Map Elites

Initially I used the “basic” version of map elites, with mutation and crossover operations that modify the genome taking into account the concepts of rooms/corridors/etc.

The main issue with this approach is the difficulty (if not impossibility) of predicting the range of values most features will fall into, which often leads in maps being unexplored simply because we chose a feature range that is unrealistic. Some features are also distributed non uniformly in the space of possible maps. For this reason, I moved to Sliding Boundaries Map Elites with good results, since the boundaries are dynamically defined during the run itself.

I also looked at others map elites variants, but none feature the improvement of sliding boundaries and instead solve other problems that we are not concerned with.

Representations

In the following segment we present the various genome representations studied and their characteristic, followed by the result of some comparisons.

All blacks

The same as in other papers and thesis, randomly placed rooms and corridors.

All black has a wide range of possible maps, resulting often in interesting layouts but also very messy ones.

When evolved using "design objectives" (high pace, high entropy, long traces) the maps tend to be of low quality to the human eye (many dead ends, single chokepoints, no loops).

Some examples of “bad” maps:

Immagine che contiene testo, schermata, diagramma, Carattere

Descrizione generata automaticamente Immagine che contiene testo, schermata, Carattere, simbolo

Descrizione generata automaticamente

Using instead “topological” features (number of loops, average mincut) the maps have a more controlled and balanced appearance, although still messy at times.

Some examples:

Immagine che contiene testo, schermata, simbolo, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, diagramma

Descrizione generata automaticamente

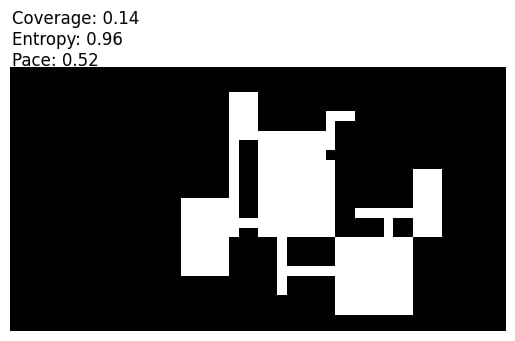
Grid Graph

Same definition as Bari’s. I did some runs with Grid Graph and the main issue is that most maps end up looking very similar. While trying to reduce noise, maps have become too simple.

While some “design objectives” can be explored (although resulting maps are uninteresting), other topological objectives are very hard to optimize/explore (average mincut, number of loops, … examples later).

Example of similar maps from different runs:

Immagine che contiene testo, schermata, Rettangolo, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, simbolo, Carattere

Descrizione generata automaticamente

For this reason, I conducted very few experiments using grid graph and instead focused on other representations.

SMT

The genome is composed by several rooms, defined like in all blacks except for their position, which is not defined in the genome, and several lines, defined by the position of two points. Finally, we have a minimum separation value encoded in the genome. This is because the SMT solver tends to favor solutions with minimum separation between rooms, so if we have set 0 as the minimum separation, we would hardly ever get distant rooms with longer corridors. Still, since the separation can’t be too high or the SMT solver wouldn’t find solutions, rooms are mostly placed very close to each other.

The genotype to phenotype conversion is not deterministic and may fail. The procedure to place rooms and corridors follows that of “Spatial Layout of Procedural Dungeons Using Linear Constraints and SMT Solvers”. For corridors, first we connect rooms with a MST to ensure connectivity, like the paper does, then to augment connectivity we also connect rooms that are directly connected by the lines (corridor placement is deterministic given two rooms to connect). This results in maps that “resemble” the lines somewhat.

A few examples (rooms in red, corridors in blue):

Immagine che contiene testo, diagramma, linea, Parallelo

Descrizione generata automaticamenteImmagine che contiene testo, diagramma, linea, schermata

Descrizione generata automaticamenteImmagine che contiene testo, diagramma, linea, schermata

Descrizione generata automaticamente

Initially, the representation had problems with failures (more than 10% of the genomes would have no feasible SMT solution, thus no phenotype) which would lead to limited exploration and many similar elites.

By making the constraints less strict and modifying the mutation and crossover operators now the failure rate is between 2% and 5%.

The maps themselves seem to allow for great variety and since we connect rooms directly the result is clean, and archive dimensions across various runs show that the exploration is successful.

