Interactive Evolution Levels for a Competitive Multiplayer FPS

Making players do the level designer's job

Author:
Peter Ølsted
IT University of Copenhagen
ptoe@itu.dk

Author:
Benjamin Ma
IT University of Copenhagen
benm@itu.dk

Supervisor:
Sebastian Risi
sebr@itu.dk

December 1, 2014

This page is intentionally left mostly blank

Abstract

Traditionally a dedicated level designer has to design every aspect of a level by hand, distribute the level and gather people to test it. Especially for a multiplayer map, the cyclic process of developing and testing can be quite cumbersome and time consuming. We present a novel approach to provide live generation of levels using procedural level generation and interactive evolution. Without leaving the game, a group of players can generate, play and improve levels to fit their preferences. This approach focuses on maintaining the level design principles for modern first-person shooters that encourages good engagements between players, using the game mode of 'bomb defusal'. We show that our approach can generate high quality levels that adapt to the preferences of players across a wide range of skills. Our approach created several distinct types of levels during testing. These types of maps range from open to closed, and complex to simple, that each fit closely with what the different players consider a good map.

This page is intentionally left mostly blank

We would like to thank our supervisor Sebastian Risi for his support on this project.
The inverted mouse control in the game is dedicated to him.

Contents

1. Introduction

1.1 Background & Introduction

First-person shooters have seen an increased popularity over the years and especially on the competitive scene, where people have embraced the fast and precise gameplay. However having poor level design can quickly turn off interested players. Developers often attempt to provide as much content as possible in multiplayer modes to keep players engaged, but designing even a single level can be a time consuming task. After the level has been mocked-up in the game, user feedback needs to be gathered before building the level in its final state. Target playing styles varies greatly for different levels, and by designing for all styles at the same time, a designer might end up prioritizing differently than the player might prefer.

Procedural Content Generation (PCG) is a technique for creating new content as the game is played. It can either be created before the game starts, at it starts or when needed as the game progress [TSN14]. Content can be a lot of different elements, it can be everything and range from new assets, like textures and sounds, to entire maps and even new game mechanics[Smi]. While Procedural Content Generation can quickly generate a lot of content, the generator usually has to be made for the specific target application.

Interactive evolution is an approach that uses humans to decide the direction of the evolution. Instead of the computer selecting which entity gets to produce offspring and which die, the human chooses. This is useful in situations where the properties of the content cannot be expressed as a single fitness value, like subjective beauty [LSS14].

In this project we will look at generating multiplayer levels that tie in with the state-of-the-art gameplay and level design of games like Counter-Strike[Cor04][CE12] and Call of Duty[War07]. Using PCG and interactive evolution, players are able to affect the evolution of levels by picking the next step in the evolution. PCG techniques are used to generate the levels and the evolution is used to adapt them. Evolutionary techniques usually require long waiting times and many iterations due to the nature of evolution, with small random mutations and multiple environments that need to be simulated (Ølsted & Ma [lM12]). This project attempts to make substantial changes in few iterations using the players as the final judges of fitness, rather than relying solely on predefined algorithms.

Multiplayer games face some unique challenges compared to single-player games.

Where single-player games use the game mechanics to explore the content of the game alone, multi-player games allow for interesting social interactions as it provides a context for human interaction[Juu11]. Multiplayer gameplay is often acted out on the same maps repeatedly, while single-player contains a narrative that tends to be played only once. This allows players in multiplayer games to become highly familiar with the maps and to predict the movement of the opponents, which in turn rewards strategic planning. Level design that encourages this familiarization and planning is a key component in multiplayer FPS games and a part of we describe as a good engagement. Interesting and balanced levels is another important aspect, which unfortunately is an aspect of level design where novelty can quickly wear off, and where procedural generation can be utilized. Generating new content however does interfere with the players familiarizing themselves with them. Instead of procedurally generating completely new levels all the time, our approach uses interactive evolution which can develop new and unique levels that can remain both as fresh and familiar as players see fit.

1.2 Project Description

The purpose of this thesis is to build and evolve levels for a simple FPS (First-Person Shooter) using procedural content generation, and to enable playtesting and the evolution of a level as soon as it is generated.

The goal is to quickly create good level layouts, that can be immediately tested in a multiplayer environment, without ever presenting the players with any level creation tools. Level generation is nothing new, but even a good generator will not always create something satisfactory for every crowd. A key challenge is how to iterate, improve and balance the generated content to satisfy the audience and hone in on their preferred layout. To accomplish this is, it important to understand current developer-created level design so it can best be recreated as part of the procedural content generation and interactive evolution.

1.3 Research Questions

- How can levels be generated and evolved to fit with the gameplay of a multiplayer first-person shooter?
- Can players in a group guide the evolution to fit their preferences?
- How well did these levels evolve and adapt to players?
- Are these levels satisfactory to the audience?

1.4 Method

Focusing on a single game mode, we will research the level design of multiplayer FPS levels, and define a set of rules and constraints that these levels all must adhere to. Using that knowledge we will attempt to develop a solution to generate and genetically evolve levels suited for that particular game mode. We will look at how a group of players as a whole can control the evolution of the level. Finally we will assemble all of this knowledge into a playable prototype, so it can be tested by people. This should include first-person shooter gameplay as well as networked multiplayer, with the abilities for players to quickly and efficiently impact the design of a map as a group.

1.5 Glossary

The Good Engagement (TGE): The engagement between two or more opposing players, see section 3.4.2. A good engagement has 'interesting' choices a player can make to gain an advantage against the opponent.

Unity: The game engine used. See www.unity3d.com

Map/Level: The playing field, in which a game session is played out. Used interchangeably.

First Person Shooter (FPS): A game genre in which the player sees the world from a first-person perspective, while equipped with a gun or similar weapon. Notable games include Doom, Counter-Strike and Call of Duty.

Game Mode: The ruleset used when playing on a map. Depending on the game, levels can support different game modes. Examples include 'Deathmatch' and 'Search & Destroy', see below.

Search & Destroy (S&D): A game mode with two opposing teams. The attacking team has to place a bomb, while the defending team has to stop the attacker team or defuse the bomb. Known as *'Bomb Defusal'* in Counter-Strike. See section 3.1 for more detail.

Deathmatch: A simple game mode where the winner is the player who kills the most opponents. There are no teams and the player re-spawn immediately after being killed.

Spawn/Respawn: Refers to a player when they are killed are placed back in the game. Depending on the game mode, the player may have to wait until the next round begins.

Spawn Point: The position from which players will start or respawn on a given Map/Level. Usually teams have dedicated spawn points.

Bomb Point: In the game of Search & Destroy, this is where one of the two opposing teams is tasked to position a bomb, usually at one of two locations.

Choke Point: Also known as bottlenecks. The points at which two opposing forces are likely to meet each other. See section 3.4.1

Genotype: The raw *DNA* like sequence of values from which the map is built.

Evolution: “The gradual development of something”[1]. Often used in biology. Here used for the gradual development of the maps.

Mutation: A random change to the genotype during evolution when a parent produces one or more*'children'*.

Interactive Evolution (IE): Human guided evolution, used to create maps in this paper.

Fitness: The evaluation of a map according to certain criteria. Usually used to evaluate the quality of a map.

Generation: A set of children evolved from a preceding generation. In this paper a new generation of maps is created from a single parent map.

Co-op: A multiplayer game where the players have a shared goal and a common enemy where cooperation is needed to win. Harming other players is impossible.

1.6 Section Overview

The project is divided into nine chapters.

The first chapter introduces the project, its background and defines the glossary used. Chapter two gives an overview of the previous work done in procedural map generation and both mixed & interactive evolution techniques. Chapter three gives an overview of modern first-person shooter level design along with our definition of The Good Engagement.

Chapter four is shortly describing the different prototypes developed and what we learned for building the final algorithm which is specified in chapter five. Chapter five introduces the process used. It overviews the different prototypes developed and what we learned from them. It includes an in-depth look at the techniques used to develop the maps and various other considerations for building the project. Chapter six shows the results gathered from the three playtests. It goes through the findings of the questionnaire and the notes from the playtests. Chapter seven discusses the results and analyses the players reactions and responses. Chapters eight & nine respectively suggest ideas for future work and conclude on results of the project.

[1] http://www.oxforddictionaries.com/definition/english/evolution

2. Previous Work

This chapter describes the previous work that this project builds upon. We give an overview of the goals, important findings and how the projects described inspired this project.

2.1 Inspirational Work

2.1.1 Evolving Interesting FPS levels

Cardamone et al [Car+11] discuss generating interesting multiplayer FPS levels using computer agents. They developed four different algorithms for level generation and ranked them with a fitness based on deathmatch played by four computer bots. In their own words, a “simple theory-driven fitness” was used that linked “the fighting time of the player” directly to the fitness. Thus, the longer a bot survived after it engaged an enemy, the higher the fitness. See figure 2.1 for the generated level with the highest fitness.

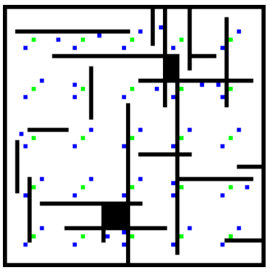


Figure 2.1: The highest rated level from Cardamone et al [Car+11]. Blue points being spawn positions and green points being resources like health or weapons. This is an example of an All-White map, where walls are added as obstacles.

Comparing the levels to what we define as The Good Engagement, see section 3.4.2, there is a clear misalignment between the generated maps and what we define as a good engagement. The level presented contains dead ends on the right hand side of the map and contains no areas more dedicated to engaging opponents than others. We present an alternative solution to the generation of multiplayer FPS levels that focuses on the funnelling of players and their engagements.

All-Black vs All-White

A key concept in the paper by Cardamone et al[Car+11] is the introduction of the two terms All-Black and All-White which refer to the initial states of map generation. Black refers to the area where players cannot traverse and white refers to the area player can walk. Thus a completely black map is unplayable as there is nowhere the player can walk since all space is occupied, while an entirely white map is playable but completely open and empty.

Hence All-Black refers to a map that is completely untraversable where traversable, white, areas are carved out. Oppositely in an All-White map, untraversable obstacles are added.

In general All-Black maps tend to closely resemble a set of rooms and corridors between them, while All-White maps tend to be open spaces with objects placed in them to limit visibility. These concepts can also be applied to existing games like Call of Duty and Counter-Strike as their maps in most cases can be classified in one of the two categories. For example does Call of Duty maps tend to look like All-White while Counter-Strike maps tend to resemble All-Black. These two approaches would both be utilized during the development of the procedural level generator, each with pros and cons which are described in chapter 4.

