Just Keep Shipping
Thaddaeus Frogley, Lead Programmer
Boss Alien, Natural Motion, Zynga
Who Am I

- Thaddaeus Frogley, Lead Programmer, Boss Alien
- 20 Years Industry Experience
- Previously: JAGEX, Climax, Rockstar, KotJ, CyberLife, & Mythos Games.
- @codemonkey_uk
- http://thad.frogley.info/
I am here
I am here.

And here.
What Is CSR Racing?

- A game built in Unity3D
- Features and Content on iOS and Android
- Infrastructure and Cloud Services
- 60-40 Split
What Is CSR Racing?

- Soft Launch 28th May 2012
- Global iOS Launch July 2012
- Current version 2.7.1 on iOS, GooglePlay, Amazon
- 32 Client Updates on iOS, 31 Data Pushes
- 120 million installs
What Is CSR Racing?

- 221kloc C#
- 25kloc Objective C
- 34kloc Java
- 320kloc XML & JSON
Why it is difficult

- PS2/PSP Game: 470K LOC
- Xbox360 game: 850K LOC
- New-gen middleware: 1580K LOC
- "One full-time maintenance person for every 20K LOC" - Thomas Pigoski
What Is CSR Racing?
What Is CSR Racing?
What Do We Do

• Regular Content Updates
• Frequent Feature Updates
• Constant Improvement
• As-required SDK Updates
What Do We Do

• Metronomic releases
• Painless
• Platform Parity
Content

- New Cars, Races, Crews and Narrative.
- Regular Multiplayer Seasons are an on-going Player Facing commitment.
- Date-locked.
Improvements

• Smaller
• Faster
• Better
Features

• New feature ideas come in all the time.
• Features come in a variety of sizes.
• Features surprise and excite fans.
SDKs

- Unity3D
- NmgMetrics
- Fabric
- Flurry
- Chartboost
- TapJoy
- AdColony
- MoPub
- Apsalar
- Upsight
- Apple
- Google
- Amazon
- Twitter
- Facebook
It’s Not Straightforward

• Feature and Content Work Overlaps.

• Work for release N, N+1, N+2, and more, can all be happening concurrently.

• Features/content for release N might be moved to N+1 … or N-1!
Surprise!

• Platform updates break the live game!

• Players discover problems QA missed in the live game!

• There is an amazing $£¥ opportunity if we can make %CHANGE% to the game before %DATE%! 
It Never Stops!

- Everybody has ideas.
- The game is always in development.
- The world is changing around us.
Communication

“Having a viable, compelling, clear, and well-communicated shared vision was more important than any other factor we looked at.”  

Paul Tozour
Communication

- Open plan office
- Whole team stand up meeting 2x week
- Email email email
- Code review
- JIRA, Spreadsheets, Whiteboards
CSR Racing

2.7.1 WT: USA
- 23 Mar 2Km from
- 31 Mar launch

2.7.2 Tech update
- 10 Apr build 7
- 27 Apr launch
- Unit upgrade
- Car峰会
- Unreal, FB, 63 30
- Font tech update

2.7.3
- "Gurney"
- Pre-International DP
  - 13 May press
  - "Penna Ergo"

2.8.0 The International
- To be scoped
- Will likely need E3 support

Backlog: UI rewrite - on hold

CSR Classics

1.6.0 Season & Feature cars
- Jaguar XJ13
- Ford Taurus X9
- Alfa Romeo Spider 33
- Unity 4.6.1p upgrade

1.7.0 Alfa Romeo Mfr Feature
- Mini store
- Replay system update
- Bigger gas tech
- Season Prize cars
- Unnamed
- Langdon

1.5.0 Ferrari Mfr Feature
- Agent Drive
- Top line
- New build system
- 30 Jan 2K
- 12 Feb launch

1.6.0 FB SDK update
- Unity 4.6.1p update
- Bug fixes
- 17 Feb launch

1.7.0 Alfa Romeo Mfr Feature
- Bigger gas
- Replay system
- Long Plunge (20k)

9 April SWB
23 April Lawmm

Bentley GT3 02 1AP
Lambo & 17
- Getting to 64bit
- WP8.0 issues
- International testing
- Git branch merging & Miscs
cleanup
- 17

www.bossalien.com
Communication

- Feature kick-offs
- Feature Reviews, show-and-tell
- Post-mortems
Post-mortem: WT USA

IAP last minute changes
- Cut scope early
- Continue
- Overloading single guy
- Stop
- More design time to polish events
- Tools to aid basic checks - no fix time yet
- NMAG team direction - late bugs being reported
- New builds
- Continued issues
- Release on 'crushed pork' issues
- QA holdup on bug fixes

Merging data
- API merge issues
- Build issues
- Test issues
- Validation
- Fix program data merge
- Document data merge
- Content to jar
- Licensing & Appeals
- Early catchers to licensing - list on
- Patent capability - there is no more time
- Localization
- Early strings going into the game
Process and Change

- Deliberate
- Dynamic
It’s a Marathon not a Sprint

• Don’t crunch.

• Crunch is bad.

• Crunch has been unambiguously shown to be both bad for the product and bad for the team.

• Don’t crunch.
Support your bottle-necks

- Identify bottle necks
- Allocate code-resource to improving their lives
Branches

• 50+ branches
• git is amazing
• the CSR asset system is also pretty good
CSR Asset System

• Any runtime loaded asset (bundle) in CSR can be branched, versioned and updated

• dev / qa / prod

• Data can be live as soon as QA approves

• AB Testing puts users on named versions
CSR Asset System

- Merging is difficult and error prone
- Focus for improvement in New Game
Branches

- Localisation DB doesn’t support branching.
Estimation

- Estimate pessimistically
- Estimate concept to completion
Planning and Resources

• Planning to full capacity means you can’t react to opportunities.

• Planning to full capacity means you can’t absorb unexpected downtime.

• Planning to full capacity virtually guarantees missing your dates.
Planning and Resources

• Triage Early
• Triage Often
Planning and People

- Freedom to Innovate
- Space to Fail
- Ownership
Planning and People

- Know your team.
- Keep your team (happy).
- Trust your team.
Tech Debt Is Real

- Time constraints
- Knowledge constraints
Tech Debt Is Real

- Actively identify and address Tech-Debt.
- Pay the principle not just the interest.
- Not all Refactorings are made equal.
- Cost / benefits.
Backwards Compatibility

- Player Data
- DLC / Bought Content
Security

- Auto-ban system
- Public-key cryptography
Tips!

• source control
• code review
• jira
Source Control

Tips!

• Delete branches when you’re finished.

• git: merge vs rebase

• Write useful commit messages. What you did, and why you did it. Include a JIRA ticket reference if there is one, but don't rely on that to fully explain the change.
• Do reviews promptly.

• Don't "Complete" code reviews with open issues.

• Mark issues as "Resolved" once they've been dealt with.

• Reply to comments. Add changes to a review where you've fixed issues.
Tips!

• Say something. Complements and questions as well as criticisms!

• Don't be defensive about your code, don't dismiss criticism with "it's a work in progress" or "it's temporary code". Learn from it.
JIRA Tips!

- Use the Workflow buttons.
- Un-assign and Stop Progress issues if you get interrupted to work on something else.
- Add comments with any information that would help other people make progress.
More Tips!

- Ship with a data-push system.
- Ship with a revenue generating feature that can be extended in data.
Finally

• Source Control
• Automated Builds
• Continuous Integration
Questions?

AMA

thad@bossalien.com