Giant Cow Games

Darkmatter

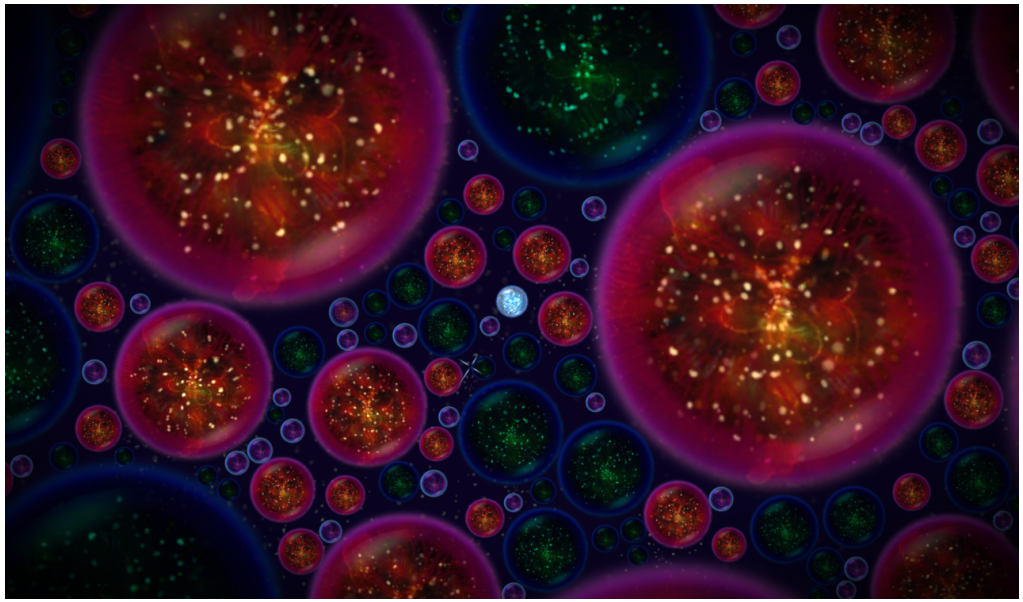
J. Greenaway C. Horrell Y. Wang J. Via

23 March 2011

Outline

Outline

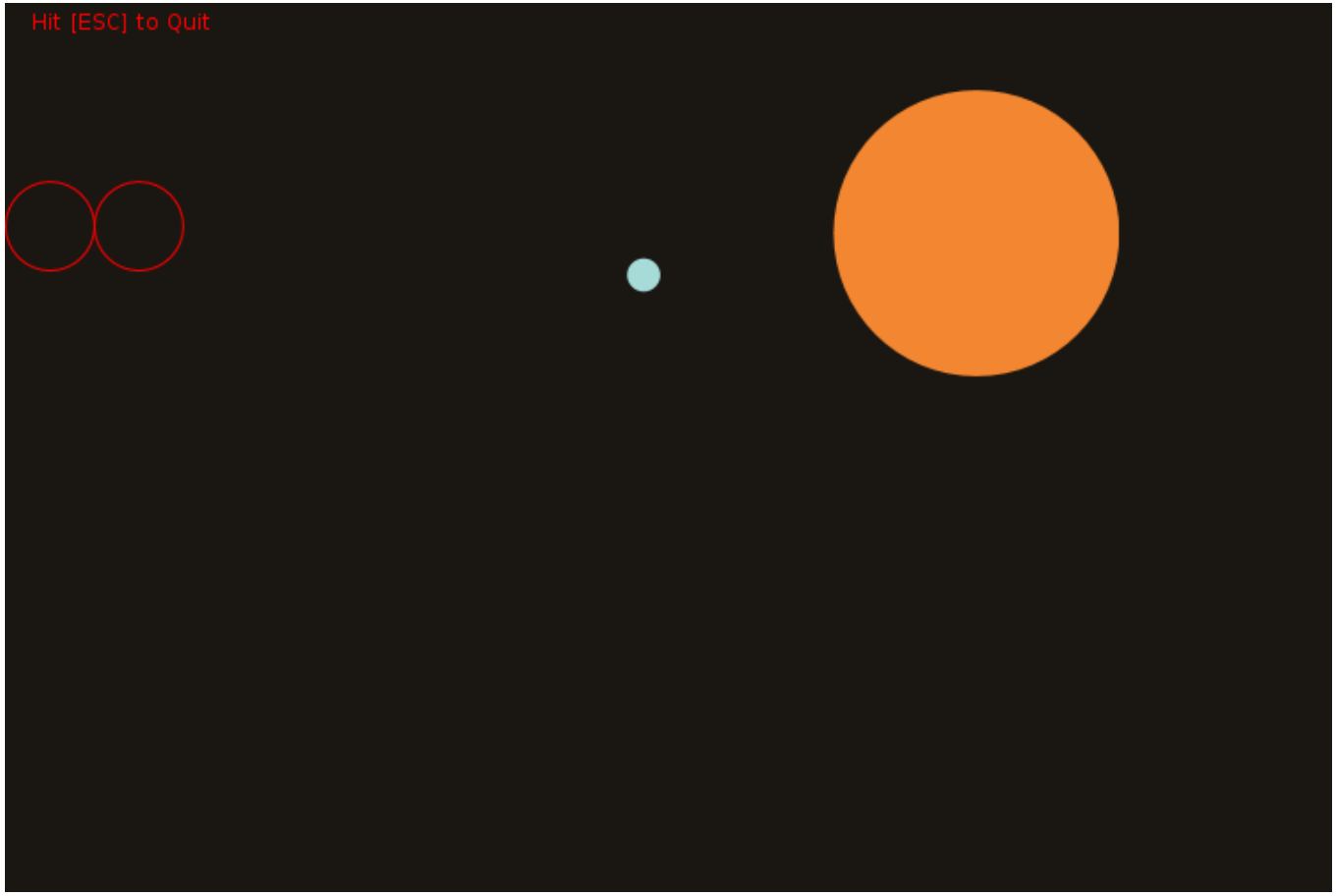
Osmos



A beautiful puzzle game created by Hemisphere games.

Outline

Early Prototype



Final Version

The Future

Enhancements

- Defensive power-ups
- Co-op mode
- Completing missions

And here we go...

Why XP?

- Release early & release often
  We built a small core and added functionality feature by feature.
- Collective code ownership
  We are all responsible for fixing bugs in any file.
- Planning game
  We all had a sense of how much effort a feature required.

But we weren't perfect

- *Not as many release as we aimed for.*
  Hard to remember to create tags whenever a feature milestone was reached.
- We coded our *unit tests after our features*.
  New domain = exploratory coding.
- *No overtime.*
  We can't push back client deadlines.

Outline

Weekly Meetings

	Tuesday	Thursday
10:00	Team Meeting	
11:00	Sessions One	
12:00		Session Two
13:00		
14:00		Session Three
15:00		Demonstrator Meeting

Outline

Declaring intent

- 1 test for every 75 lines of code.
- Added new test when we found a bug.
  Example: conflating radius and diameter.

Outline

Or when building became easy

- Automatic dependency resolution
- Supported in all IDEs
- Built in unit testing & documentation generation

```
|- src
| |- main
| | |- java
| | | |- com
| | | | `- giantcow
| | | | `- darkmatter
| | | | |- level
| | | | |- net
| | | | `- player
| | `- resources
| `- test
| |- java
| | `- com
| | `- giantcow
| | `- darkmatter
| | |- level
| | |- net
| | `- player
```

Mistakes

- We needed to *communicate more*.
  Team leader disappeared and the void was never officially filled.
- Making code assignments *explicit*.
  Ambiguity of responsibility caused code to go unwritten longer than it should have.
- Missed meetings and pair programming sessions.
  We let deadlines get in the way of our sessions together.

The take away

- Working in a team is hard.
- Creating a final product requires a lot from the whole team.
- Constant forward momentum.

Summary

- Game programming is rewarding but hard.
- Team work requires compromise and dedication.