

A Parameter-Space Design Methodology for Casual Creators

Simon Colton, Mark J. Nelson, Edward J. Powley, Swen E. Gaudl,
Rob Saunders, Blanca Pérez Ferrer, Peter Ivey and Michael Cook

The MetaMakers Institute, Falmouth University, Cornwall, UK

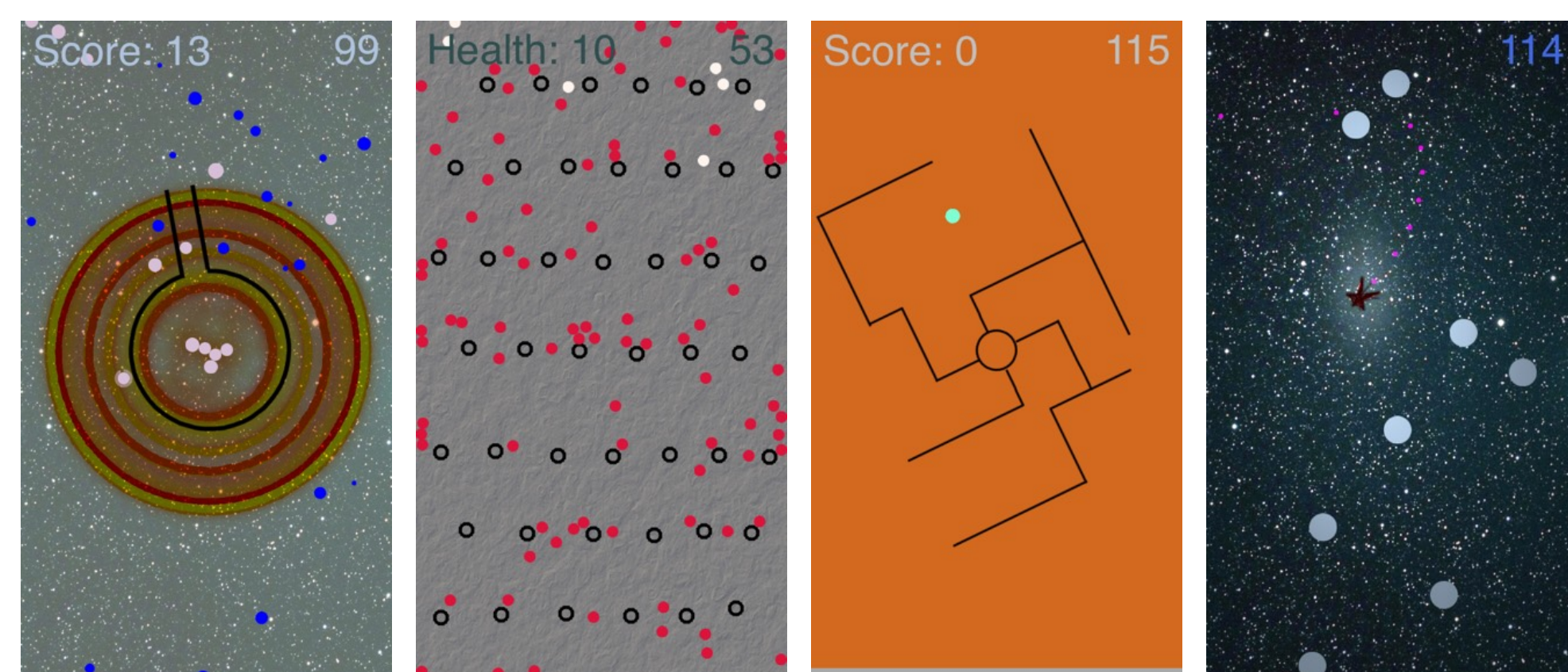
Casual Creators

A **casual creator** is a creativity support tool intended to be fun and easy to use for exploratory creation (Compton & Mateas, ICCG 2015).