2.1.2 Mixed-initiative

Liapis et al [LSS14] give a broad overview of a number of projects which all involve some level of design created by a collaboration between a human and a computer. In short, mixed-initiative means that the player as well as the computer are able to build levels. Both computer and human must have a significant impact on the design. Games like the Civilization series, where the player can only adjust simple settings like the size of the map, are not considered mixed-initiative. As humans must have a significant part of the design, it can potentially be quite time consuming compared to interactive evolution. As we found in the early prototypes (section 4.1.2), direct influence resulted in a very chaotic and non-structured work. An example of mixed-initiative mentioned is the Sentient Sketchbook which is described in more detail below.

2.1.3 Sentient Sketchbook

The sentient sketchbook [LYT13] is a tool to help game designers create levels. The designer can sketch different elements onto the map like resources and non-walkable areas. Figure 2.2 displays the interface of the 'sentient sketchbook'.

To help the designer, the sketchbook can visualize different properties of the map like navigation-mesh[1], unused space and how safe the games resources. This helps the designer iterate quickly as no manual work is needed to produce these overviews. The sketchbook can visualize the level in different settings like nature, dungeon and waterways. See figure 2.3 for some of the visualization it can produce.

[1] A common technique to help game character pathfinding.[CS12]

In the user study section of the paper, Liapis et al found that “surprisingly, suggestions targeting balance were not chosen often;” while suggestions that increased the *'Exploration score'* were often chosen as they eliminated manual work. This was an important observation that made us opt for always generating balanced maps, rather than letting it be up to the user to make interesting maps while also considering game balance.

2.2 Interactive Evolution

Allowing the player to impact the evolution, but not directly build or adjust values, is interactive evolution (IE). While values can be modified indirectly, IE does not offer the same level of control as mixed-initiative. With less control, evolution can generate novel results the designer did not expect. On the other hand evolution requires a way to rate generated content in comparison to other generated content. A fitness function summarizes all aspects into a single number and when generating a level, it can be hard to summarize all facilities in a single number. Many aspects of a level is neither positive or negative, but some aspects may be favoured by different players. As such IE can utilize humans evaluation skills to direct the evolution, where a classic fitness function could not.

2.2.1 Picbreeder

Picbreeder.org [Sec+08] is a website on which users can collaborate to create pictures via evolution. To generate a picture a user can either use a new randomly generated structure or continue from a previously generated picture another user has saved. This overcomes problems with user fatigue where other users can resume any previous work and branch out at any point to evolve new pictures. Figure 2.4 display the user interface of Picpreeder. The user can select any number of the shown offspring to produce the next generation. It also contains a slider setting how big or small the changes should be for the next generation of pictures.

Generated pictures often increase in complexity as users become more specific in their goals and end up generating pictures that resemble cars, animals and abstract figures among others. Several offspring can be selected to breed the next generation, allowing users to try and combine their favourite properties of multiple offspring. This is an example of an approach where a fitness function would not work as there is no way to quantify if a picture is better than all other pictures.

Picbreeder has a user friendly approach to interactive evolution. It can be used with little to no introduction and was used as the base for the voting interface in this project. A drawback with this method used is that the process of evolving a picture takes tens, if not hundreds, of generations of carefully selected pictures, before desired results can be created.

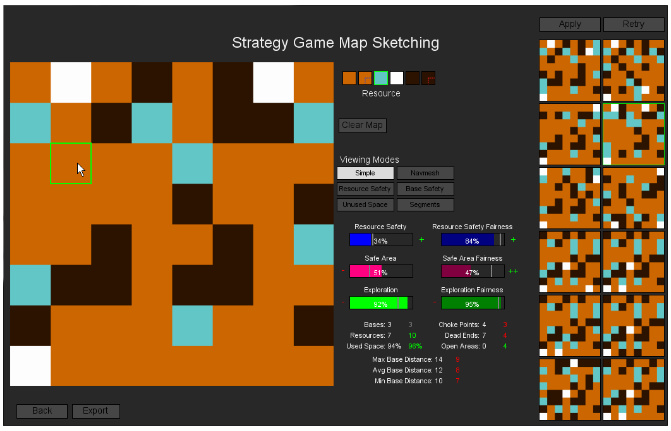
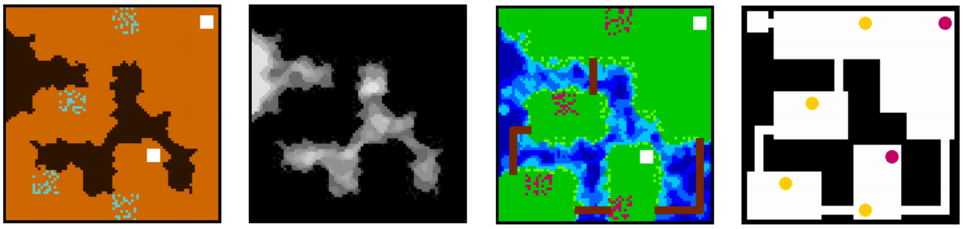


Figure 2.2: Sentient Sketchbook interface. The designer to the left, evaluation of it in the middle and suggested new maps from the current sketch. Koutník et al [LYT13]



(a) Default (b) Heightmap (c) Waterways (d) Dungeon

Figure 2.3: Different types of output from the the Sentient Sketchbook. Each suitable for a different purpose or game.

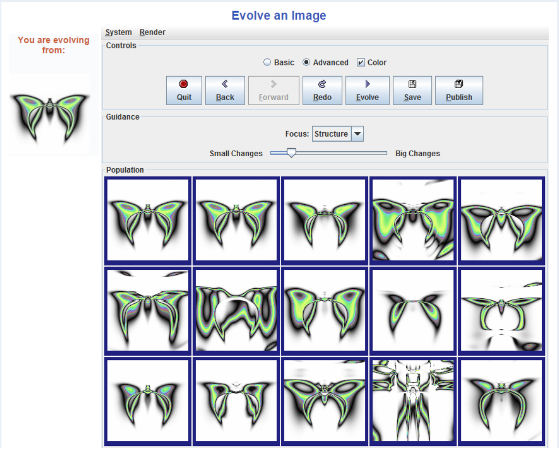


Figure 2.4: Picbreeder interface. It shows the parent in the upper left and a set of evolutions the user can select to generate the next generation of pictures.

2.2.2 TORCS

TORCS [CLL11a] is an open-source car racing simulator for which an IE track generator was developed, where players rate the levels generated as they evolve. It has an interface where you can visualize the best levels along with their rating, and users can interact by voting on the quality of levels. Cardamone et al [CLL11a] tested the system using single users with 5-star ratings, but made some preliminary tests with multiple users, using an up- and downvote system, that both showed an increase in the ratings for the levels as users kept evolving levels, which ended up inspiring our own voting system.

2.3 Level-generators

As it can be seen in section 2.1, limited work has been done on procedural multiplayer level design. For singleplayer and co-op multiplayer games however there are commercial games that utilizes PCG for their content. This includes dungeon generators, online co-op games and titles like *'Warframe'*[Ext13], *'The Binding of Isaac'*[Gam11] and *'Spelunky'*[Yu08]. These projects gave inspiration for possible techniques that could or could not be used.

2.3.1 Formal Language Based

Formal Language is a set of symbols or strings which are constraint by a rule-set [Sak]. A common known example is 'Context-free grammars' (CFG) like 'Backus–Naur Form' (BNF) which is often used for parsing programming languages and has been extended to parse regular expressions.

In 3D, formal language is often used to generate tree- and plant-like structures as their recursive branching features are well suited to generate realistic plants [Pik07]. CFG has been used as an experimental tool for generating procedural levels for the video game *'Hitman Absolution'*[Int12]. Mousten & Ermacora [ME13] produced fully playable levels for the game using a CFG and a Monto-Carlo search to connect map templates to generate the levels. An example of a generated level can be seen in figure 2.5. While this approach generated playable levels, they are always built in a tree structure. This structure would not fit with a multiplayer map as it should not contain dead ends. Generating non-tree-like levels using a formal language is possible and could be an interesting approach. We decided however not to pursue this approach based on the difficulty in making a cyclic grammar and 'unfolding' them into valid three dimensional maps.

2.3.2 Templates

A popular approach for generating procedural levels is stitching templates together in a way that suits the purpose of the game. A template is a pre-build section of a map which can be stitched together like Lego. 'Warframe', 'The Binding of Isaac' and 'Spelunky' are commercial successfully games based on this approach. Warframe is an 3D online co-op third person shooter game where the players fight through hordes of enemy NPCs.

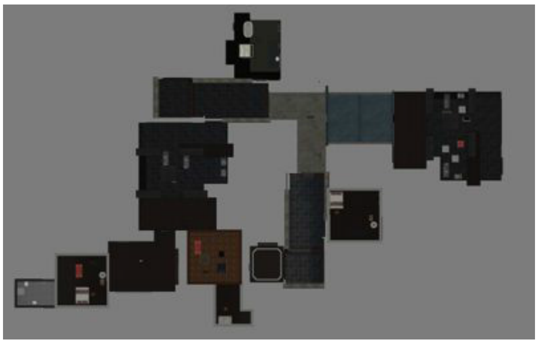


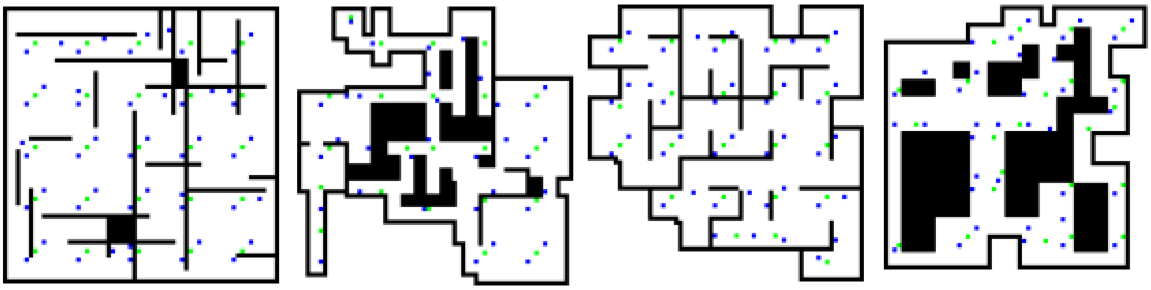
Figure 2.5: An example of a generated Hitman level. Note the tree structure of the level.

The Binding of Isaac is a 2D singleplayer game where the player plays as a child battling through your avatar's mother's basement, that is constructed as a tree structure. Spelunky is 2D a co-op game where the players excavates and fights through five different sections each with three to four levels.
This technique can generate a lot of variations, but as they are all based on the same level segments, template repetition can quickly appear. 'The Binding of Isaac' has more than 1000 level segments each with additional random elements inside them to avoid this problem[McM11]. 'Warframe' contains ten unique environments and seventeen mission types to keep content varied[Ext]. Map designs using templates is restricted to how they can be stitched together, novelty thus partially depends on quality and quantity of the templates. We chose not to use a template approach due to the high number of required templates for good and varied levels.