Point

This new genome is inspired by “Interactive Evolution of Levels for a Competitive Multiplayer FPS”. The genome has rooms that are identical to those of All Blacks, while corridors are replaced with couple of points (that represent the extremities of the corridor) and an “orientation” value, that determines the shape connecting the points (whether the “L” shape is upright or sideways).

This genome was interesting since it led to better results than AB on some tasks, but I quickly moved away from it in favor of a new variant: the main problem with this representation was that rooms and corridors were completely unrelated, while in all blacks singular corridors are short, here they can be very long, so we often produced long corridors with no rooms on either side.

Examples:

Immagine che contiene testo, simbolo, schermata, Carattere

Descrizione generata automaticamente Immagine che contiene testo, schermata, Carattere, Rettangolo

Descrizione generata automaticamente

Point AD

Variant where rooms are tied to the extremities of the corridors, those points act as center of the rooms, and we encode the “radius” of the room starting from the center. Corridors may have no room at either side (radius 0).

Examples:

Immagine che contiene testo, schermata, Rettangolo, diagramma

Descrizione generata automaticamente Immagine che contiene testo, simbolo, Carattere, schermata

Descrizione generata automaticamente

With respect to AB, we note that each room will almost always at least be connected to another room, while in AB a room is connected to another room if only by chance a certain number of corridors happens to be put in just the right way for them to connect. I argue that this impedes the ability to effectively explore the space of AB genomes, and thus that this new representation should lead to higher QD score of an archive. This looks to be true but not for all fitness considered. More on this later.

We also note that in general maps produced by this genome tend to be more complex and allow for longer corridors than those of SMT while avoiding being messy like AB.

Comparisons

Several experiments were conducted using multiple representations to compare results. Particularly, I looked at the Quality Diversity score, the maximum fitness and the archive size. Following are some examples to highlight differences between representations which are not exhaustive but meant to give some idea of the differences of these representations. All experiments try to fill a 10x10 sliding boundaries grid.

Experiment 1: *Fitness: pace | Features: average traces, fight time*

|  |  |  |
| --- | --- | --- |
| AB | GG | SMT |
| ***Immagine che contiene schermata, Policromia, testo  Descrizione generata automaticamente*** |  |  |
|  |  |  |

Archive size was around 100 for all representations, and max score about 0.95 for all.

So, on “design objectives” such as pace, grid graph can attain results comparable to AB, while SMT underperforms. It is worth noting that the maps of GG are rather simple, so pace is an easy objective to optimize since it favors simple maps. On the contrary, SMT maps tend to be not as simple by default.

Experiment 2: *Fitness: average mincut | Features: average length of cycles, std deviation of room betweennes (low => rooms are equally “in between”, high => rooms are scattered)*

|  |  |  |
| --- | --- | --- |
| AB | GG | SMT |
|  |  |  |
|  |  |  |
| Archive size: 100 | Archive size: 40 | Archive size: 85 |
| Max score: 4.0 | Max score: 3.0 | Max score: 3.0 |

With “topological” features the main limitations of grid graph become apparent; the range of possible maps is simply not big enough to explore different layouts, leading to massive underperformance. SMT also still underperforms AB in this scenario.

Note: in these first two experiments SMT had not yest been tweaked, leading to the problems explained above.

I then tested SMT again on this second task after tweaking it, getting results similar to AB.

Immagine che contiene schermata, Policromia, testo, diagramma

Descrizione generata automaticamente Immagine che contiene testo, diagramma, Diagramma, linea

Descrizione generata automaticamente

With max score of 5.0 and archive size of 90.

Experiment 3: *Fitness: Balance Topology (feature I tried to create to express how balanced a map is topologically, more on this later) | Features: diameter (max distance of two rooms), symmetry*

|  |  |  |
| --- | --- | --- |
| AB | *Point AD* | *SMT* |
|  |  | *This heatmap may seem very different but note that a single outlier at 0,0 is making it look like it. They are very similar in terms of ranges.* |
|  |  |  |
| Archive size: 98 | Archive size: 100 | Archive size: 100 |
| Max score: 3.9 | Max score: 4.0 | Max score: 4.0 |

Point AD significantly outperforms both AB and SMT, while we also note that SMT outperforms AB in this scenario. This balance measure takes into account the mincut, the standard deviation of the room distance and the number of loops, so it is entirely possible that other measures would see AB outperform PointAD. For example, in another experiment optimizing sight loss rate, AB outperformed Point AD, likely because AB produces messier maps where losing sight of the enemy is easier.

