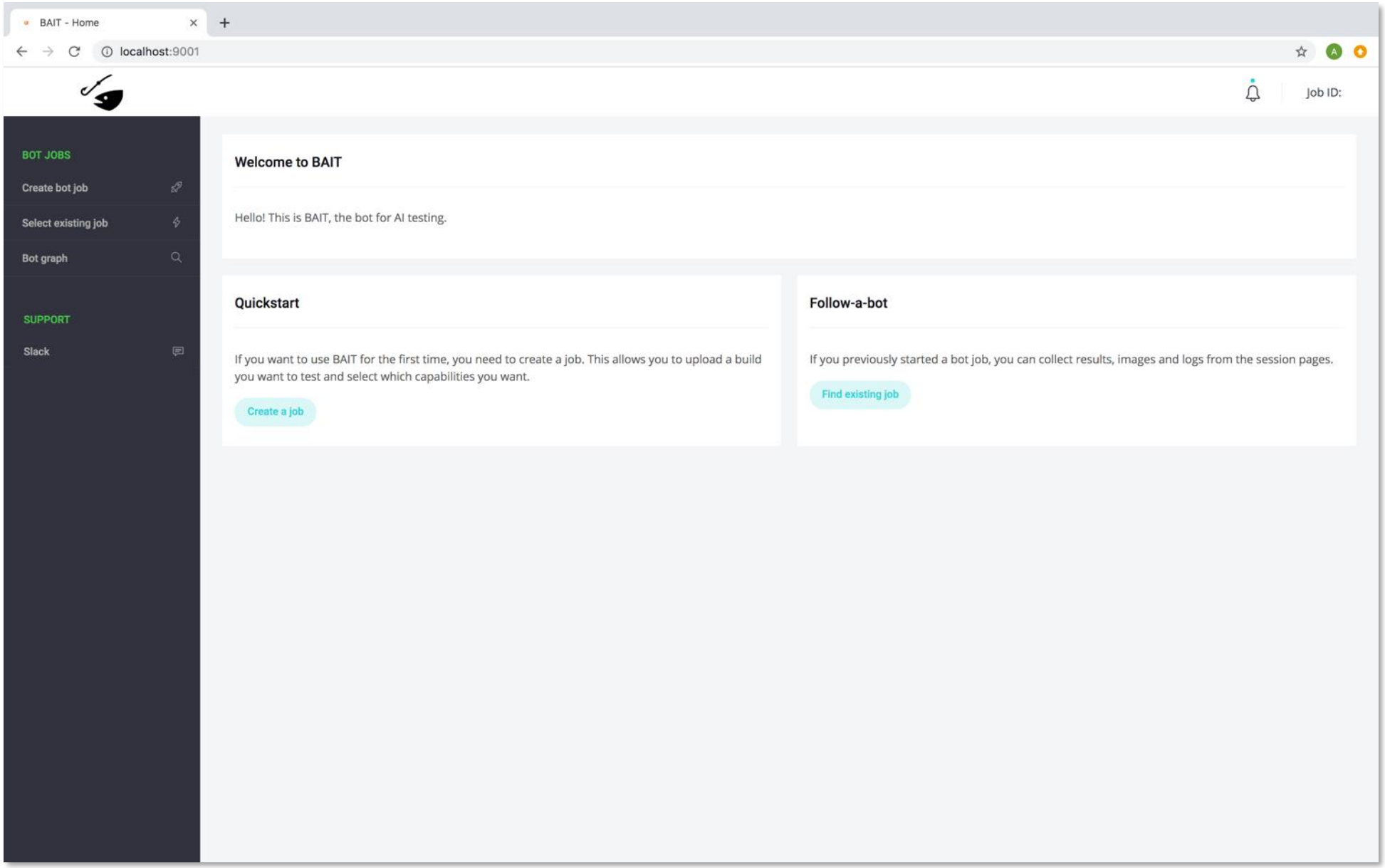
GET automation/level/win

GET automation/fbuser

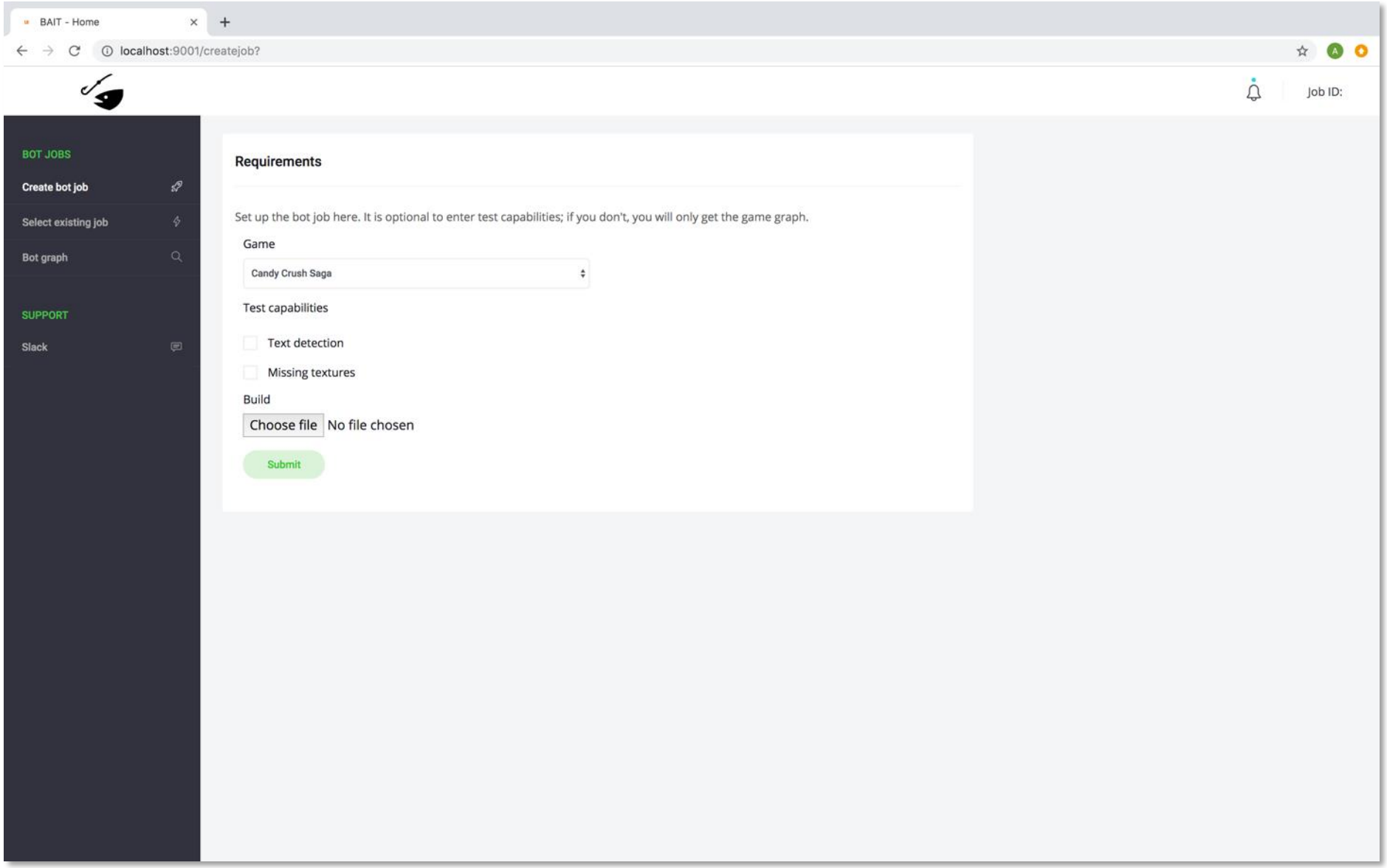
Click BAIT - Overview