2.3.3 Evolving Interesting Maps for a First Person Shooter

As mentioned in section 2.1.1, Cardamone et al [Car+11] described four different approaches to generate FPS levels. The names of the types are All-White, All-Black, Grid and Random-Digger in the same order as presented in figure 2.6. The fitness of the maps is determined by the fighting time of the agent, which is how long the agent survived after the first encounter with an enemy. Using bots, they simulated a ten minutes match in ten seconds and simulated 50 generations with a population of 50 in each. Four bots were simulated using the 'deathmatch' game mode where the goal is to kill the highest amount of opponents.

As can been seen the presented maps, they contain several undesired features such as dead ends, long corridors and often a lack of choices in navigation. More details about what we deem as a good map can be seen in chapter 3. Some of these seem to stem from the fitness as survival time indicated a better fitness, as such changing the fitness and making human plays instead would probably steer evolution another direction. While the approaches presented here fit with a 'deathmatch' game mode, we did not deem them suited for the team based game mode used in this project. Not using



(a) $f = 22882$ (b) $f = 15801$ (c) $f = 22520$ (d) $f = 13202$

Figure 2.6: The highest ranked maps generated by each of the four approaches. The number under each level is the calculated fitness of the maps. The approaches are in order All-White, All-Black, Grid and Random-Digger.

bots is another difference. We deemed that bots do not yet have the ability to properly reflect human team based behaviour, despite the work of projects like the 2k BotPrize competition[Hin10] that tries to model human behaviour in bots.

3. Exploring Key Principles in Good Map Design

In this chapter we look closer at first-person shooter games, and in particular the Search & Destroy game mode used for this project. This analysis will feed into the generation of levels and aid us in creating a suitable fitness function for which levels to present to the players. We introduce the term *'The Good Engagement'* (TGE) and how it is used in various games. In conclusion to this chapter we relate to *'The Good Engagement'*, what it entails and how our level generator is affected by these observations.

In this chapter we will explore some of the key principles in 'Counter-Strike' and 'Call of Duty'. Both are among the most popular FPS games.

3.1 Search & Destroy

Before introducing TGE properly, we introduce the game mode used in this project. *'Search & Destroy'* is a game mode where an attacking team of players has to place a bomb which the defending team has to defuse. When killed, players do not spawn before one team wins. This mode is known as 'Bomb Defusal' in Counter-Strike. The bomb is by the start of the round given to one player on the attacking team and has bomb is placed in one of two predefined bomb points. If the player carrying the bomb dies, a player on the same team can pick it up. If the attacking team successfully plants the bomb, they have to defend it for a period of time, until the bomb explodes, while the defending team has to defuse the bomb. If the bomb is not placed and all players on a team are killed, the surviving team wins. If the attacking team places the bomb and then gets eliminated, the defending team still has to defuse the bomb before it explodes, to win. After a few rounds, the two teams are swapped, making the attackers the new defenders and visa-versa.

3.2 Counter-Strike

Released as a mod for Half-Life[Cor98] in 1999, Counter-Strike[CE12] was probably the game that popularized the S&D game mode. Players take the role as either terrorist (T) or counter-terrorist (CT) forces on a variety of S&D and hostage rescue maps. The game has been through a number of iterations and re-releases, but many levels and gameplay features remain fundamentally unchanged. In S&D the counter-terrorists are the defenders and the terrorists are the attackers having to place the bomb.

3.2.1 Gameplay

Each player is able to carry one hand gun, one rifle, a small number of grenades, and a knife in their arsenal. The player is only able to use one of these weapons at a time, but is able to freely switch between them on the fly. The firearms generally do a high amount of damage with only a few shots needed to kill a player and for most weapons just a single shot to the head is fatal. All firearms have different properties which makes them better suited for different playstyles and distances to the target. For example is a shotgun unsuited for hitting targets far away, while a sniper would be well suited for it.

Besides the S&D game mode, there is another game mode called *'Hostage Rescue'* where the counter-terrorist team has to locate and escort a number of hostages, back to their teams spawn point. The goal of the terrorists is to stop the counter-terrorist players from rescuing the hostages and if any player kills a hostage he is penalized. If the timer runs out, the terrorists win by default.

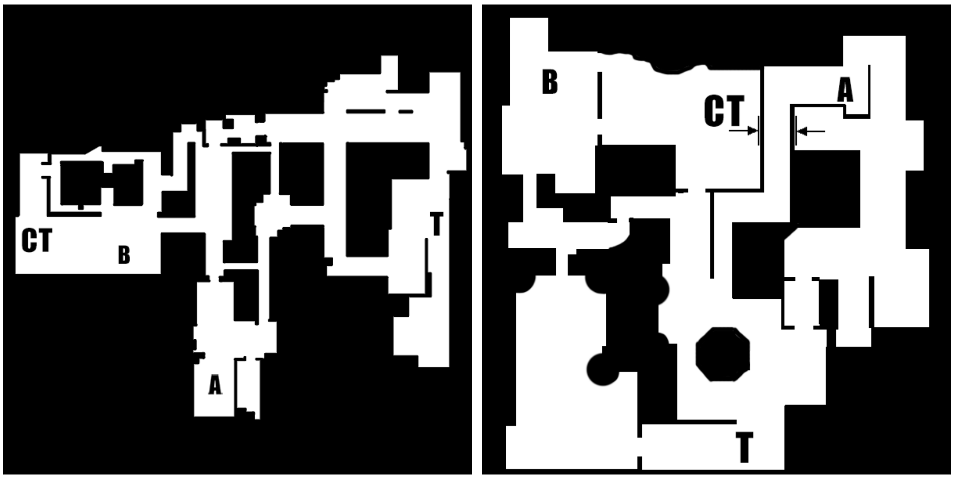


Figure 3.1: *'Dust 1'* (left) and *'Dust 2'* (right). Two examples of Search & Destroy maps. 'CT' is the spawn of the Counter-Terrorist team, while T is the spawn of the Terrorist team. A and B are the two bomb positions.

3.2.2 Levels

We have drawn the outline of three different Counter-Strike levels: *'Dust 1'*, *'Dust 2'* figure 3.1 and *'Office'* figure 3.2. We chose to analyse these levels based on their popularity and their simplicity. These levels have relatively little height variation which makes it easier to look at from above. The spawn points have been marked with Counter-Terrorist (CT), Terrorist (T) and A & B for the bomb points.

First and foremost, we can clearly see that CT and T are positioned in direct opposite ends of the levels, and there is no line-of-sight between the two spawn points. Had this not been the case, the tactical aspect of the game would be irrelevant, since the players would have no time to make any decisions before engaging the enemy. Therefore on all maps the spawn positions are not visible from one another.

Another observation is that the levels are rather *blocky* and most corners are perfectly perpendicular. They seem similar to the levels generated by dungeon generators like the one seen in figure 2.3 and the level generator for Hitman seen in figure 2.5. Very few features are curved and non-perpendicular. In *'Dust 1'* and *'Dust 2'*, figure 3.1, we can see that there are never any four-way intersections in the level. This means that players are never faced with more than two choices when moving forward in the level. This helps guide the players towards the enemy and avoid circling back to their starting position. The levels have no dead ends so every positions have at least two directions for the player to take. General features of the levels we looked at were a mixture of rooms or an outdoor area, connected via corridors of varying length and width.

To help guide the players through the levels, red graffiti on the walls are placed to point to the path that will take the player to either A or B bomb point. The bomb points themselves are marked with red graffiti on the ground.

Finally we can see that the bomb points are always located closer to the CT spawn point making the defending CT players have a defensive advantage. Having the bomb points closer to the CT spawn position, it means they can reach these areas first and have a tactical advantage when the attackers arrive.

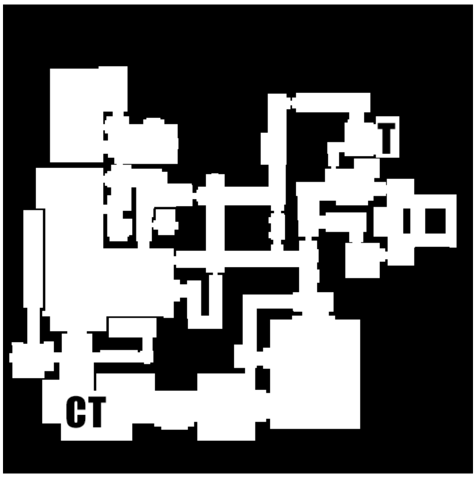


Figure 3.2: The Counter-Strike map Office for the 'Hostage Rescue' game mode.

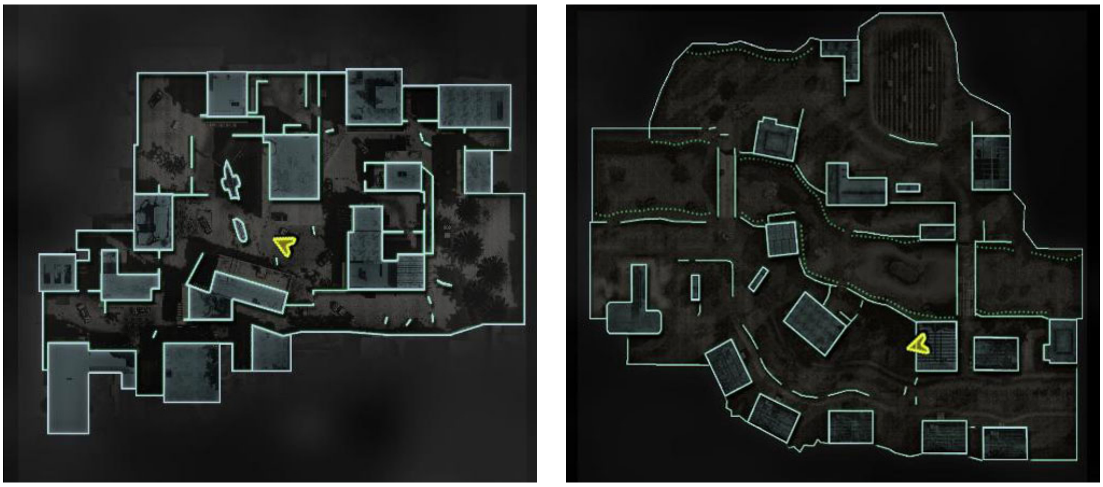
The *'Office'* level in figure 3.2 is not meant for Search & Destroy, but instead 'Hostage Rescue'. This level clearly differs on some of the points stated above. For example does this level contain four-way intersections. It contains a higher amount of corridors and corners than S&D maps, since in this game mode it should be likelier for the CT to move with few encounters to the T spawn point (where the hostages are located) and back again. It still has the perpendicular level design, and CT and T still spawn in opposite ends of the level.