Metrics

Following are all the metrics that we extracted. Due to the high volume of metrics, I could not test them all. I focused on testing “unique” measures (some metrics are different but likely to produce similar maps, or have similar exploration capabilities, so I only tried some of those. E.g. top distance and average distance)

Match metrics

Match metrics are directly collected by the framework. I only included those tested and used for this category.

* pace
  + Low pace maps are labyrinthine while high pace maps are smaller with large rooms and small corridors, often just squares.
  + Illuminating pace seems to give rise to many different types of maps.
  + When optimizing for larger, well-connected maps (fitness was explorationPlusVisibility, explained below) we get interesting results. It is hard to obtain high pace, but we do get
* fightTime
  + Strictly correlated to pace. High fight time means small maps.
* sightLossRate
  + Kind of opposite results compared to pace. Although maps tend to be messier rather than bigger.
  + While a high sightLossRate may be beneficial to gameplay, it does not correlate well with map visual quality.
  + When illuminating it, similar maps can get completely different results. This puts into question the reliability of this measure, as it may depend a lot on random bot behaviour.
* entropy
  + When using balanced bots, the number of maps with an entropy of 1 is large, it is not difficult to obtain most solution in the archive having 1, or >0.9 as for entropy.
  + No distinctive characteristic of the maps is elicited by illuminating entropy, this means that while we could get balanced maps, we have no information strictly on their quality.
  + When looking to balance agents of different skills, entropy can be very relevant.

Extracted metrics from match data

Starting from data such as positions, kill positions and death positions we derive an heatmap by dividing the map in a grid and assigning a score to each cell dependent on how many times it appears in the data. We then apply a gaussian filter before extracting the following metrics.

For traces we do not consider an heatmap but their length instead.

* **Position Heatmap**
  + maxValue
  + localMaximaNumber
    - High values ensure big maps (since there are more possible spots for local maxima to form) but not necessarily a good coverage. Map quality still varies vastly, and many dead ends exist.
  + localMaximaTopDistance
  + localMaximaAverageDistance
    - Doesn't seem to elicit any particular map characteristic.
  + averageLocalMaximaValue
  + stdLocalMaximaValue
  + quantile25
  + quantile50
  + quantile75
    - Leads to maps being explored almost always fully when optimized, but in contrast leads to much smaller maps than "coverage", sporting the same benefits.
    - Results are very similar using any quantile, except that higher quantiles are harder to get on bigger maps, so we get smaller maps on average
  + coverage
    - Coverage means the percentage of the walkable map that has a value greater than zero, in this case, it means the percentage of the floor roughly covered by the bots.
    - High percentages are ideal, since we get maps that sport fewer "useless" features such as dead ends, and more connected maps (since its easier for the bots to navigate those). I would still argue that if the objective is to obtain good maps, then topological features are better to define what we consider to be a good map, while here we only look for well explored maps and hope that they are also well-made.
    - Optimizing only this feature may lead to smaller maps, since those are obviously more easily explorable by bots, especially small, squared rooms. So, if using it for fitness, we should either punish maps too small or illuminate features that explore maps of different dimensions (average traces, local maxima distance, room betweeness, ecc.)
* **Kill Heatmap**
  + maxValue
    - High max value often relates to less local maxima, while a low max value could be either due to the fact that kills are spread across multiple spots, or because not many kills were done in the first place.
    - Either way, it is hard to spot differences in maps with high max value and low max value.
  + localMaximaNumber
    - Not always relevant. While it is true that a small number of local maxima of kills leads to a small map, with a singular checkpoint, it may still happen that a map with many local maxima may still have a singular maximum much bigger.
  + localMaximaTopDistance
  + localMaximaAverageDistance
    - By illuminating distance of local maxima in the heatmap of kills, we get maps that range in how far the most kills happen.
    - Low values lead to small maps with either a single hotspot or several close, likely in the same room. On the contrary, high values lead to big maps.
    - Having large distance does not mean not having chokepoints
  + averageLocalMaximaValue
  + stdLocalMaximaValue
  + quantile25
  + quantile50
  + quantile75
  + coverage
    - This is a feature that is hard to illuminate, leading to maps being very small to allow for most of the ground to be covered in kills.
    - Does not seem to be all that usable. Even when paired with average traces, which should elicit also bigger maps, the result was still lackluster.
* **Death Heatmap** (Basically the specular of the kill heatmap, so I didn’t explore it and focused on kills)
  + maxValue
  + localMaximaNumber
  + localMaximaTopDistance
  + localMaximaAverageDistance
  + averageLocalMaximaValue
  + stdLocalMaximaValue
  + quantile25
  + quantile50
  + quantile75
  + coverage
* **Traces**
  + maxTraces
  + averageTraces
    - The average length of kill traces by both players.
    - Higher values should lead to longer corridors or big rooms where the bots can shoot each other from far apart.
    - Maps resulting are as expected, we also get maps with interesting and long line of sights, which makes this feature interesting for getting map design that elicit these long trajectories.
  + quantile25Traces
  + quantile50Traces
  + quantile75Traces