- BAIT frontend
- Run scheduled BAIT jobs
- BAIT statistics

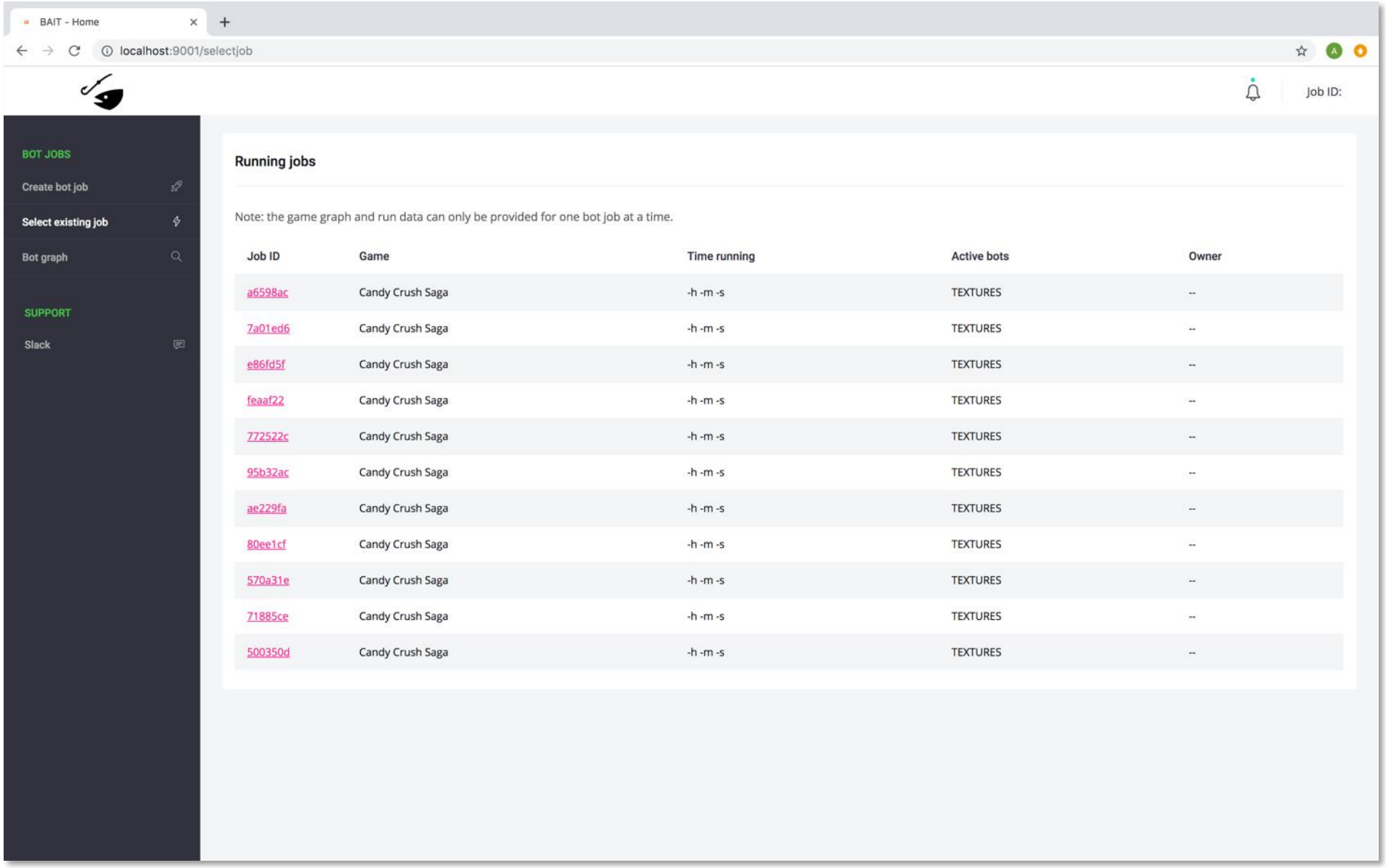
Click BAIT - Dashboard



