Game Testing

Ensuring a quality product

Programming – Game Development Foundations

Last modified 26/11/15 by Richard Taylor

aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

Contents

- Why do we test games?
- Different types of testing
- Testing for fun
- Testing for reliability
- Testing for requirements and obligations

Why do we test games?

Why do we test games?

- Make sure they’re fun
- Make sure they work reliably
- Make sure we’re meeting our requirements and obligations
- These different testing goals necessitate different approaches to testing.

How do we make sure a game is fun?

- Why?
  - So that people want to play it when it’s done!
- What do we do?
  - Play testing

How do we make sure a game is fun?

- Who does it?
  - Designers
  - QA staff
  - Most importantly: members of our target audience!

- When do we do this testing?
  - Start early during design time
  - Keep going until the game is released

How do we make sure a game is fun?

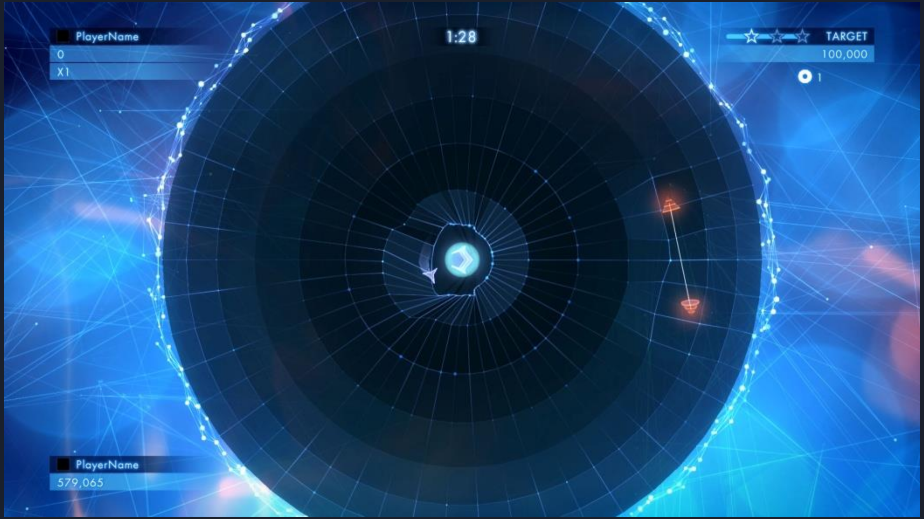
- How do we do it?
  - Get players to play the game. If it’s not playable yet that’s cool – use concept art or mock screenshots.
    - Record their impressions.
    - Ask for feedback, thoughts and opinions. Eg: have a survey.
    - Use analytics tools in your game to record how people play.
  - Have testing goals, and address them specifically in test sessions.

How do we make sure a game is fun?

- What do we do with the results?
  - Analyse them to inform updates to your design.
  - You want to identify the underlying issues!
  - Early in the game’s life cycle this will likely lead to large changes to things like mechanics, rules, or style and presentation.
  - Late in the game’s life cycle it will likely lead to smaller changes like control tweaks, level layouts changes and difficulty tuning.

Play testing example

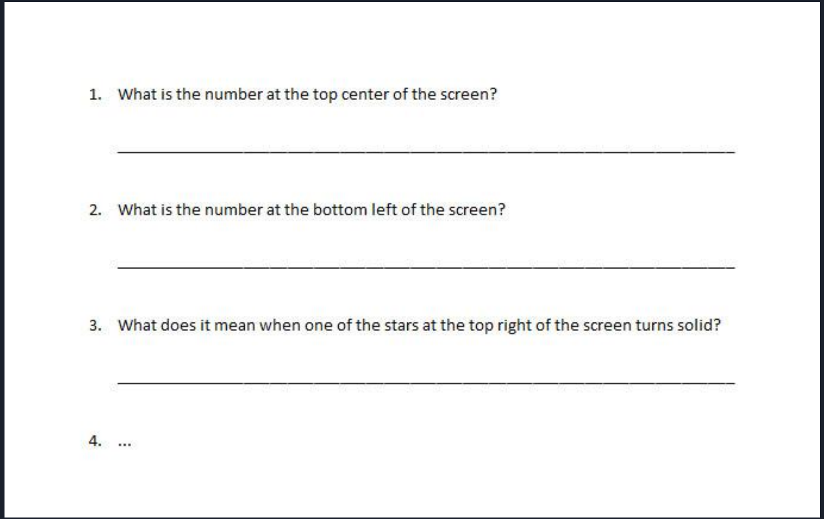
- 1. Set a play testing goal.
  - *"See if players understand our game's HUD."*



Geometry Wars 3: Dimensions Evolved

Play testing example

- 2. Make a set of questions which address that goal.