Map analysis

These metrics are extracted by the topological feature of the map phenotype.

Graph metrics are obtained by finding a graph representing the map with the approach in *“Terrain Analysis in Real-Time Strategy Games: An Integrated Approach to Choke Point Detection and Region Decomposition”*, and then analyzing the graph.

Visibility metrics are obtained by crafting a visibility graph; each walkable cell is a node, and an edge is added when the two tiles have a clear line of sight. We check it by drawing a DDA line between two cells and ensuring that no wall is on that line. We then calculate a cell visibility in percent as the number of edges over the number of walkable tiles.

Symmetry is checked horizontally and vertically by asserting if a cell is walkable on both halves of the map. The result is a percentage of walkable tiles that are symmetrical.

* **Graph** **metrics**
  + roomNumber
    - Useful metric since rooms in the genotypes often become fused into bigger rooms. This number is instead related to the graph nodes that are identified as rooms by the algorithm.
    - Since the algorithm uses some heuristics to determine and merge rooms, it’s not always the most accurate
  + averageRoomMinDistance
    - Interesting feature to illuminate. Close rooms often mean small maps, but we also explore big maps with very close rooms.
    - It is well illuminated overall, but I still would have expected some maps with longer corridors. Could be the fault of the fitness used the avoided searching that part of the grid.
  + stdRoomMinDistance
  + averageRoomRadius
    - Room radius is the distance to the closest wall. Higher radius relates to bigger rooms. It is calculated by the algorithm that creates the graph.
    - Interesting measure to illuminate. For example, when optimizing maps with a balanced topology looking at different average room radiuses, we can see different interesting designs
  + stdRoomRadius
  + averageChokepointRadius
    - Like room radius, but for nodes that represent chokepoints. This measure is not that useful, since most chokepoints are going to be corridors which have a fixed width, meaning that different radiuses are hard to illuminate
  + stdChokepointRadius
  + averageRoomBetweenness
  + stdRoomBetweenness
    - For lower values, we expect all room to be similarly "in between", meaning that we expect maps with close and well-connected rooms. Higher values mean that there will be rooms with varying values of betweenness, so we will have farther away rooms and generally bigger maps. This works better than closeness, and we can see clearly in the results the effects.
  + averageRoomCloseness
  + stdRoomCloseness
    - Similar to betweenness, but for some reason it’s harder to explore and lead to similar maps.
    - We expect lower values to represent maps where all room have a similar closeness rating, leading to closer rooms. This metric does not seem to represent actual closeness well however, and better results can be obtained with betweenness.
  + AverageMincut
    - In my opinion of the most crucial characteristic of a map.
    - Lower values correctly represent maps with chokepoints corridors, which are reflected also in the kills/death heatmaps. Higher values correctly represent more connected maps with many escape routes.
    - Compared to improving the sightLossRate, maps are far more interesting and still sport escape routes.
  + stdMincut
  + maxMincut
  + minMincut
    - Hard to explore, most maps will have usually have either 1 or 2 as the value.
    - Giving fitness a boost if minMincut is at least 2 can be a good idea to favor well connected maps without necessarily favoring maps with high average mincut.
  + averageEccentricity
    - High eccentricities mean high distance between rooms. This relates to maps with long corridors, often labyrinthine, since it makes paths longer on average.
  + stdEccentricity
  + diameter
    - Similar results to average eccentricities, but since it only counts the highest distance, it is prone to maps that look not that big but have a single long path between rooms at the extremities.
  + radius
  + periphery
  + peripheryPercent
    - Percentage of rooms considered in the periphery of the graph (eccentricity of the room almost like the graph diameter)