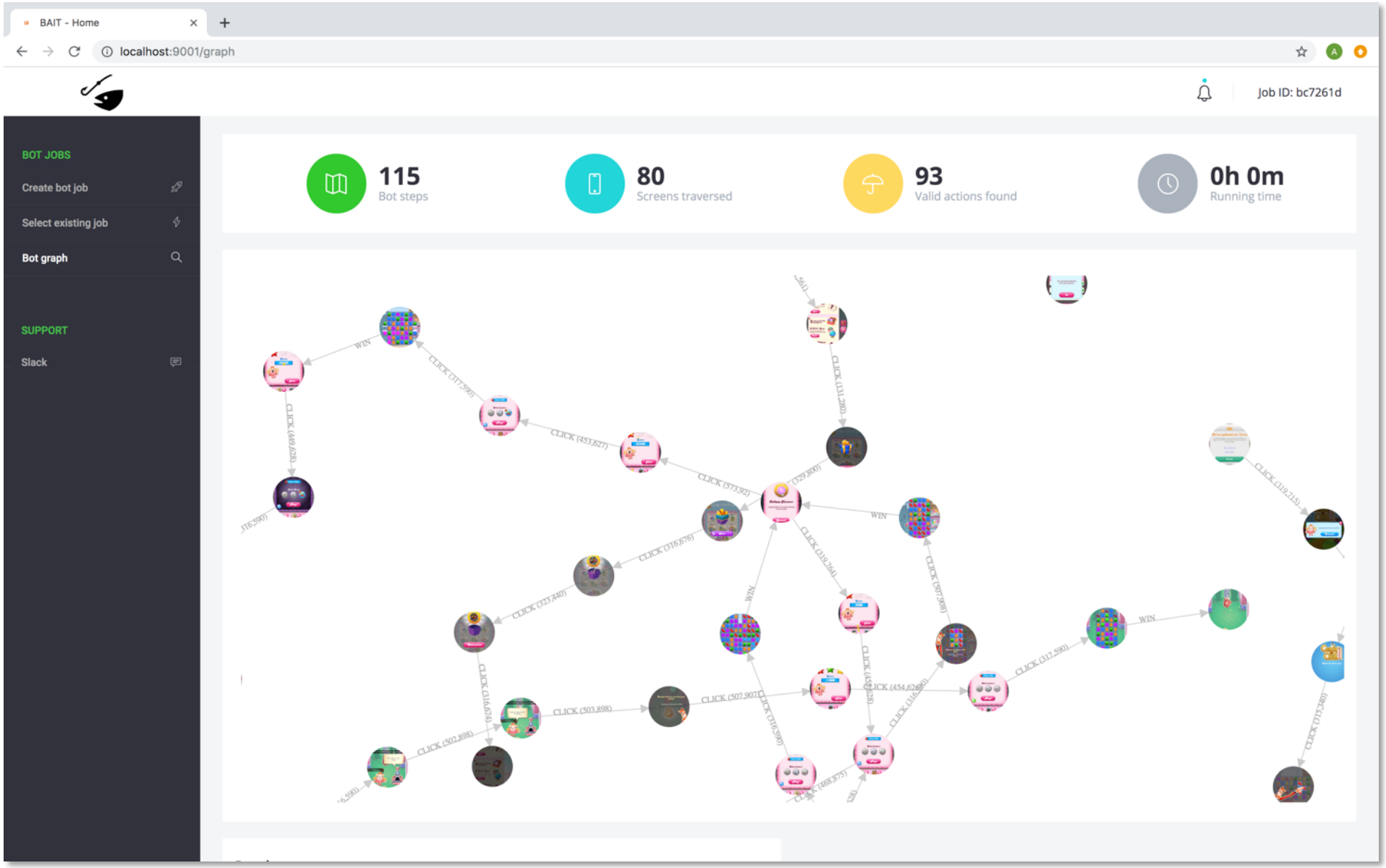
Click BAIT - Create BAIT job



Click BAIT – Bait jobs scheduler



Click BAIT - Statistics



BAIT Demo

Summary - What can BAIT help out with?

- Crashes
- Text bugs
- Missing textures
- Audio errors

Questions?

alexander.andelkovic@king.com

king.com/jobs

Thank you! :)