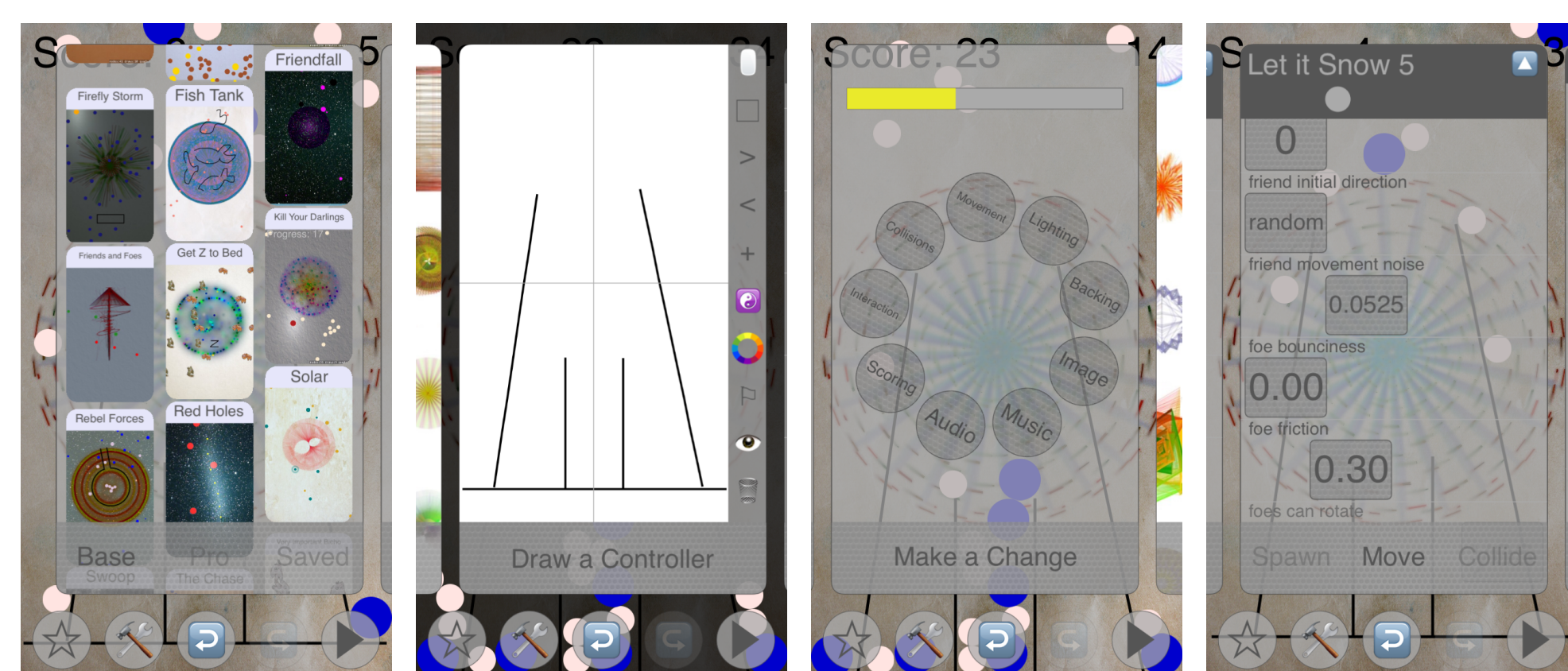
We have developed casual creators for game design called **fluidic games**. They support exploratory game design on mobile devices without programming, by encoding games in a parameterised design space. Player/designers can create new games by varying the parameters in a design interface and seeing changes in their design instantly.

Gamika Technologies

Gamika Technologies is our parameterised mobile game engine. It has 284 parameters and a power-user design interface, *Cillr*. Our fluidic games (see right) were designed starting from cut-down versions of the full Gamika design space.



Some games in the Gamika Technologies engine.



Cillr design panes for designing Gamika games.

Research Question

When building a casual creator that is based on a parameterised design space, what methodology should be used to choose a suitable parameter space?

Based on our experiences developing fluidic games, we propose seven sources of parameters, and a four-stage process of incremental parameter-space expansion and contraction.

Sources of parameters

- 1. Capture an initial example:** Choose one game to implement, and identify the minimum set of parameters needed to represent that game.
- 2. Externalise fixed parameters:** Systematically turn values that were hard-coded in the first game into parameters of the design space.
- 3. Capture an inspiring example:** Think of a game close to the existing space, and add new parameters until either that game, or something like it, can be represented.
- 4. Pass through parameters from underlying technology:** Study the fields and input parameters to methods of the APIs in the programming environment being used, and expose suitable ones as parameters in the design space.
- 5. Balance other parameters:** If it becomes clear that having parameterised a game setting X, you really should be able to change Y too, then add the extra parameter.
- 6. Reify emergent features:** When experimenting with a parameterised design space, combinations of features often produce novel emergent effects. Add these as explicit parameters.
- 7. Split parameters:** Some parameters may be employed for more than one aspect of the game design due to bundling things together in the code; consider splitting these.