3.3 Call of Duty

Call of Duty[War03] is an FPS game series first released in 2003 with a major iteration released every year since then. Although many parts of the gameplay largely remain the same, we are focusing on the series from Call of Duty 4: Modern Warfare [War07] and onwards.

3.3.1 Gameplay

The Call of Duty-series controls similarly to Counter-Strike on a lot of areas. The rules are similar, where players have to eliminate the opposing team in a number of different game modes, including Search & Destroy. Players can only equip a single weapon at a time, and a round of gameplay is limited with a timer. Where Call of Duty diverges lies in some of the finer details. Players can aim down the sight of their weapons for added precision. Players can sprint and lie down on the ground instead of just crouching like in Counter-Strike, and grenades and knives can be used without changing weapons first. Other very small differences exist to speed up the gameplay in subtle ways, resulting in a more fast-paced and chaotic game experience.



(a) The map 'Crash'[1]

(b) The map 'Overgrown'[2]

Figure 3.3: Two maps from the game Call of Duty 4: Modern Warfare.

[1] Source: http://www.cod4central.com/cod4-map-overgrown.php

[2] Source: http://www.cod4central.com/cod4-map-crash.php

3.3.2 Levels

Unlike Counter-Strike, the levels in Call of Duty are not designed for a single game mode, but instead work with all game modes (the game mode count varies between 11 and 15 depending on the game) and the levels need to work for multiple kinds of encounters. The overall gameplay of Call of Duty is a bit faster, and along with the general purpose levels which is noticeable in the level design, with less guided routes and more choices as the player moves forward. An interesting consequence of this is that since there are so many choices, it is hard for players to ever be fully aware of what is behind them, and as such always have to look over their shoulders to make sure that enemies are not approaching from behind. The levels are generally more open and less angular in their design. The levels we looked at were chosen because of their ranking online as the best levels among players. Both maps appeared in Call of Duty 4: Modern Warfare as well as the sequel: Call of Duty: Modern Warfare 2[War09]. Looking at figure 3.3a and 3.3b, we see that the levels are less constrictive as opposed to Counter-Strike's corridor design. These levels are more open, and are not as suited for directing the players towards any particular point of interest without using pointers as part of the user interface. Since there are no clearly defined paths, it is more difficult in S&D games to meet at constrictive choke points, rather than a big battlefield. However in order to guide the player during a S&D match, an overlay will show the position of both bomb points.

3.4 The Good Engagement

3.4.1 Choke Points

In Counter-Strike, choke points are areas where the two opposing teams will encounter each-other and is one of the most important aspects in the game. Choke points can be referred to as bottlenecks or control points. In Counter-Strike this is where most of the battles take place, and the rest of the level is either used as incentive to get to these choke points, or as 'timers' for the defending team to reach the choke point just before the attackers arrive. A choke points is always positioned before the attackers reach the bomb position and it is usually fast for the attackers to get to another choke point without confronting the defenders. The exact location of engagements changes slightly depending if one of the teams play aggressive or not. In terms of the playing experience, the choke points give places with high probability of encounters, which for most players is where the fun in the game lies. These points help limit the amount of choices for the player, since most levels only have two-four choke points.

3.4.2 Overview

When designing a multiplayer map for S&D, almost all maps follow some relatively simple but powerful guidelines. In order for us to create good maps, it is vital to understand what these guidelines are. Following these rules will result in what we call *'The Good Engagement'* (TGE); a level experience where the intended strategic and skill-based gameplay shines through. TGE does not mean that the game is fun to play, but rather the level design supports and encourages interesting choices, from which fun should emerge.

If a S&D map design follows the rules of The Good Engagement, the bomb positions, spawn points, paths and choke points will all be placed in a balanced way and make sure that when players engage each other, that they have the appropriate advantages and disadvantages, depending on which path they choose. A level should have different types of engagements to suit different play styles. In general a map should have both short, medium and long engagements which refers to the distance between opposing players when they spot each other. This is due to some players preferring sniping from long distances while others prefer quick reflexes in close encounters.

In S&D interesting choices often occur around the choke points. The players are at the start of each round given the choice of which choke point they want to engage the enemy at. On both sides of the choke points player often have the ability to move to another choke point without entering the opponents side of the battlefield. Playing a map enough times will give the player a familiarity with both the map and how the it is usually played. Given this knowledge the player can make 'interesting' choices about how, when and where to engage the enemy to create an advantageous situation. With S&D it includes when to plant and defuse the bomb in safety from the opposing team.

After having looked closer at interesting choices, Counter-Strike & Call of Duty's maps and multiplayer gameplay, we have narrowed down some key aspects when designing levels.
Maps should always have a few choke points which the teams would reach at about the same time. In Counter-Strike it would normally take around 10-13 seconds to reach the choke points. Furthermore bombs should always be closer to the defending team, since the defenders should have a small advantage. Bomb points, as well as spawn points, were never visible between each other, and neither should ours be, as it removes choke points and turns the areas into a single big fighting area.

Call of Duty has both levels that are very open as well as corridor-based while Counter-Strike is almost exclusively corridor based. Some open Call of Duty maps have less discernible choke points, where corridor based maps have clearly discernible choke points, which we should aim for.

Both games clearly indicate where the bomb points are located, to make it easier for players to find their way to their objective, which further helps players to move quickly forwards and funnel them to the choke points.

3.5 Discussion

Our approach to this chapter has been to read about the competitive aspects of the games online, play the games, look at level design courses for Counter-Strike maps and then look at the maps ourselves. Unfortunately there has been some scarcity in terms of research papers on the game- and level design of either of these games, and that is why we had to analyse the games ourselves. We chose to give brief descriptions of very basic aspects of FPS games because our test game needed to have these aspects implemented.

3.6 Conclusion

We have looked at both Counter-Strike and Call of Duty, narrow down some key aspects of the games and introduced the term 'The Good Engagement'. The games varied a lot on level design, with Counter-Strike focusing on a single game mode per level, while Call of Duty creates their levels for all game modes. Introducing 'The Good Engagement' allows us to use it as a rigid rule-set for our approach to generate good levels

4. Preliminary Approach - Prototypes

4.1 Map Generation Prototypes

As multiplayer FPS map generation is a largely unexplored topic, several prototypes were developed to test what approaches could generate high-quality levels.

4.1.1 Connected Rooms

Connected rooms was a paper prototype where several rooms where placed and connected manually where two approaches were tried. The first approach placed rooms randomly and assigned functions to them, such as bomb- or spawn-position. The second approach placed the important rooms manually instead of randomly and then connected them.

Both attempts did not result in any pleasing maps. Represented as nodes and vertices, the layout seemed to fit TGE, however when drawing the rooms and corridor layout, there were many overlapping corridors which created a high number of four-way connections. This did not fit with our definition of TGE, as the maps did not contain any interesting choices. The corridors between the rooms inevitably resulted in simple movement paths where it was not clear whether interesting engagements could take place. It turned out to be too complicated to make sure that all rooms were well-connected and corridors would not overlap or be too long.

4.1.2 Direct Player Influence

One of the first prototypes was a simple All-White map generation where the players could directly modify the map. After the players killed each-other a few times, the game mode would switch to a building mode where the players could change a single or a few level building blocks, which consisted purely of cubes. This direct influence inevitably resulted in 'Minecraft'-inspired[Moj11] gameplay with no coordination and not enforcing TGE, but instead encouraged random play. As such we opted to not use direct player involvement or any sort of mixed-initiative.

4.1.3 Voronoi

A voronoi diagram is a method of dividing an area into regions. Usually a number of points are placed on an empty area, and its corresponding region is the area which is closer to it than any other point. The Voronoi technique has successfully been used to generate realistic landscapes and islands[Pat10]. For this prototype an All-Black map was used, and the edges along bordering regions of the voronoi diagram were used as walkable corridors. The resulting maps did not fit with the TGE despite having no four-way intersections. Using both random and more rigid methods of placing the points, the maps were consistently too open, had too many long corridors, a lack of interesting choices and had a number of dead ends. While some of these problems could be minimized, it did not seem promising and was discarded.

4.1.4 Base Skeleton

This was a very simple prototype where a pre-generated skeleton for the map was used, consisting three paths from one spawn point to the other. The algorithm could add connection points on the paths, connect two paths and slightly change the position of a connection point. It did not work at all as it just created more connections with only minor impact and did not provide much variety to the levels generated due to the rigid skeleton.

4.1.5 Prefabs

Using a few game assets like houses, walls to duck behind and pillars, this approach combined them through anchor points when being placed to create more elaborate constructions. This created a sense of familiarity as you could clearly see that the prefabs themselves were designed by a human. As the buildings were made to be stackable, tall traversable towers would emerge from the random generation. The items were placed randomly on an All-White map and rotated in 90 degree intervals. However without a more advanced placement algorithm it could only create simple deathmatch inspired levels that could not guide the player around, or provide any structured layout. Creating levels with good engagement required more than random placement. While prefabs added familiarity and visually pleasing obstacles, it did not help solve the actual problem of generating the level in a meaningful way.

4.1.6 Wandering Agents

Wandering Agents was an attempt to see if interesting level layouts could occur naturally by wandering agents carving out an All-Black map. Agents were placed in two of the four corners of a map, and would randomly wander the map. They would often change their direction slightly, and the further they walked, the more they would seek out the opposite team's spawn point. The resulting maps were often very open as the agents could end up walking in circles. There were also a high number of dead-ends on the maps. Following the same idea we tried to place clusters of obstacles on an empty map, but it resulted in very open space and no meaningful choices for the players.

