

Jugglers: Post Mortem



We chose to develop Jugglers, a ball-based multiplayer physics game that mixes motoric skills with concentration/multi-tasking and tactics. Keep juggling your balls, mess up your enemies with fun attacks, don't get messed up yourself and - most of all - do gain the highest score before that match ends.

The gameplay tends to be both relaxed and demanding. Gravity and your opponents constantly force their pace on you. But juggling a single ball to and fro can be quite calming.

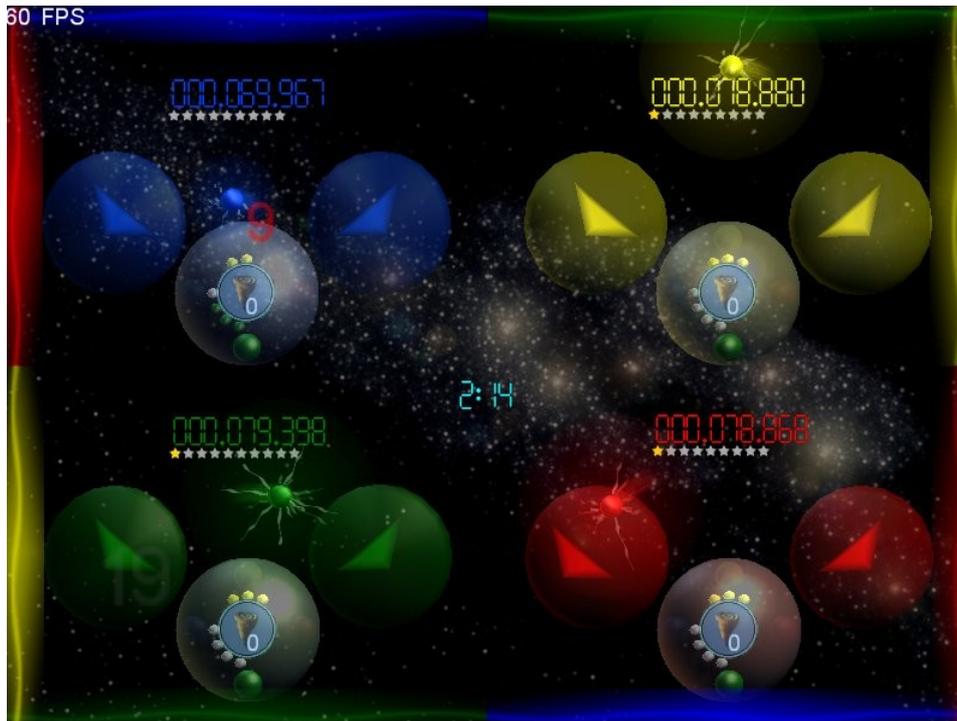
What went well in our team

1. Decision making was a simple enough process at times. Specifically, the proposal to implement a map editor was brought up early by one team member. Another then spontaneously proceeded to build it from scratch (whipping up a formidable tool in mere two weeks). The editor allowed us to explore quite some possibilities in gameplay and became an important engine of iteration. In the final stages, it saved us valuable time by doing WYSIWYG rather than C#- or XML-hacking of map definitions.

The lessons learned: It can be important to make strategic decisions early. Fast realization is just as critical.

2. Fast realization of plans was among our strengths all the way through. Starting from 3 prototypes being developed in but a week, thus allowing us to give (almost) each of them enough additional love to enable us to predict their respective potentials.

Other key elements such as the above mentioned map editor, the in-game tutorial and underlying screen management system were put to work rather quickly, so that time could be spend on refinements, additional features and iteration.



Peaceful juggling. Blue just lost some load.

3. Meeting up in person was kept at a reasonable level. During the early stages, we would not meet up at all. Our first „session" was two days prior to the deadline for the final decision among our prototypes (and was arranged for the sole purpose of making that decision).

Weekly meetings were not held until about half the allotted time for the project was up, and even then we would only meet a couple of hours before the public presentation/testing-session to implement final tweaks and do some „pre-iteration".

Apart from those brief sessions, we worked each on his own, spending as much time as possible actually working instead of talking (or traveling to and from each other's). This is a matter of personality, of course, and we so happened to each function best when left alone, rather than requiring massive feedback from our peers.

