Fair Voronoi Split-Screen for N-Player Games

Tobias Lenz[1]

1 **Hochschule für Technik und Wirtschaft Berlin**
tobias.lenz@htw-berlin.de

Abstract

A consistent expansion of the well-spread dynamic two player split-screen to a larger number of players is introduced and formally defined. Unfortunately such a pure solution does not exist, as is proven in this paper. A visually appealing approximation is presented and discussed.

1 Introduction

1.1 Motivation through Multiplayer Games

In the early days of 2d computer games, most local multiplayer games were played on a single screen. Newer gaming consoles revived this scenario for multiple players in front of a single television set or console display.

This leaves two options to the developer: either all players have to stay very close to each other in the virtual realm, or each player has its own independent window to the game's world and is allowed to stroll around freely. The former case severely limits gameplay while the latter case requires a subdivision of the physical screen space into as many independent windows as players participating. The most obvious and often used subdivisions for two players are horizontally or vertically in the respective center. Applying both subdivisions simultaneously solves the typical four player scenario.

The mentioned stationary horizontal and/or vertical subdivisions have multiple drawbacks, we are going to tackle in this paper:

1. If two players stand right next to each other, both their windows would show the exact same surroundings, essentially wasting half of the total screen space.
2. Multiplayer games might want to hint the players on their relative positions, e.g. player one being left and slightly above player two. This requires additional display elements like arrows, further cluttering the screen.

1.2 Dynamic Split-Screens to the Rescue

The two problems mentioned above can be fixed with a single simple concept often called "dynamic split-screen". A simple variation of it already appeared in a game in 1983 [2] basically only solving the first mentioned drawback. Developers sporadically used and improved it to also solve the second drawback ever since without much scientific interest. It realizes the simple idea, that the separating line between two players need not be static but might change and even vanish according to the player positions, as indicated in figure 1.

For two players there is not a lot of math to it. The separator in screen space is perpendicular to the vector between the player positions in world space, and can therefore be computed with a single arctan.

1.3 Formal Problem Formulation

Note that there is no ground truth. All criteria were chosen with aesthetics, fluent graphics and gameplay in mind.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

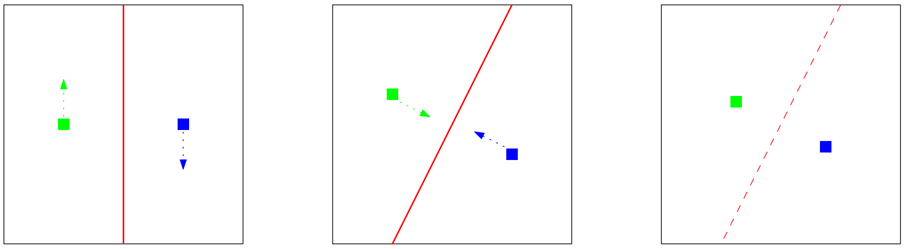


Figure 1 The separator (red) changes its orientation dynamically according to the player positions (green, blue) while they move (arrows). It will vanish for players being close to each other (right image).

The following definition is helpful to simplify writing.

▶ **Definition 1.1.** Let the sum of a polygon P and a vector v be the polygon translated by that vector, i.e. the vector added to every point of the polygon: $P + v = \{p + v \mid p \in P\}$.

Today game worlds are massive in size but usually not infinite. We consider them finite and surrounded by thick impassable walls, so we can treat them as infinite here.

▶ **Definition 1.2.** The *world space* is $\mathbb{R}^2$. Players move continuously through the world with position $w_i(t) \in \mathbb{R}^2$ for player i at time t.

The display hardware has a fixed amount of pixels and a given aspect ratio, but we abstract this and use a square with real coordinates.

▶ **Definition 1.3.** The *screen space* is $[-1, 1]^2$. The screen space position for player i at time t is denoted by $s_i(t) \in [-1, 1]^2$.

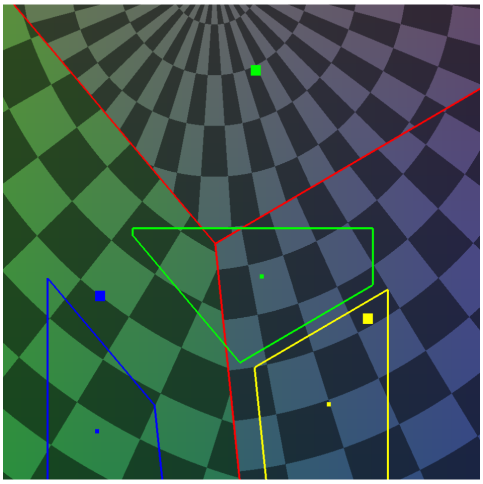
Since every coordinate used here is time dependent, the parameter t is omitted throughout the paper to avoid cluttering. A "scene" with fixed t is considered and the following main definition must hold for every t.