Expansion/contraction

Stage 1. Unconstrained expansion: Add a large number of parameters, sourced via all seven methods, to map out as general a space as is feasible. This is how we arrived at Gamika.

Stage 2. Cut down the parameter space to just a single game: Choose a promising game and build a bare-bones casual creator UI for designing variants of that game. For *No Second Chance* and *Wevva*, we cut down the parameters in Gamika to those required just for the initial games, respectively *Pendulands* and *Let it Snow*.

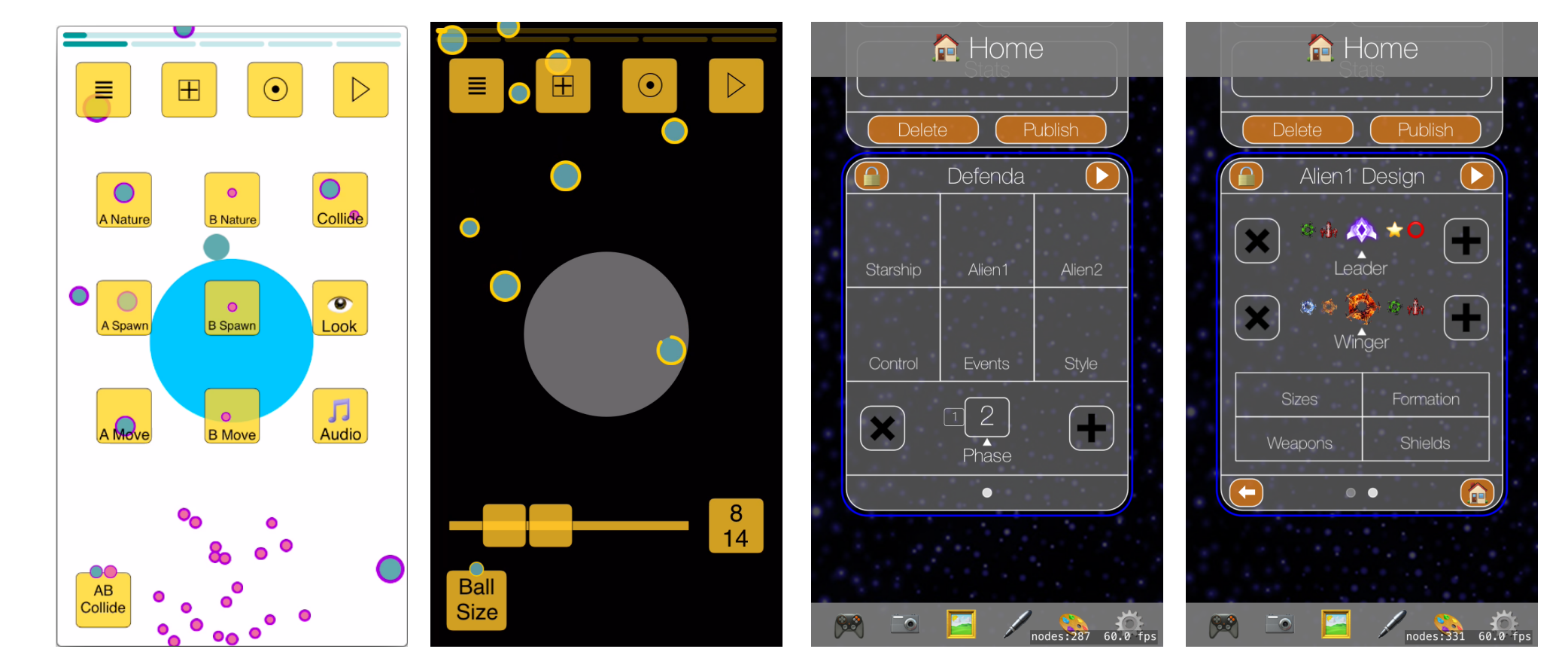
Stage 3. UI-constrained expansion: Re-add parameters with a strong UI constraint. Every new parameter must be added to the UI immediately, and the interface must remain usable as a casual creator. In some cases, we have chosen further constraints to structure this process, such as *Wevva*'s 3x3 grid and top-level rule summary.