4.1.7 Connect the Points

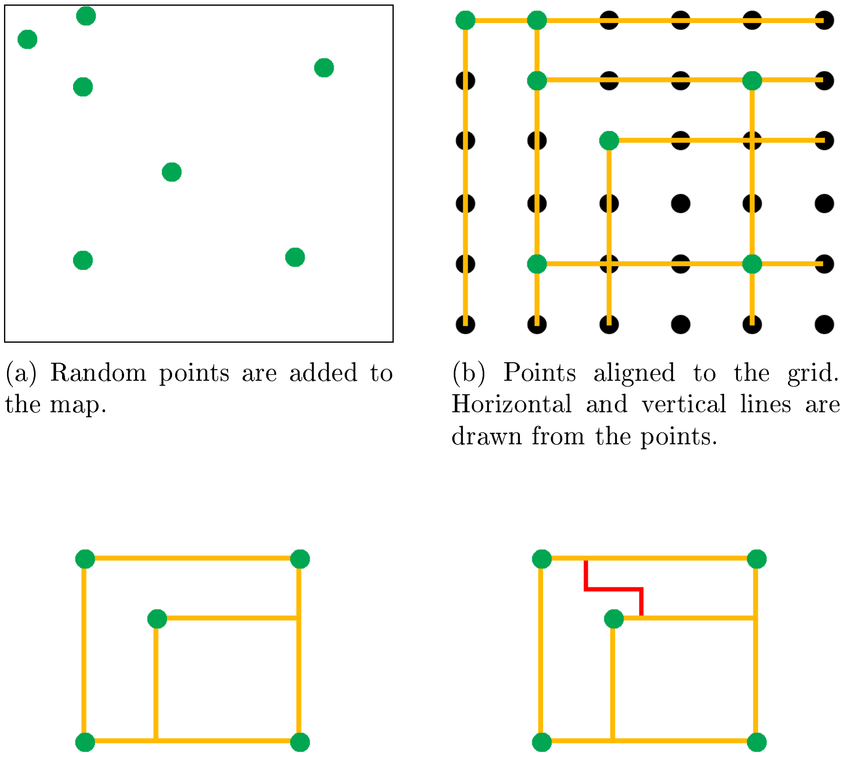
With the previous unsuccessful results in generating any meaningful levels by pure random movements or placement of things, a more predictable level building algorithm was needed. Since three of the attempts generated too open levels, an All-Black approach was taken. This method focuses on the connections themselves where the rooms, spawn- and bomb points are placed on top. This approach was inspired by the 'Connected Rooms' and built from a more predictable method which could be encoded into a genotype and mutated.

Algorithm

This approach comes from the observation that most Counter-Strike maps have a very perpendicular design, and can be seen as a grid if corridors are simplified. In figure 4.1 the four steps in the process are visualized.

When the basic structure had been created in the third step, the final structure addition was adding shortcut paths. The idea behind these paths was that a simple skeleton structure could be generated similar to the approach in section 4.1.4 where shortcut paths would make up for any limitations in the skeleton. In this approach the base skeleton was made less rigid and mutable. Spawn positions were added to be on opposite ends of the level and assuring that they were not directly visible from other. The two bombs points would always be placed in positions that could be reached faster from the defending team's spawn position than the attacking. While simple in concept, these placement rules did not change much for the final approach as described section 5.4.

This approach did have certain limitations that lead to the development of a revised approach. One of the problems was that the mutation of a single points position could result in drastically different levels. In the example given in figure 4.1, if the lower right point had moved further south or east, it would not have connected to the other points, resulting in the removal of the entire lower right corner of the map, which in turn could affect the culling of other parts of the level structure. This problem only grew as the grid resolution increased. Furthermore this approach had a tendency of generating a lot of long parallel corridors that did not result in good engagement. In addition the shortcut paths did not prove to be successful, as we could not find a good balance between the complexity of the skeleton and the frequency of more flexible shortcut paths.



(a) Random points are added to the map.

(b) Points aligned to the grid. Horizontal and vertical lines are drawn from the points.

(c) Dead-ends are removed.

(d) Paths are added

Figure 4.1: The four steps of the 'Connect the Points' method, see 4.1.7.

5. Approach

5.1 Interactive Evolution

The decision between interactive evolution and mixed-initiative was one of the first to decide between. Both approaches allowed for player-driven evolution of the levels, but mixed-initiative generally requires more interaction with the player, where interactive evolution requires less. Since the final solution had to be playable by multiple people, the fewer interactions with the system were required, the faster the iteration speed would be. Mixed-initiative often encompasses some sort of tool that the player works with, and having seen the chaotic nature of multiplayer creation, we opted for interactive evolution as the superior approach. As experienced in the 'Direct Player Influence' prototype in section 4.1.2, direct control easily resulted in very chaotic and unorganized level creation the more players were in direct control. Interactive evolution also solves the problems with agent-based fitness[Car+11], that we did not consider good measurements fun.

5.2 Game and Gameplay

If the gameplay was not at least representative to that of the inspirational games, the results of the playtest might be skewed by players' opinion of the gameplay, and could affect their overall impression of the level design. It should be noted that the gameplay and the level design cannot be completely evaluated on their own terms as they mutually complement each other. Everything from movement speed and map size to game modes and map layout will inadvertently affect the other.

Using Counter-Strike as our primary example, it would have been an obvious approach to produce the levels directly in the game's engine, as it could save time on gameplay development. The reason we did not take this approach was due to the game engine not being as flexible as we needed it to be, to generate levels during runtime execution.

5.2.1 Game design

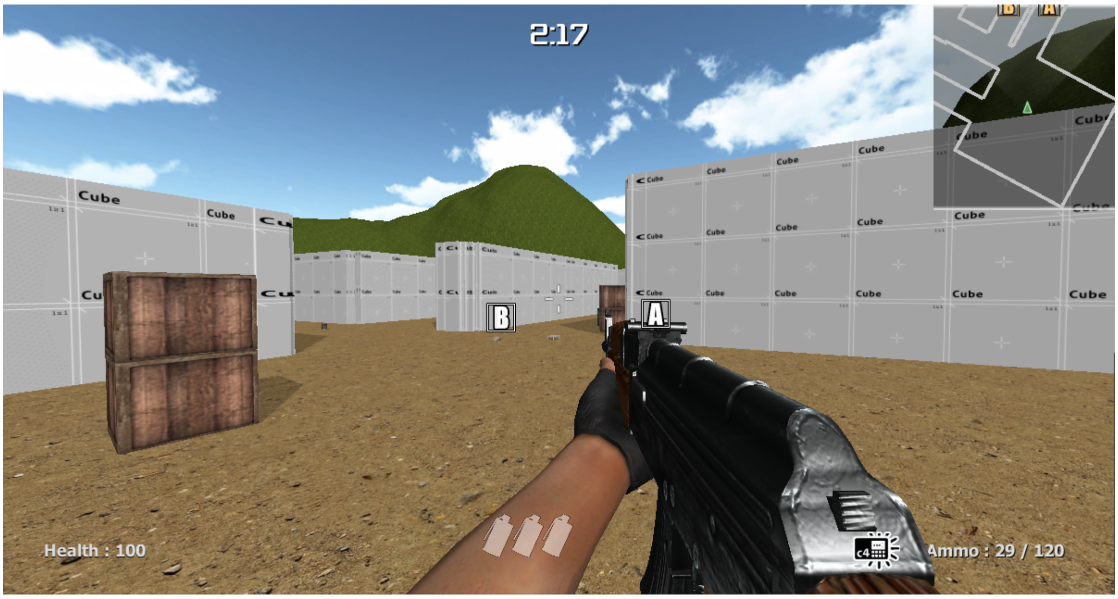


Figure 5.1: The game as the player sees it.

Building a custom game instead of leveraging an existing, allowed us to more easily adapt aspects of the design to suit the game better. In the interest of time we chose to work with a pre-made system for Unity with modern shooting- and movement mechanics similar to that of a modern FPS game that we could easily modify. The 'Realistic FPS Prefab' [1] provided most of the features that was needed for the game and resembled Counter-Strike and Call of Duty in both visual elements and gameplay. Only few small changes were required to make it more similar to the mentioned games, such as matching the movement speed to the general map size of the levels, adding missing features like explosive grenades and integrating the networking. The pre-made package provided the necessary mechanics that players would be expecting. Crouching, shooting, the floating hands in front to the camera and systems to handle hit points and ammunition are features that exist in both Counter-Strike and Call of Duty. Sprinting and precise aiming by looking down the sight of the weapon were features that only appeared in Call of Duty, but we chose to include these, as they are both popular mechanics in modern FPS games. We chose to implement three common weapons: an automatic rifle which would be suited for close and medium combat, a sniper rifle for long ranged combat and finally a hand gun that is used as a backup weapon in case the player runs out of ammunition.

An important layer on top of the game mechanics is the user interface (UI) elements that informs the player on the status of the game, with pop-up messages when players

[1] 'Realistic FPS Prefab' by Azuline Studios `https://www.assetstore.unity3d.com/en/#!/content/7739`

plant or defuse the bomb, and when the round begins and ends. Icons were also added next to their ammunition counter, to show whether or not players have the necessary equipment to plant or defuse a bomb. These icons will also flash when the bomb is in range to be interacted with. Furthermore players had to be able to orient themselves and navigate with as little difficulty as possible. To keep players aware of their surroundings and positioning, a mini map was added to the corner that could be expanded to a full map at the press of a button. This map would highlight all enemies that were within line-of-sight of the player, and display small prompts when explosions or gunshots could be herd, as well as their origin. Replicating Call of Duty, we also added in-game icons that precisely showed the position of the two bomb points, as can be seen in figure 5.1, that also shows many of the aforementioned UI elements.

5.2.2 Game mode

Game mode and map design depend on each other to deliver an optimal experience. If the map design does not facilitate the objectives of the game mode used, it can create a suboptimal experience. Table 5.1. In terms of gameplay, we needed a game mode to build our levels for, that also would require players to consider their surroundings, and where the fun in the game was dependent on the quality of the level. Looking at different online FPS games, we listed the most popular ones from a multitude of games.

Through a process of elimination Search & Destroy was the chosen game mode. The most important consideration was the optimal strategy of play in the game mode. The game mode should incite players to confront the opponent team, thus eliminating modes like 'Hostage Rescue', 'Last Man Standing' and 'Capture the Flag'. The second consideration was that players should be penalized by their death, eliminating the modes both variants of 'Deathmatch' and 'Domination' as respawning is fast. 'Search & Destroy' was the only game mode that both incentivized confrontations and survival at the same time. It forces the bomb placement team to try and place the bomb, and if successful it forces the defending team to try and defuse it before it explodes. If the bomb placement is not used much, it still forces survival to win the round. It is an advantage for either team to quickly familiarize themselves with the level to gain an upper hand in confrontations, by utilizing alternative routes for flanking or simply using the shortest path.

5.2.3 Networked Multiplayer

In terms of distributed gameplay, an issue was whether or not to implement a solution for local or online multiplayer. The problem with online multiplayer is the inherent delay and the infrastructure required to support many concurrent players. While Unity has basic networking built in, it does not contain any features to help with online gameplay synchronization. Developing a game and an infrastructure that could let people play against each other online would be a major undertaking that would remove focus from the evolution and gameplay of the project. An online game would generate a lot more data that could be analyzed as access to the game did not require local test sessions,

Game mode	*Explanation*	*Considerations*
Deathmatch	Kill everyone else. Highest number of kills wins.	Chaotic and too much random movement.
Team Deathmatch	Kill the opponent team. Highest number of kills wins.	More structured, but sill little emphasis on tactics/planning.
Capture the Flag	Steal the enemy flag and bring it back to your teams base.	Does have strategy involved, but relies on symmetric maps for each teams half. Fast pace where avoiding the enemy team is the optimal strategy.
Search & Destroy	Plant the bomb as attackers. Prevent the planting of the bomb or defuse it as defenders.	Requires interesting choices and facilitates good engagements. Slight advantage required for defending team. Dead players have to wait. Forces confrontations to win.
Last Man Standing	Kill everyone else. Last surviving player wins.	All levels likely applicable. Potentially very long games, as hiding is the optimal strategy for survival. Dead players have to wait.
Domination	Team who controls each of three small areas for the longest combined time wins.	Less dependent on good levels due to generally open maps with few choices. Encounters are not random, but chaotic around domination points.
Hostage Rescue	Rescue the hostages or eliminate the other team.	Requires a lot of planning, and complex levels to remain obscured. Optimal play for the hostage rescue team is avoiding the enemy and thus good engagements are less frequent.