▶ **Definition 1.4.** A *fair voronoi split-screen* for n players comprises a set of n convex polygons $S_0, \ldots, S_{n-1}$ forming a subdivision of the screen space, one for each player, and one designated point (screen space position) inside each S_i, fulfilling the following criteria:

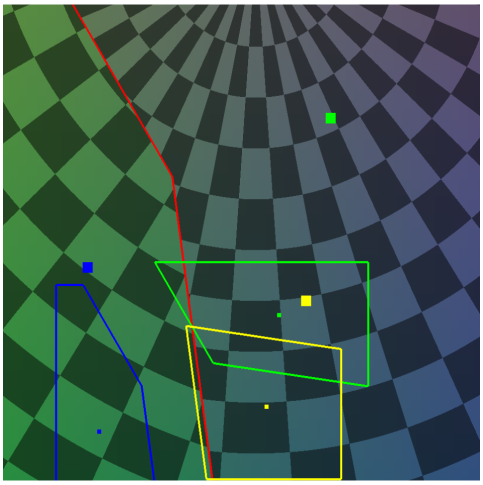
1. **Fair**: All S_i have equal area.
2. **Direction-indicating**: If S_i and S_j share a boundary, this boundary would be parallel to the bisector of w_i and w_j.
3. **Fusible**: If S_i and S_j overlap in world space, formally $(S_i - s_i + w_i) \cap (S_j - s_j + w_j) \neq \emptyset$, the boundary between S_i and S_j would be omitted, thereby *fusing* S_i and S_j.
4. **Centered**: s_i is the center of the inscribed circle of S_i for all non-fused S_i. If S_i is fused with one or more other polygons $S_j, S_k, \ldots$, the centroid c_i^W of $w_i, w_j, w_k, \ldots$ is mapped to the centroid c_i^S of the inscribed circles of $S_i, S_j, S_k, \ldots$ and $s_i \leftarrow c_i^S + w_i - c_i^W$.
5. **Continuous**: Just as w_i moves continuously, so too s_i and the boundary vertices of S_i.

Fairness is obvious from a gameplay perspective, direction-indication and fusibility are the requested features from section 1.1, centeredness helps providing good visibility in every direction, and continuity is required for aesthetics and fluent animation.

Figure 2 illustrates parts of the definition. The three equally sized (fair) S_i are depicted including the corresponding s_i (centered). The red boundaries have the correct angles (direction-indicating). In figure 2b two screen space regions are fused.



(a) All three players are sufficiently far away from each other in world space, resulting in three disjoint screen space regions.



(b) The screen space regions for the green and yellow player overlap in world space, hence they become visually fused in screen space.

Figure 2 Three players (thick squares in green, blue, yellow) with their respective region in screen space, the shared boundaries of the S_i in red, and the relative positions of the S_i projected to world space (colored polygons at the bottom).

Parts of definition 1.4 resemble the well-known *voronoi diagram* [3]. Hence the name *voronoi split-screen* became commonly accepted. Do not jump to conclusions because observation 1.6 tells us that we are not dealing with normal voronoi diagrams here.

▶ **Observation 1.5.** *The* dynamic split-screen *for two players mentioned in section 1.2 equals the voronoi diagram of the two player positions and fulfills definition 1.4.*

▶ **Observation 1.6.** *The* voronoi diagram *of $n > 2$ players in world space scaled uniformly to screen space usually violates the fairness and centeredness conditions, while providing direction-indication and being fusible. Therefore it is not a fair voronoi split-screen.*

1.4 Related Results

The two player dynamic/voronoi split-screen appears quite often in games without special emphasis, but only a few approaches for more than two players exist.

At GDC 2016 Eiserloh presented an approach [4]. They build the voronoi diagram of the player positions in world space, map it to screen space, and reposition the player in screen space to be the center of the inscribed circle in their corresponding area. Although the last step guarantees a nice centeredness, the split-screen is neither fair, nor continuously fusible.

A different implementation, utilizing only the GPU, was made freely available by an author with the pseudonym gorsman [5]. The voronoi diagram is used directly and hence the cells are nicely fusible, but the split-screen is neither fair, nor centered.

Both mentioned approaches are expandable to an arbitrary number of players without further complications. They lack fairness and either centeredness or fusibility.