4. The roles were assigned just right. Period.

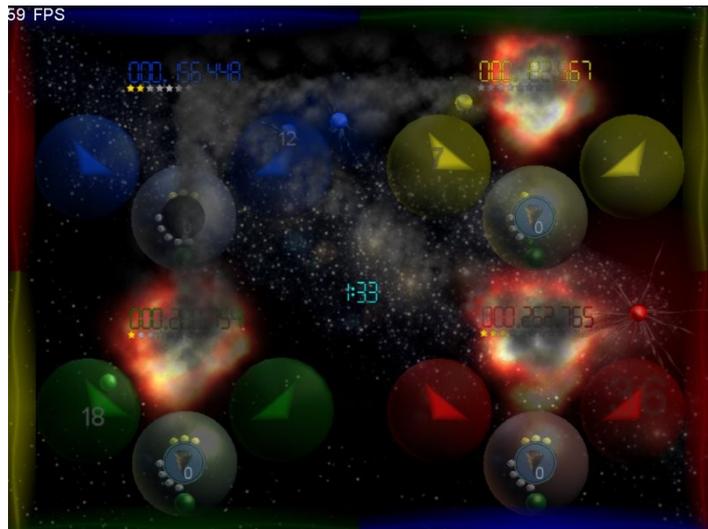
Most of the time we even adhered to them quite completely. Our graphics programmer used his specialized experience to great advantage, creating an increasingly stylish Sci-Fi-Look over time and construct a flexible foundation for geometric rendering that defies XNA's shortcomings in that area.

Our game programmer brought on lots of coding experience, allowing us to tackle challenging aspects early and to implement difficult concepts in feasible time. The team lead is a hobbyist game designer, so go figure.

Note: Our team didn't comprise even the remotely best coders or graphics people in the project group. Still, we made good use of what we did have.

5. Using the tools went rather smooth. Event though not leveraging all aspects any used tool might offer (see „What went bad" for a hint as to why we didn't pay much love to XACT), we did have our moments.

We fed VisualStudio quite a couple of `///summary>` and `///TODO` comments to help us with. Subversion and the fine Windows shell extension of TortoiseSVN were our best friends in cooperating. We'd even branch and merge here and there, but in hindsight we would suggest to go easy on automated merging and to consider reimplementing of proven (in a branch) concepts in the trunk.



Blue attacks everyone rather fiercely.

6. Giving up on bad ideas and generally throwing things away. This is tremendously important. In certain development stages, Jugglers offered

- the need to analogously bank each paddle inward using the triggers
- a HUD menu for each player to choose among...
- 7 different Extras

etc. etc.

It often makes more sense to take something out than adding something. Listen to your testers intently. Take notes. *Throw things out.*

What went bad in our team

1. Lack of iteration was our single greatest flaw. We pretty soon had a pretty clear image of what Jugglers should be.

This is wrong thinking.

We soon let our time be eaten up by implementation of „necessary" features, while the game core remained relatively stable throughout the greater part of the semester.

This is *wrong thinking*.

A lot of concepts might have been tried (many of them rejected, no doubt) if we had only taken the time to hack up more prototypes along the way. We feel that Jugglers might have turned out fundamentally different and possibly a lot better (even if technically inferior).

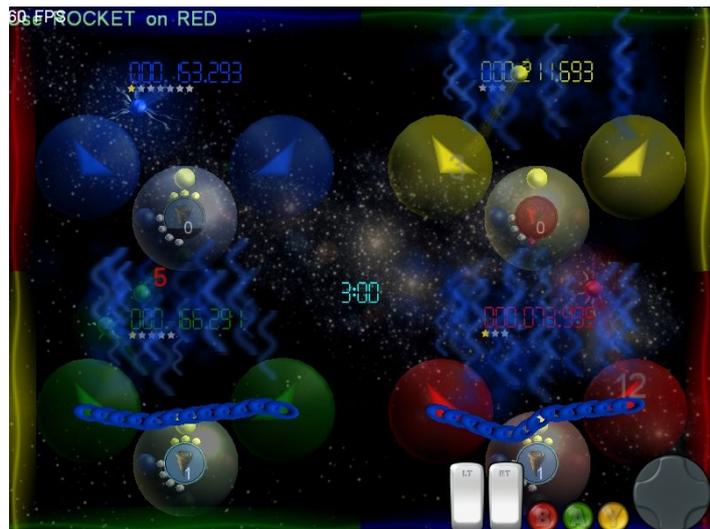
2. Communication was a big one. We met no more than once a week. We never talked on the phone. Only two of us were on IM or chat regularly. We did write 450 Emails all told, but there was no discipline to the written contact. We also had no way of punishing slackers in the team.

The end of the story was us kicking out our fourth member (our content creator) after we had failed to make him work harder (or work at all) on the project.

3. Decision making was listed under „what went well“, but in many places, we spent too much time thinking and arguing where quick implementation and iteration would have been the way to go.

That one's easy mistake to walk into, so beware.

4. Code quality and comments were rather decent when we began. Soon, the hacks kept sneaking in relentlessly though. Towards the end, we had to spend valuable hours refactoring.



Blue has handcuffed some opponents and now throws in some Rising Wind.

