Game Dev Life Cycle & UI Design

Jaanus Jaggo
Game development life cycle (GDLC)

Different from standard software development?
Game development life cycle (GDLC)

Different from standard software development.

- Game consists of code and **assets**
- Game dev has more roles/people
Designers

Responsible for the look and feel of a game.

- subroles: scriptwriter, level designer, ...

Game Designer

What my mom thinks I do  What my friends think I do  What society thinks I do

What programmers think I do  What I think I do  What I actually do
Art and animation staff

Responsible for the visual look of the game.
Sound and music engineers

Produce the music and sound effects for the game.

● also integrate the soundtrack into the game programming
Programmers

Create the code for game engines.
- gameplay programmer
- graphics programmer,
- AI programmer,
- etc.
Quality Assurance and testing

Make sure the game works and meets its specifications.
- should have a broad view of games to compare against
- not just finds bugs but provides information to fix them
- responsible for the delivery
Managers

Oversee the entire project.
- make sure everything is running well and project is in schedule
- many subroles: art lead, programming lead
- someone has to think about the business
The Phases

The GDLC typically consists of six phases shown below:
Initiation phase

The developer decides what kind of game they will make.
Pre-Production

Before the real production begins.

First pre-production:

- **Game Design Document**
- **First prototype** - shows gameplay *(has to be fun)*

Next cycles -> bug fixing and balancing
Production phase

The game assets and source code are made. The result of production is the playable game in form of:

- **Formal Details prototype** - a playable game, has win-lose rules, co-relations between features, runs well.
- **Refinement prototype** - most mature prototype which only needs more polishing. Should be feature complete and almost ready to ship.
Vertical slice

Tiny portion of game with the final quality.

- standard for the final quality
- early promotional material
Testing phase

Evaluation of game features, value, concept and design.
Testing phase

Evaluation of game features, value, concept and design.

Some questions the testing should be answer:
1. Is the game still buggy?
2. Is it possible to get stuck in the game?
3. Is there any sign of exploits/glitch?
4. Is the game too easy/hard to beat?
5. Is there any feature missing?
6. Does the game run well on every platform?
Beta Phase

The beta test is testing cycle conducted by third party:

- publisher,
- potential buyer,
- game reviewer

Should result a test report.

Decide whether the game is ready for shipping.
Release Phase

Work may seem to be done but it is not!

- Bugfixing, patching
- Additions, special events
- Marketing
- Community management
UI design

What information should the player have on his screen? Where? When?
UI design

- Menus
- In game UI
Types of **in game UI**

In game UI:

1. Non-Diegetic
2. Diegetic
3. Meta
4. Spacial
Types of in game UI
Types of in game UI
Dead Space integrated UI

Opening the menu doesn’t pause the game
Minimaps

Using your Witcher Senses, search for Temerian shields with small white flowers painted on them.
Alternative for Minimaps
Subway-Legibility

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

- Easier tutorialization
- Improves organic discovery
- Easier marketing

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

Principal of three reads:

- First read

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

Principal of three reads:

- First read
- Second read

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

Principal of three reads:

- First read
- Second read
- Third read

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

First Read

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

First Read

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

Second Read

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

Using animation to swap reads

http://stfj.net/DesigningForSubwayLegibility/
Subway-Legibility

Toggle to the forth read

http://stfj.net/DesigningForSubwayLegibility/
Accessability

1 in 100 people are green-blind

Accessability

Subtitles, and not just for dialogue
Menu Design

Your game experience starts in menu

Splash screen is OK, but a montage video is too much!
Balance

- Rules
- Space and void
- Consistency

Example of League of Legends UI Style Guide:
Design rules for menus

- Allow players to skip the splash screens
- Make “Continue” the first option in main menu
- Automatically save settings once they are changed
- Use ‘A’ to advance and ‘B’ to go back
- Allow players to quit to desktop

https://kotaku.com/5955855/the-ten-commandments-of-video-game-menus
Use the same font
Plan ahead
Controller support

Make sure that all the functionality is accessible with controller.
Controller support
Test your UI design

Welcome to our website
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Learn more

Click rate: 52 %

Welcome to our website
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Learn more

Click rate: 72 %
Create visual hierarchy
Tooltips

Total County Tax

Yearly Tax Paid to Us / Total Yearly Income

Mathrafal: 4.00/4.00
Radnor: 3.05/12.20
Dishonored - stylized UI
Besiege - flat design
Evolution of Hearthstone UI
Evolution of Hearthstone UI
Conclusion

1. Predict what the user wants to know.
2. Easy to find information.
3. Easy to use and navigate.
4. Make it obvious what the user can do.
5. Eliminate repetitive tasks.
Project task

UI mockups + scheme

- **Mockup** - layout of your views
- **Scheme** - how the mockups are connected

**Required:**
- Head-up display (HUD) mockup,
- All the menu view,
- UI navigation scheme