2 Fair Voronoi Split-Screens are Almost Impossible

The fact that no fair split-screen by definition 1.4 is known for at least three players becomes quite understandable, as the following theorem states their non-existence.

The proof is quite lengthy, so it is given in two lemmas. Both consider the following case to derive a contradiction:

Let three players be positioned in world space at $w_0 = (0, 2), w_1 = (0, -2), w_2 = (100, 0)$. w_0 and w_1 are close to each other, but their distance is slightly larger than the screen space's side length. w_2 is very far to the right, vertically between w_0 and w_1. Obviously no two regions can overlap in world space, hence fusion is prohibitive in this setting.

▶ **Lemma 2.1.** *S_0, S_1, S_2 all pairwise share a bounding edge ("they are neighbors").*

Proof. Since S_0, S_1, S_2 are convex polygons with equal area and they form a subdivision of a square, either all are neighbors and we are done, or one polygon must separate the other two. Since the angles of possible boundary edges are fixed by the direction-indication property, one can go through all three cases and show that a correct subdivision is impossible under these circumstances. Illustrated in figure 3a is the case where S_2 should separate S_0 and S_1. Hence no polygon can separate the other two. ◀

▶ **Lemma 2.2.** *Fair Voronoi Split-screens are impossible for three players.*

Proof. Since S_0, S_1, S_2 must be pairwise neighbors in screen space by lemma 2.1, and the angles of their pairwise boundaries are given by the direction-indication property, we have two possibilities:

- *Try to keep all $\overline{s_i s_j}$ parallel to the corresponding $\overline{w_i w_j}$.* The only angle preserving transformation of the player positions from world space to screen space is uniform scaling. Since w_2 is far away, the scaling factor must be very small, and therefore the distance between s_0 and s_1 becomes very small.
  The common boundary of S_0 and S_1 must be between s_0 and s_1 which are very close to each other, hence there is no placement for the boundary with s_0 and s_1 being centered in their respective polygons, see figure 3b.
- *Parallelity is not preserved from world space to screen space.* Let x, y be two players where the shared boundary between S_x and S_y is perpendicular to $\overline{w_x w_y}$ due to direction-indication, but not perpendicular to $\overline{s_x s_y}$. Let player x and y move towards each other on the line $\overline{w_x w_y}$, effectively not changing the angle of the boundary of S_x and S_y. At some point $w_x = w_y$, but $s_x \neq s_y$, see figure 3c. This motion is not continuously fusible.

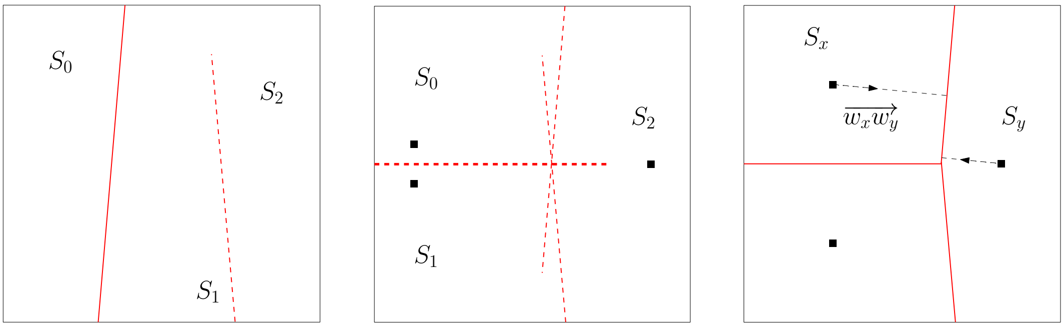
The only two possibilities are either not centered or not continuous and therefore contradict definition 1.4. ◀

▶ **Theorem 2.3.** *Fair Voronoi Split-screens are impossible for three or more players.*

Proof. For any number larger than three, we place the first three players as in the proof for lemma 2.2 and the others reasonably far to the right of the first three points. For the local situation of the first three players, the proof of the three player case applies respectively. ◀

3 A Quasi-Solution for Three Players

As established in the previous sections, a fair voronoi split-screen exists and is easy to implement for two players, while being impossible—and therefore obviously quite hard to implement—for three or more players.



(a) Placing S_1 not next to S_0 is impossible with given angles for boundary edges.

(b) The common boundary of S_0 and S_1 prohibits centeredness.

(c) Displacing centers leads to non-continuous fusion problems.

Figure 3 Illustrations for several parts of the main lemma's proof.

Since the proposed split-screen might appear in fast-paced games and the human eye is sluggish, some minor violations of the properties in definition 1.4 could be tolerable. The maybe most noticeable violation would be disruptions in the continuous movement and fusion, while a slight deviation in area sizes or centeredness could go unnoticed.