5. Balancing the load was probably the thing we did second worst of all. This became most obvious when we had to break up with our content creator, leaving us with XACT, Photoshop and all the other fun tools we had no time for applying. Furthermore, one team member got somewhat left behind in regards to responsibilities, so that the bulk of the work had to be shouldered by mere two people.

Do evaluate individual contributions early on, and find ways to make optimal use of everyone's abilities and resources.

Conclusion

Developing Jugglers from a small prototype to a mostly complete game was a rather interesting experience none of us would like to miss. It was a lot of work (though we had been warned), but in the end we managed to finish a playable and consistent looking game that proved to be fun to play.

Furthermore, we created this game around the subject of juggling balls, something which hasn't been done often in the past.

Specs

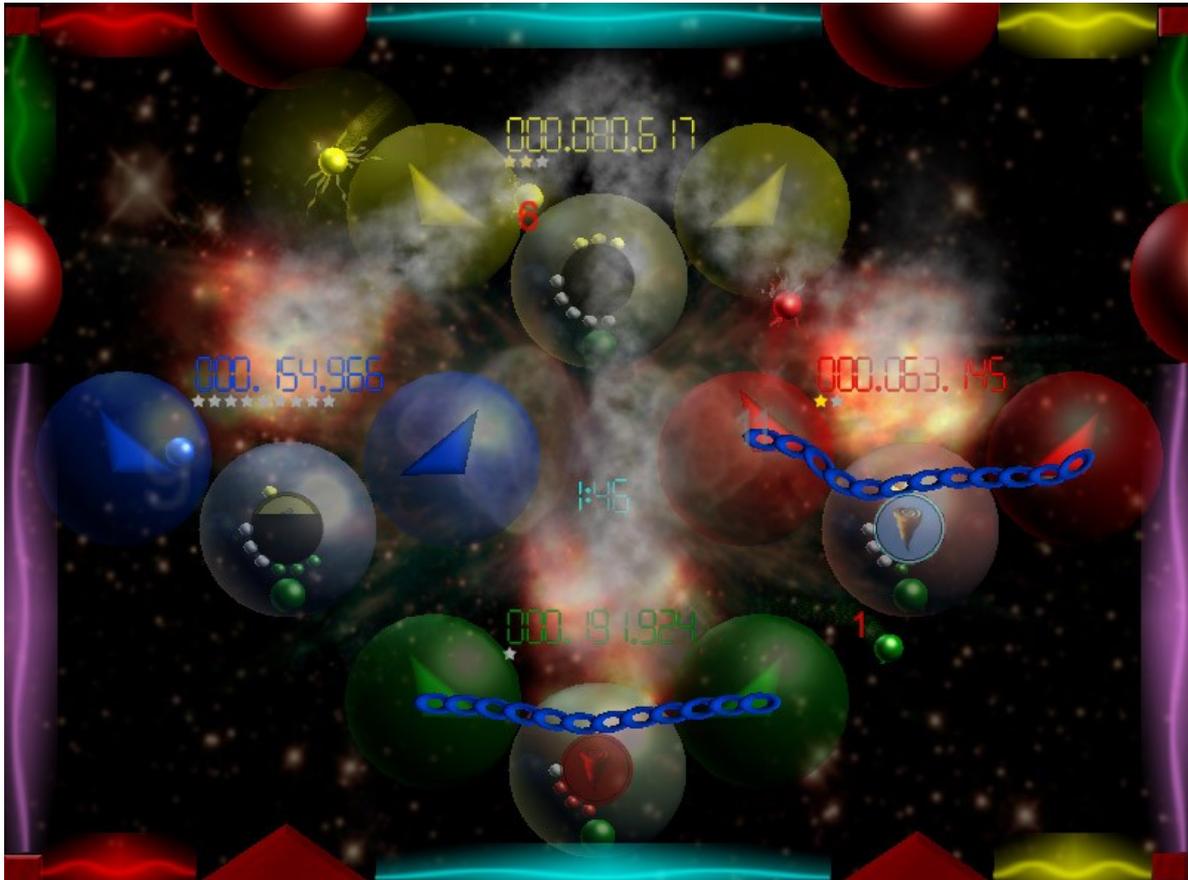
Name:	Jugglers
Developer:	Better Than Work
Team Members:	3 (originally 4)
Development Time:	10 Weeks
Tools used:	XNA Gamestudio 1.0 Refresh Photoshop Audacity TortoiseSVN
System Requirements:	PC to run WinXP, .NET Framework 2.0 and XNA Runtime 1-4 Xbox360 Gamepads preferably a graphics adapter that supports Shader Level 2.0

[Better Than Work] are:

Georg Zabel

Felix Frank

Jakob Tonn



Fierce fight for points: Blue has handcuffed two of his opponents, Yellow makes full use of his rockets

Where to get it?

<http://www.cg.tu-berlin.de/gamepro.html>