Table 5.1: Common FPS game modes and considerations for each.

however we deemed the additional work out of scope for the project. We opted for local networking as it is an easier problem to solve. As the ping is nearly zero is means synchronization is less of a problem and very little infrastructure is required. Local networking at the complexity we needed was handled using the standard solution in Unity.

5.3 Selection - Level voting and Rating

A key part in evolution is selection. A fitness function usually decides the best suited parents for new offspring, but we intended to go for a route where the fitness was also based on user input. Usually in a simulation is run with the latest offspring [lM12], and all simulations are rated using fitness functions. An example of this can be seen in Cardamone et al [Car+11], but we did not consider the fitness function used in that instance a good measurement of fun, and relies heavily on the AI to play similar to real players. Our implementation uses a system related to that seen in the Sentient Sketchbook [LYT13] and Picbreeder [Sec+08] (see section 2.1) that would let a designer

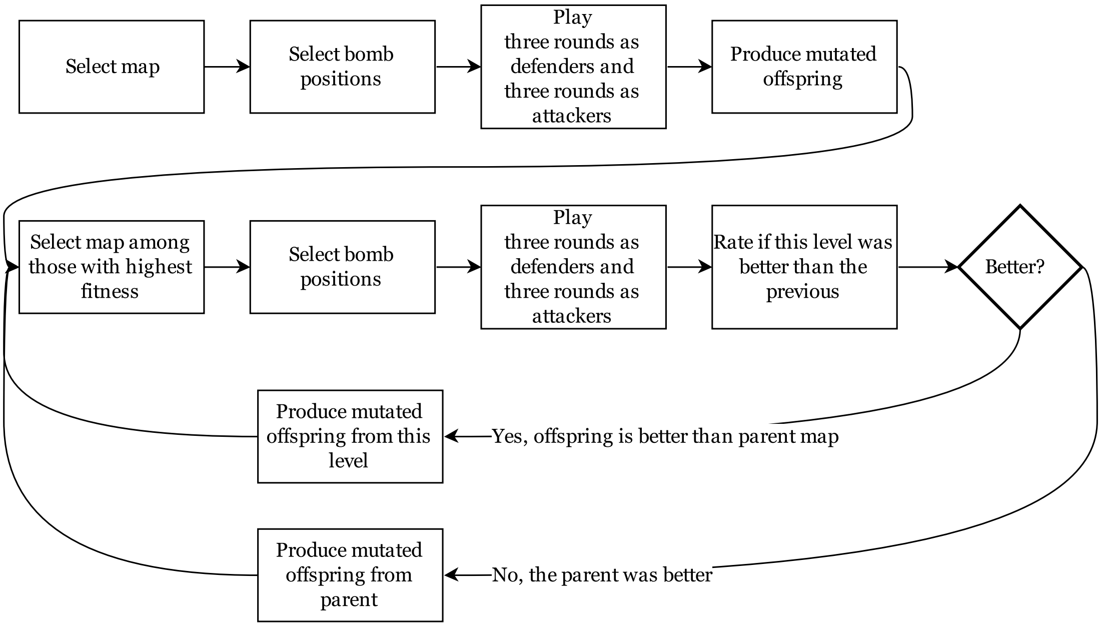

Select map
Select bomb positions
Play three rounds as defenders and three rounds as attackers
Produce mutated offspring
Select map among those with highest fitness
Select bomb positions
Play three rounds as defenders and three rounds as attackers
Rate if this level was better than the previous
Better?
Produce mutated offspring from this level
Yes, offspring is better than parent map
Produce mutated offspring from parent
No, the parent was better

Figure 5.2: The flow of the game. Note how the rating screen first does not appear after the first session and how a negative rating can make the flow backtrack to a previously selected map.

vote on the level he thinks is best. Similar systems are often seen in multiplayer FPS games where two levels are displayed and can be voted on, and the level with the most votes will be played next. Our voting screen (seen in figure 5.3a) replicated this, but would display six levels instead of two. Using fitness algorithms that were based off of our rules of The Good Engagement seen in section 3.4.2, we could filter out the levels from a large group of offspring, that did not follow our rules of TGE and present six levels that passed the fitness algorithms. This fitness algorithm mostly acted as a filter rather than a traditional fitness function, since it provided little or no granularity, which was the role of the players. The filter would discriminate against levels that suffered from the problems described in table 5.2 with too long corridors without any choices, spawn points that were immediately visible from one another, imbalanced maps where one team had a clear advantage (more than 85% of the level was closer to one team) and the spawn points were not reachable from one another, which would indicate that the level was broken into two. Even though it is standard for multiplayer FPS games to have only two levels to vote on, we chose this higher number based on the fact that selecting two levels amongst a huge crowd of potential offspring would limit the view for the user, which would not allow for freedom in guiding the evolution. Since we intended to use a voting system with a single vote at first, this number would also make sure that votes could be distributed without leaving too may levels that had received no votes. We also included a button that would refresh the set of levels which could be voted on if

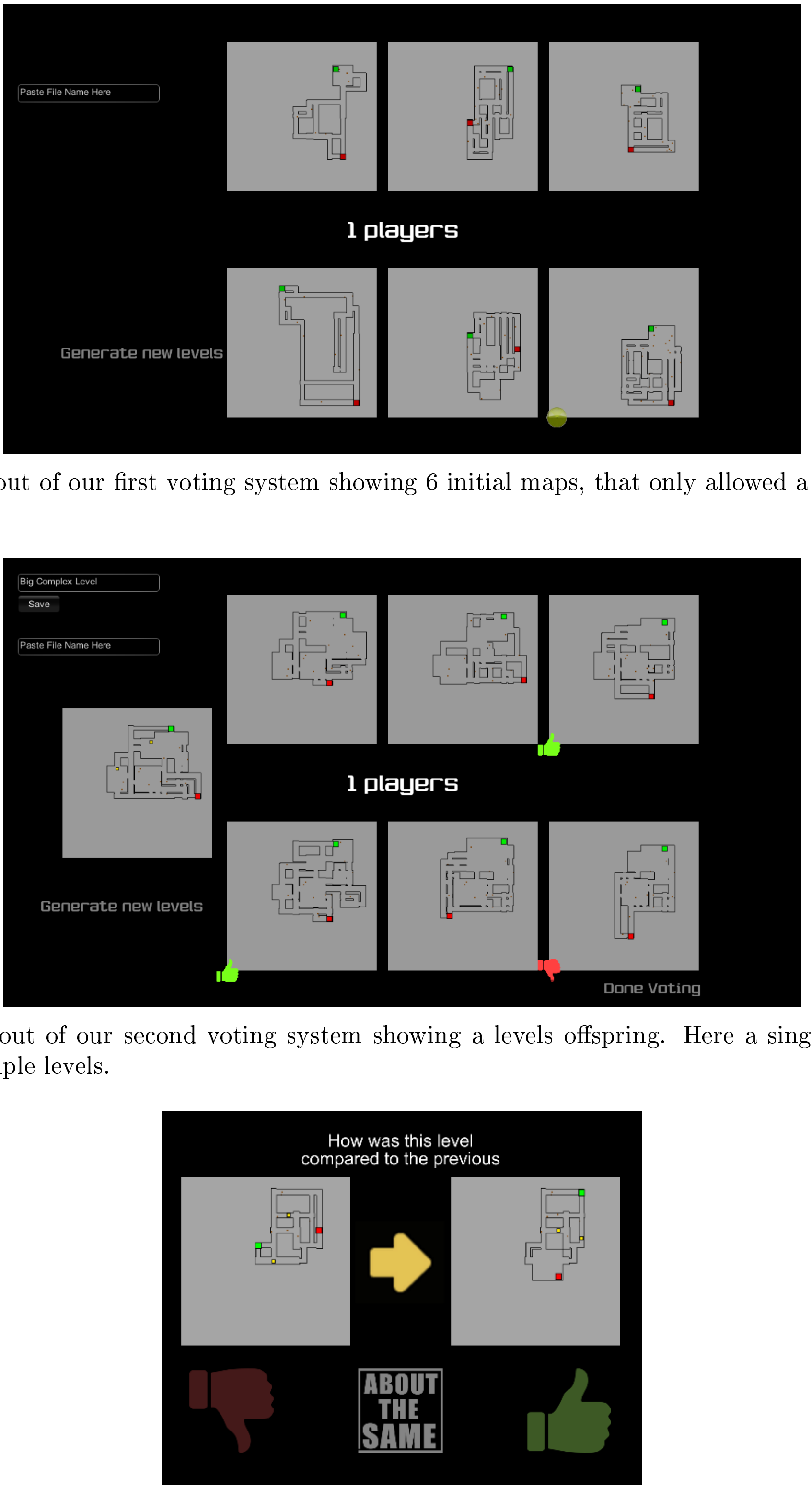
Problem	*Description*
Bomb point proximity	Bomb positions are too close to one another
Long corridors	There are too long corridors without exits
Too high complexity	The map is too connected, making any single choice meaningless
Too little complexity	The map is too simple, thus giving few choices
Imbalanced map	Too big part of the map is faster for one team to reach than the other
Directly visible spawns	Opposing teams' spawn points are mutually visible
Traversability	It is impossible to reach one spawn from the other

Table 5.2: The most common issues for generated maps before being filtered.

people were not satisfied with the selection of levels. Since this button followed the same voting rules as the levels, it also required to be the best voted, to refresh the selection of maps. When new maps were generated, a population of 75 was created before the six contenders were presented to the players. The population size was determined though a balance of variety and the time required to generate the levels.

To make sure that levels always improved in a lineage, and would not be hampered by a bad level that user might have expected to be better, we included a rating screen after each level, asking whether or not the current level was better than the parent. This can be seen in figure 5.3c. If this was the case the lineage would continue with the current level creating more offspring. If the level received more negative votes than positive however, the lineage would rewind to the parent, and would only progress once a better offspring had been found. Since this rating system required that users had played both levels, it would only appear after the second round during testing. This can also be seen in the flowchart in figure 5.2 that describes this process.