The presented algorithm 1 will utilize a relaxed centeredness condition to achieve fairness, fusibility and a smooth movement. It starts by computing the screen space regions. Collinear (or "almost collinear" for numeric stability) player positions are handled as a special case because their voronoi diagram has no voronoi vertex with finite coordinates. Otherwise the voronoi diagram is computed and moved around until all voronoi cells in the intersection with the screen space have almost equal area. Since we work with a finite amount of pixels in the end, a reasonable error threshold would be 0.01.

Computing the s_i involves the "cheating" and violates the centeredness condition for close-by player positions. The real center is computed and then slightly offset towards the centroid from definition 1.4 scaled by the distance to the other players.

Afterwards fusibility is checked for all regions. If all are fusible, all three players would act in the same window to the game world. Otherwise only two or none are fused. The OpenGL stencil buffer [1] or a similar tool can be used to limit the rendering to an arbitrarily shaped region of the screen space. Rendering the game world is always the same procedure, only translated individually for each region.

4 Conclusion & Room for Lots of Variations

Fairness and centeredness are both very important gameplay aspects. This is the first solution to provide both almost always, and its the first fair split-screen ever. The impossibility to achieve all naturally desired features is shown. The approximation is visually quite close to the (non-existing) optimum and barely interfering with fast-paced gameplay.

Quasi-solutions for more than three players remain a mystery. It seems, that fairness becomes much harder to achieve for more than three players. Maybe fairness and direction-indication together are impossible for a large enough number of players.

Many variations are possible: different kinds of "center", i.e. intentionally misplaced with more visibility in front than in the back, allow different sizes where stronger/faster units have a larger area, relaxed direction-indication condition by a few degrees. Let's discuss…

if *w_0, w_1, w_2 are collinear on line L* **then**
    $S_0, S_1, S_2 \leftarrow$ find two copies of $L^{\perp}$ dividing the screen space into three equally sized parts
else
    $VD \leftarrow$ compute voronoi diagram of w_0, w_1, w_2
    $vv \leftarrow$ sole voronoi vertex of VD
    translate vv (and the complete diagram with it) to $(0, 0)$
    repeat
        $S_0, S_1, S_2 \leftarrow$ cells of $V\cap$ screen space
        compute area sizes s_i of all S_i
        $\mathsf{error} \leftarrow \max_{i=0}^{2} s_i - \min_{i=0}^{2} s_i$
        translate vv by an amount scaled by error in the general direction of largest S_i
    until error *small enough*;
for $i \leftarrow 0$ **to** 2 **do**
    $\mathsf{center}_i \leftarrow$ compute center of S_i
for $i \leftarrow 0$ **to** 2 **do**
    $s_i \leftarrow$ is obtained by linear interpolating between center_i, and the midpoint of center_i and the center of its closest neighbor, weighted with their distance
foreach $(i, j) \in \{(0, 1), (0, 2), (1, 2)\}$ **do**
    if $(S_i - \mathsf{center}_i + w_i) \cap (S_j - \mathsf{center}_j + w_j) \neq \emptyset$ **then**
        mark S_i and S_j as fusible
if *S_0, S_1, S_2 all marked as fusible* **then**
    render world centered at $\frac{1}{3}(s_0 - w_0 + s_1 - w_1 + s_2 - w_2) - vv$
else if *only two regions marked as fusible: S_a, S_b* **then**
    set stencil mask to $S_{\text{non-fusible}}$
    render world centered at $s_{\text{non-fusible}} - w_{\text{non-fusible}} - vv$
    invert stencil mask
    render world centered at $\frac{1}{2}(s_a - w_a + s_b - w_b) - vv$
else
    for $i \leftarrow 0$ **to** 2 **do**
        set stencil mask to S_i
        render world centered at $s_i - w_i - vv$

Algorithm 1: Computing an almost fair voronoi split-screen for three players

References

1 Opengl stencil test. URL: https://www.khronos.org/opengl/wiki/Stencil_Test.

2 Gray Chang. Bumpomov's dogs. Atari 2600, 1983.

3 G. Lejeune Dirichlet. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die reine und angewandte Mathematik*, 40:209–227, 1850.

4 Squirrel Eiserloh. Juicing your cameras with math. *Game Developers Conference (GDC)*, 2016. URL: http://www.mathforgameprogrammers.com/gdc2016/GDC2016_Eiserloh_Squirrel_JuicingYourCameras.pdf.

5 gorsman. Voronoi multiplayer split screen, 2016-05-11. URL: https://www.shadertoy.com/view/4sVXR1.