1. What is the number at the top center of the screen?

   ____

2. What is the number at the bottom left of the screen?

   ____

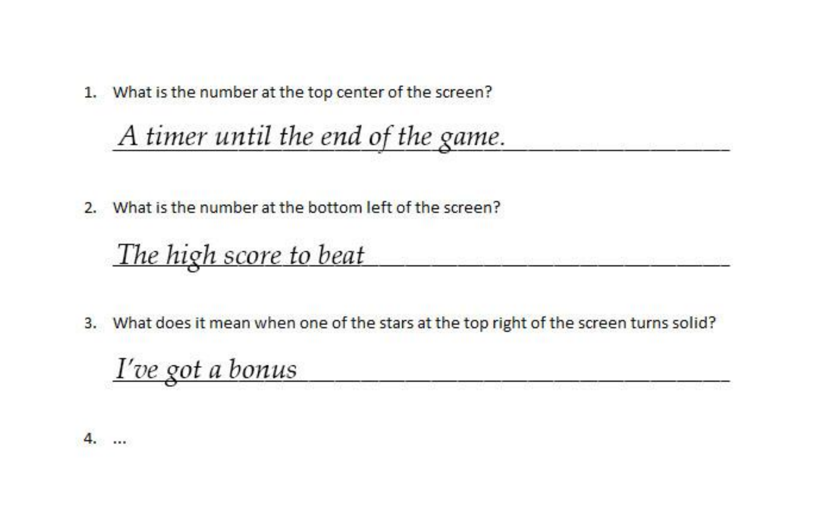
3. What does it mean when one of the stars at the top right of the screen turns solid?

   ____

4. ...

Play testing example

- 3. Have people play your game and answer the questions.



1. What is the number at the top center of the screen?

   A timer until the end of the game.

2. What is the number at the bottom left of the screen?

   The high score to beat

3. What does it mean when one of the stars at the top right of the screen turns solid?

   I've got a bonus

4. ...

SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Play testing example

- 4. Analyse the results.
  - Was the HUD generally understood?
  - Were there any common mistakes?

- 5. If necessary, use the results to update your design.
  - Don’t forget to test your changes!

How do we make games work reliably?

- Why?
  - Minimise bugs and crashes.
  - Consistent and acceptable performance.
- What do we do?
  - Functional testing.

How do we make games work reliably?

- Who does it?
  - Developers
  - QA staff
  - Play testers

- When do we do it?
  - Ongoing from the beginning of full development.
  - This often continues past release!

How do we make games work reliably?

- How do we do it?
  - A mix of “black box” and “white box” testing.
  - “Black box” testing is players reporting when something unexpected happens.
    - Player bug reports.
    - Support requests.

How do we make games work reliably?

- How do we do it? (continued)
  - “White box” testing is developers making sure their code and content does what they think it should do.
    - Buddy checks.
    - Internal bug reports.
    - Various code tests (eg: “unit tests”)
  - “Soak testing”. Running a build on a target device for days.

How do we make games work reliably?

- What do we do with the results?
  - Log them into a bug tracking system.
    - Eg: Jira or FogBugz
  - Confirms or discard each one.
  - Prioritise them.
  - The development team then addresses them.

Functional issue example

- 1. A bug report is received.
  - *“High score table does not show at the end of the game. Using iOS 8.2 on iPhone 5S.”*

Functional issue example

- 2. QA reviews the report.
  - Loads the game on multiple devices
  - Finishes a level and waits for high score to appear.
  - Notices that in each case the high score window appears and has player icon images... but no text is visible!
  - The bug is confirmed.
  - We have additional information to add:
    - The window appears and icons show
    - We can give specific steps to reproduce

Functional issue example

- 3. The issue is prioritised
  - The issue is given a high priority
    - This is a common issue on multiple devices.
    - It effects one of the game’s major features.

- 4. The development team addresses the issue
  - A developer finds that the text colour is the same as the background colour. Easy fix!

How do we make sure a game meets requirements and obligations?

- Why?
  - Make sure our game gets released.
  - Make sure we get paid.
  - Make sure we don’t get in trouble!

- What do we do?
  - This is a part of our project management.

How do we make sure a game meets requirements and obligations?

- Who does it?
  - Project manager.
  - Designers.
  - QA

- When?
  - Ongoing from the project’s conception.

How do we make sure a game meets requirements and obligations?

- How do we do it?
  - Identify and document any requirements and obligations as they arise.
    - Publisher contracts?
    - Platform agreements? (Eg: Apple’s iTunes App Store Review Guidelines)
    - Legislation
    - Cultural considerations
  - Make sure that these requirements are addressed in our design
    - This may have to be reviewed from time to time!
  - Make sure that the implementation follows the design
  - Keep positive relationships with third parties!

How do we make sure a game meets requirements and obligations?

- What do we do with the results?
  - Guide design decisions.
  - If issues do crop up, work with the design team to address them and get changes into the development pipeline.

Summary

- Now you have some understanding of:
  - Why we test games.
  - Different goals of testing.
  - Different types of testing to meet those goals.
  - When different types of testing should take place, and who should be involved with each

References

- Ambinder, M, 2009, Valve’s Approach to Playtesting: The Application of Empiricism, GDC 2009
  - http://www.valvesoftware.com/publications/2009/GDC2009_ValvesApproachToPlaytesting.pdf

- Extra Credits, 2012, Playtesting, Extra Credits
  - https://www.youtube.com/watch?v=on7endO4lPY

References

- Schreier, J 2015, Quality Assured: What It’s Really Like To Test Games For A Living, Kotaku
  - http://kotaku.com/quality-assured-what-it-s-really-like-to-play-games-fo-1720053842
- St. John, V 2013, Best Practices: Five Tips for Better Playtesting, Gamasutra
  - http://www.gamasutra.com/view/feature/185258/best_practices_five_tips_for_.php
- Brown, D, The Five Essentials for Software Testing, iSixSigma
  - http://www.isixsigma.com/industries/software-it/five-essentials-software-testing/