Between two of our playtests an alternative voting system was implemented, inspired by Cardamone et al [CLL11b], where players would rate the levels instead of voting for them using either a thumbs up or down to indicate whether the level was good or bad. This can be seen in figure 5.3b. If a user did not feel either way they would not rate that level at all, which would be equal to a neutral vote. A score would be calculated for each level, where a positive vote would increase the score by one, and a negative reduce it by 1. At the end the level with the highest score would become the successor. If two or more levels shared the highest score, a random draw would be made among these. The voting process was also expanded with a second voting screen, where all the levels were similar, except the bomb points. This was due to the fact that during a playtest, players responded negatively to the position of bomb points, and selecting the layout separately from the bomb position helped create good level layout with good bomb position.



(a) The layout of our first voting system showing 6 initial maps, that only allowed a single vote per player.

(b) The layout of our second voting system showing a levels offspring. Here a single player is rating multiple levels.

(c) The rating screen players are shown between levels for determining whether or not the new level is better than the previous.

Figure 5.3: The interactive tools players are presented with when determining selection.

5.4 Algorithm

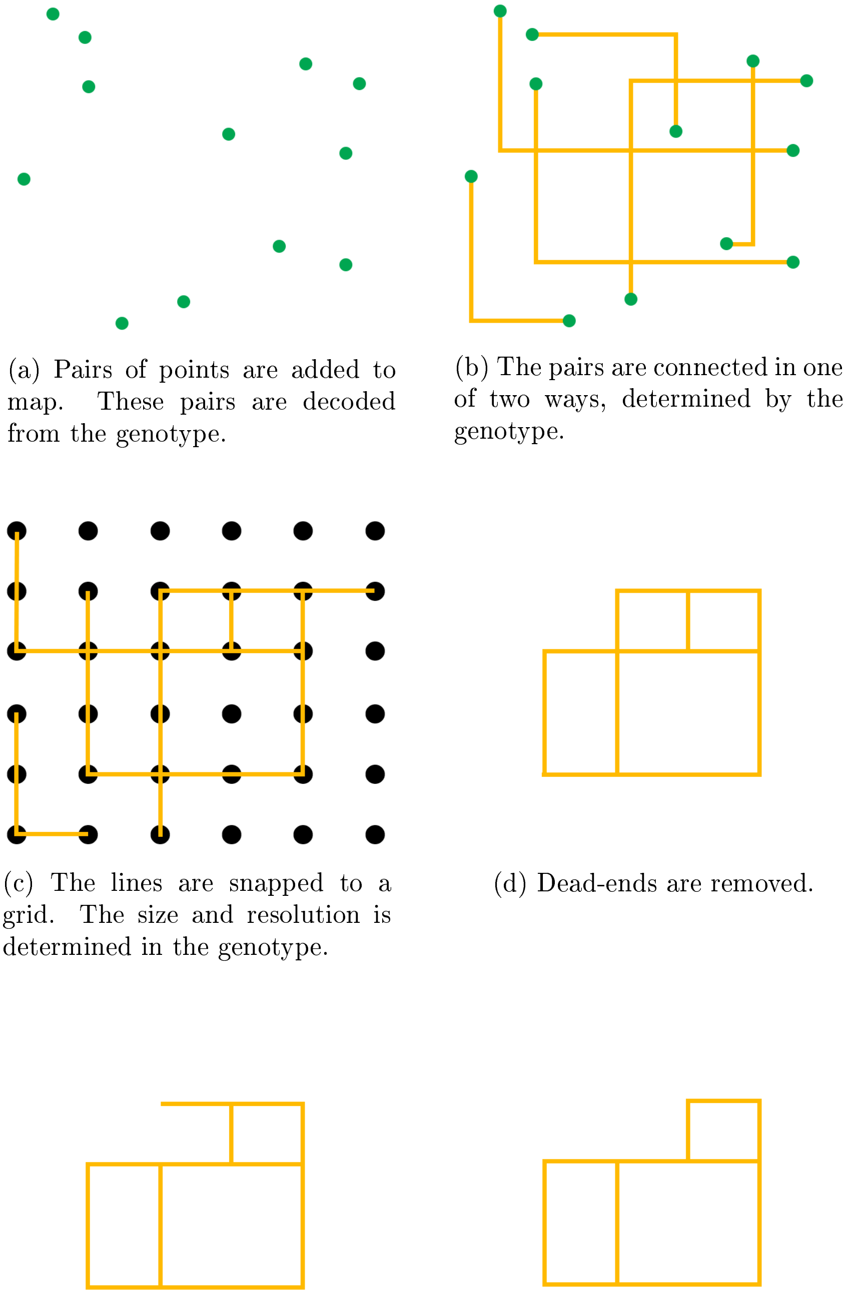
In section 4.1.7 we looked an approach called Connect-the-Points for making good level layouts using random data (or in our case, encoded as mutable data). This algorithm was the closest we had gotten to creating levels with interesting choices. A number of issues with Connect-the-Points made it unsuited for evolution, which we attempted to fix with a redesigned version of the algorithm. The biggest problem with Connect-the-Points was its fragility when being mutated along with its tendency to make long parallel corridors. These problems were counted by redesigning the algorithm.

By placing pairs of point, we could end up with interesting levels that resembled the layout and perpendicularity of Counter-Strike levels. Pairs of points would be placed and snapped to a grid, and the pairs would be connected using a horizontal and vertical line to create two corridors. It is important to point out that the points could be connected in one of two ways, either horizontal first and vertical second or vice versa. Which way the connection would be made, was determined by the genotype, which is described in section 5.5.1. Once all the pairs had been connected into the L-shapes, the lines would begin to resemble the structure of a level. Due to TGE presented in section 3.4.2, a number of modifications to these lines had to be made, to make sure that the rules previously established were being upheld. The first step would be to eliminate all dead ends and reduce 4-way connections to 3-way connections. Again, the direction would be determined by the genotype. This part of the process created the outline for the map and an example of this process can be seen in figure 5.4. In the second part, seen in figure 5.5, these lines would be carved out in an All-Black space, and more details added based on values in the genotype. Areas would be carved out as rooms, and props and doorways would be placed in the corridors.

Finally the spawn points and bomb points were placed, which is described in section 5.5.1

It is important to note that the approach is deterministic since all the necessary values are encoded in genotype, and no random values are used when building the level. This encoded genotype also meant that the levels could be more easily transferred over the network, since the data size of the genotype is substantially less than that of the physical level encoded into values.

5.4.1 Level Outline



(a) Pairs of points are added to map. These pairs are decoded from the genotype.

(b) The pairs are connected in one of two ways, determined by the genotype.

(c) The lines are snapped to a grid. The size and resolution is determined in the genotype.

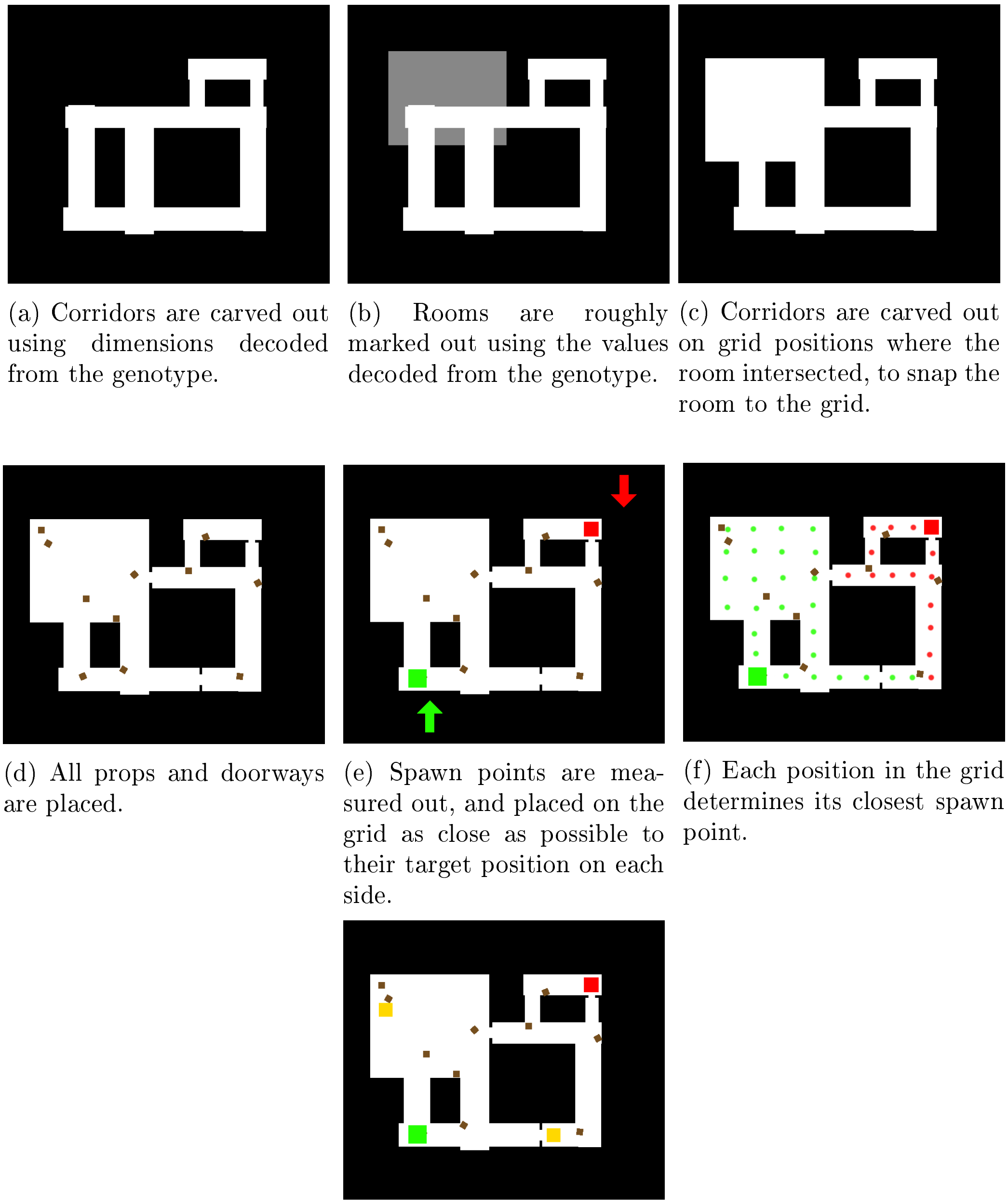
(d) Dead-ends are removed.

(e) One connection is removed in every four-way intersection to eliminate openness.

(f) Finally dead-ends are trimmed again.

Figure 5.4: Step-by-step walkthrough of the algorithm where level outline is created, as described in section 5.4.

