







**Björn Ritzl**  
Principal Engineer  
Product Owner  
[bjorn.ritzl@king.com](mailto:bjorn.ritzl@king.com)



**Mathias Westerdahl**  
Principal Engineer  
[mathias.westerdahl@king.com](mailto:mathias.westerdahl@king.com)

Kung

200

King



The image features a vibrant, colorful background filled with various types of candy. There are round candies with horizontal stripes in orange and white, blue and white, and red and white. There are also solid-colored candies in shades of blue, yellow, orange, and red. Some candies are shaped like fish, including a large purple one and a smaller red one. A chocolate candy with colorful sprinkles is visible in the upper right. In the center, the 'Candy Crush' logo is prominently displayed in a golden, bubbly font. A small red heart is positioned between the words 'Candy' and 'Crush'. To the right of the main logo is a white rectangular tag with a red border and a small hole at the top, containing the word 'SAGA' in red capital letters. The entire scene is brightly lit, giving the candies a glossy, three-dimensional appearance.

# Candy Crush

S  
A  
G  
A

268000000



2

King



**DEFOLD™**





- C/C++ engine
- Lua for game logic
- 3D with 2D focus
- Component based
  - Physics, 2D sprites, Spine models, tilemaps, 3D models, physics, sound, scripts, etc
- Modular
  - Remove what you don't need
  - Extend with additional native code
- Small
  - Google Play Instant
  - Facebook Instant Games
  - Playable ads



The logo for GDC 2016 features the text "GDC" in a large, white, sans-serif font. Below it, the year "2016" is written in a smaller, white, sans-serif font. The background is a solid blue color with a subtle pattern of overlapping, semi-transparent geometric shapes, including triangles and polygons, in various shades of blue, creating a layered, architectural effect.

GDC

2016





# Blossom Blast SAGA

King

287



286



285



SAMSUNG

15

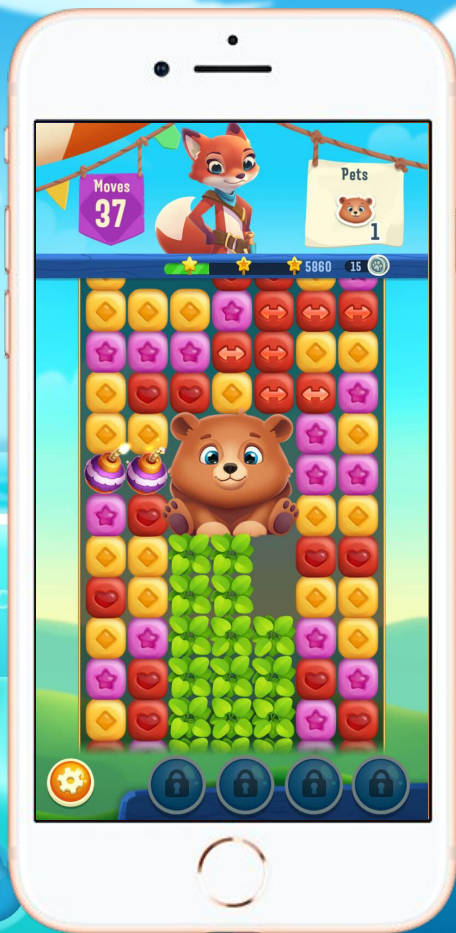
0

11





King



Pet  
Rescue  
PUZZLE











**Dominate The Enemy!**



# Debugging



# Debugging in a cross platform game engine

- Engine core in C/C++
- Platform specific code on top
  - JS for HTML5
  - Objective-C for iOS and macOS
  - Java for Android
  - More C/C++
- Ratio is somewhere around 20:1
- Good debugging tools is a must
- Usually enough to debug on host platform



# Debugging HTML5 in a cross platform game engine

- HTML5 builds using Emscripten
  - With support for WebAssembly
- Emscripten can generate source maps
- Browser dev tools support source maps
- Debugging can be done in the browser
- Even better if you can debug using a debugger of your choice
  - Remote debugging + IDE plugin
- But what about WebAssembly?



# Debugging graphics

- Inspect textures, shaders, draw calls and frame composition
- Open GL ES 2.0 and WebGL
- Desktop and mobile has RenderDoc, XCode, GAPID
  - Very powerful tools
- Debugging WebGL
  - Spector.js
  - Can the browser dev tools provide this?