    - Conceptually an interesting measure that should distinguish maps with a lot of peripheral rooms to those without
  + center
  + centerPercent
    - Specular to periphery, includes rooms with eccentricity almost like the graph radius
  + density
  + numberCyclesOneRoom
    - Interesting measure to illuminate. Different representations can illuminate this measure to different degrees.
    - Maps with loops are generally more competitively interesting.
  + averageLengthCyclesOneRoom
  + stdLengthCyclesOneRoom
  + numberCyclesTwoRooms
    - Like one room version, but at least 2 rooms must be on the same loop for it to be counted
  + averageLengthCyclesTwoRooms
    - Interesting feature to illuminate. We expect bigger cycles to mean longer escape routes and longer corridors. This hold mostly true.
    - For AB the effectiveness may depend on how rooms were formed and composed. However, clearly higher result can be seen clearly in bigger cycles in the maps, as expected
  + stdLengthCyclesTwoRooms
* **Visibility** **metrics**
  + maxValueVisibility
  + maxValuePercentVisibility
  + averageValuePercentVisibility
    - High values mean that on average a spot can see a lot of parts of the map. This results in open maps with few corridors.
    - Interesting to illuminate to get varying types of maps
  + stdValuePercentVisibility
  + localMaximaNumberVisibility
    - Each local maxima corresponds to a zone in the map that is important for gameplay
    - Illuminating the number of these zones leads to maps of different complexity and with different features.
    - Since a high visibility spot is closely correlated with the emergent gameplay, this is in my opinion a good metric.
  + localMaximaTopDistanceVisibility
  + localMaximaAverageDistanceVisibility
  + averageLocalMaximaValuePercentVisibility
    - Varying how much visibility “important spots” on the map have is an interesting metric, although it does not lead to extremely interesting layouts by itself.
  + stdLocalMaximaValuePercentVisibility
  + quantile25PercentVisibility
  + quantile50PercentVisibility
  + quantile75PercentVisibility
* Symmetry
  + xSymmetry
  + ySymmetry
  + maxSymmetry
    - Maximum of the two above.
    - Hard to maximize, but high values clearly relate to more eye pleasing symmetrical layouts.

“Constructed” metrics

The following metrics are an attempt to mix different metrics to describe a certain characteristic of a map.

I believe that fitness functions of this kind would allow to produce maps that are more fun to play on average, since it will not rely simply on luck for certain characteristic to explicated (e.g. a balanced (high entropy) map with looping topology).

* balanceTopology
  + This measure is meant to give a higher score the more “balanced” the topology is, meaning that rooms are eqaually distant, well connected and present loops.
  + The formula is the following
  + The idea is that a map is balanced if: rooms are at a similar distance from one another (30 is the empirical average of the stdRoomMinDistance, so we want a standard deviation of the distance as close to 0 as possible, meaning all rooms are qually distant), the average mincut is at least 2 and the minimum mincut is 2 (no single chokepoint), and if there are at least 5 cycles between two different rooms
  + We clip because too many cycles or an average mincut too big may lead to maps that are overcomplicated and would be very hard to navigate and remember for a human playing them
  + Resulting maps are of good gameplay value, although a bit too similar. Results also vary a lot depending on which measures we illuminate.
* explorationPlusVisibility
  + This measure is meant to score how much a map is easily explored while having few low visibility spots
  + The formula is the following
  + The idea is that below 1.7 average mincut the maps tend to be less easy to explore, so we punish those maps, while we want maps that have at least 5 “visibility spots”, to avoid maps with a single big room with only one or two spots that cover it. However, we also want the average visibility to be high, to avoid hard to see spots.
  + Resulting maps are interesting from a gameplay perspective, although mostly too similar to one another, with big arenas and few walls in between.