Stage 4. Consolidation and polishing: Even though Stage 3 only adds parameters under a strong UI constraint, the incremental addition process may have resulted in parts of the UI becoming complicated or inconsistent. Relatively modest streamlining can be done by consolidating or linking similar parameters, rethinking arrangements, etc., in preparation for the final version.

Fluidic Games

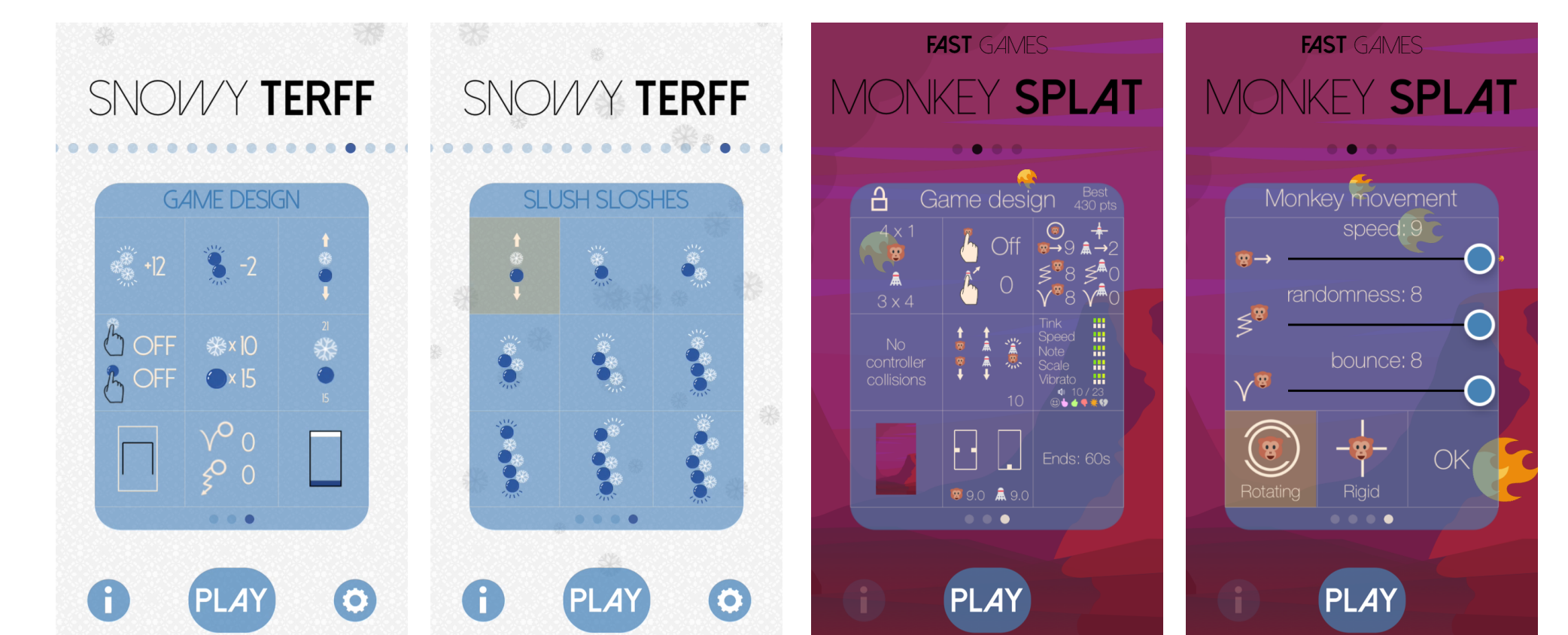
We have built three fluidic games by focussing on specific parts of the larger Gamika space: *No Second Chance*, *Wevva* and *Blasta*.

Wevva has gone through the most complete development and playtesting of the three, and has been released on the iOS App Store.



(a) *No Second Chance*

(b) *Blasta*



(c) *Wevva*, early version

(d) *Wevva*, released version

Design screens from our fluidic games.

Acknowledgements

This work is funded by EC FP7 grant 621403 (ERA Chair: Games Research Opportunities).

FALMOUTH
UNIVERSITY