5.4.2 Building the Level



(a) Corridors are carved out using dimensions decoded from the genotype.

(b) Rooms are roughly marked out using the values decoded from the genotype.

(c) Corridors are carved out on grid positions where the room intersected, to snap the room to the grid.

(d) All props and doorways are placed.

(e) Spawn points are measured out, and placed on the grid as close as possible to their target position on each side.

(f) Each position in the grid determines its closest spawn point.

(g) Bomb points are placed in the defenders area.

Figure 5.5: The process of building the map from the outline in figure 5.4, as described in section 5.4.

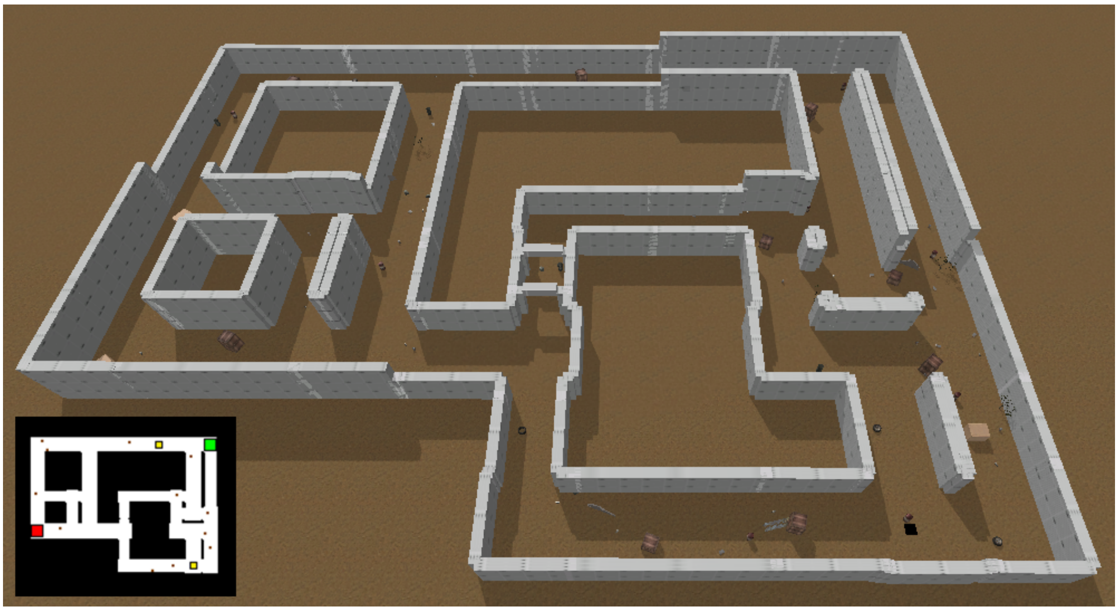


Figure 5.6: Bird's eye view of a generated level. The level is part of a lineage of levels from one of the playtests, that can be seen in appendix A.2. The map representation has had black areas cropped out to best fit it in the picture.

5.5 Evolution

5.5.1 Genotype

The first data we had encoded in the genotype was the size of the map. This is represented as the size and resolution of the grid, as well as a unique ID and the parent ID. The ID data had to be encoded to help us follow the evolution of the levels. All values detailing positions would after this point have to be between zero and one (normalized), so the resolution and size could still be altered, but still have all points scale to fit the new dimensions.

To encompass an L-shape, 2 positions and a boolean for the L-shaped connections would be required, but more data had to be encoded to create levels using our algorithm. For example the dimensions of the corridors had to be defined as well. These were added to the connections in the normalized values as well.

Secondly we needed a list of directions for the L-shapes. An important rule established in our Counter-Strike level analysis is that 4-way connections should be avoided. After placing the L-shapes we would have something resembling a grid, so we needed to trim one direction every time there was a 4-way intersection, and this direction had to be genetically encoded. This list of directions was separate to the list of connections.

Next we encoded the details of the levels: rooms, props and doorways. Props are

Property	Example values	Mutation probability
ID	5421168	
parent ID	9458422	
width	155	5%
depth	172	5%
resolution x	20	5%
resolution y	20	5%
spawn direction	up	10%
spawn A	0.74	14%
spawn D	0.48	14%
bomb A	972	(85)
bomb B	3254	(85)

(a) The genotype header containing the overall level info. Each of these properties only occur once per genotype. The values in parenthesis are exponentially decreasing probabilities described in section 5.5.2.

Property	Example values	Mutation probability
position x1	0.14	1%
position y1	0.54	1%
position x2	0.75	1%
position y2	0.21	1%
combine horizontal first	true	1%

(b) The values contained in the genotype required for an L-shape. These properties are repeated throughout the genotype string.

Property	Example values	Mutation probability
cut direction	down, up, ...	5%

(c) Directions that need to be cut, when encountering a 4-way connection.

Property	Example values	Mutation probability
position x	0.42	5%
position y	0.32	5%
opening width	0.75	5%
direction	down	5%

(d) The values contained to create a doorway. If a direction does not match with the corridor it has been placed in, the doorway will not appear. The width is also limited to the size of the player, and openings too small will be expanded to allow players though.

Property	Example values	Mutation probability
position x	0.35	5%
position y	0.85	5%
width	0.88	5%
depth	0.39	5%

(e) The values used when creating a room.

Property	Example values
position x	0.42
position y	0.49
rotation	0.07
type	3

(f) The values contained to create a prop. These values are never mutated.

Table 5.3: The subsets of values contained within a genotype. Note that position values between 0-1 are normalized and the real value depends on the 'width' and 'height' in sub-table a.

represented by four values for position, rotation and a type. Rooms contain a position, a width and height. Doorways are a bit more complex as they contain a 2D position, the relative width of the opening and the direction that the doorway is facing.

Finally we encoded the spawn and bomb positions. Spawn positions are contained in two values and a direction. The direction dictates what side of the map the attacking team will spawn at, and the defending team will simply spawn on the opposite side. The values give a one-dimensional position for the two spawn points for each side, and they are finally placed on the closest active grid positions. Bomb points are simply integer values. After the map has been analysed, these values will count through the possible bomb positions, and then be placed where there is no spawn point or other bomb. This method for placing the bomb points could be further iterated upon to provide better and more predictable results.

5.5.2 Mutation

Mutation for this genotype is as basic as it gets. We have defined individual probabilities for each part of the genotype and how much it can change. As each part is defined via primitive data types using lists of booleans, floats and integers, it was straightforward to mutate as well as removing or inserting segments to/from the genotype.

Mutation probabilities can be seen in table 5.3. Values shown as percentages had that amount of probability of mutating. To keep the mutation of normalized position values from becoming too erratic, it was decided to keep variation within 0.2 of the original value, which corresponds to on fifth of the levels dimensions. Values contained in a parenthesis, were decreasing probabilities, used for mutating some integer values. First it was determined if the value was negative or positive. The value was determined by a series of probability rolls, where the value had *(x)* amount of chance of increasing until a roll failed. This provided an exponentially decreasing probability that could theoretically create numbers that approached infinity, but in practice resulted in values that were nicely balanced in size and frequency. Rooms and doorways also had a 24% probability of being added or removed. The mutation of props was mistakenly omitted from both playtests.

Considering that voting systems are in essence a human controlled selection, it could have been fitting to expand the mutation to include crossover. We did not implement this because the results could betray the players' expectations. When voting for levels, a preview picture is shown, but if the offspring did not represent the features that the players voted on, it would go against what people were voting for. It was safer to have a single parent, and only choose one child. A different method for voting with crossover implemented would be an interesting exploration in any future work.

5.5.3 Complexity

When referring to complexity in a formal sense, it refers to the number of three-way intersections on the map, as these add choices to player movement as discussed in section 3.4.2. Two-way connections are either corners or corridors, neither of which adds any choice to the map, while four-way connections are considered open spaces where its complexity comes from the three-way entrances. An example can be seen in figure 5.7. While it can be argued that not all the highlighted points add any real complexity, we chose this approach as it is a simple metric and avoids discussions if a point adds *'real'* complexity or is a part of a bigger intersection.

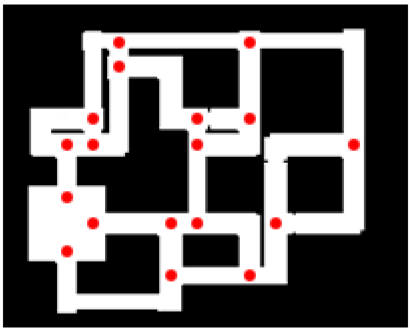


Figure 5.7: A map from a playtest with intersections highlighted in red. Intersections is the measurement used for complexity. Spawn positions and bomb positions have been removed, and the surrounding grid has been cropped. This map shows a complexity of 18.

6. Results

In this chapter, we present the results that are related to our research questions. As an introduction, the approach and details of the playtests are explained to make sure that readers are familiarized with how these data were gathered.

The main questions we aim to answer are whether or not the interactive evolution was a viable option for a group of players, whether or not the players evolved levels that matched their preferences and if they were satisfied with these results.

6.1 Playtests

Throughout the project, three playtests were conducted with players who had at least some prior experience with FPS games. For the second and third playtest, the testers filled out a questionnaire after having played the game. The quantitative results can be seen in figure 6.10 and appendix section C.2, where qualitative answers can be found in appendix D and E. The quantitative answers were either yes-no questions or a rating between 1 and 6 where 1 is the lowest and 6 is the highest. The maps played were recorded for playtest two while every map, vote, rating, kill and death was recorded for playtest three. The flow of the playtest can be seen in section 5.3. The maps displayed have black space around them, which is not cropped since it shows the full map size. Cropping the maps would not allow us to properly display the changes in the sizes of the maps or how big the levels are in relation to one another.

The goal of the first playtest (pt1) was to make sure that the game was ready to be played by multiple people without crashing or losing connection, and finding all annoyances and gameplay issues. This test was not meant to record data, only putting everything through its paces. A significant observation was that players had problems navigating the levels using the on-screen bomb point markers and the mini-map in the corner. The players also felt the level design seemed like mazes rather than real levels as the maps purely consisted of walls with no open spaces or other objects. For the second playtest we added open rooms to the levels, small props for cover, doorways, visual elements like bushes and bricks and a full map overview that could be displayed at any time.

The second playtest had three separate groups of four to five people (pt2.1, pt2.2 and pt2.3) and lasted 90 minutes each. For the first hour they played the game and evolved the levels. Every sixth or seventh generation we would reset the game to start a new lineage of maps. Between levels people would rate the relative quality of the maps. If the average result was a negative evaluation of a level, the lineage would backtrack and evolve from the previous level. An overview of the flow can be seen in section 5.3. After an hour of playing the game, the participants would then fill out our questionnaire (questions and answers can be found in appendix D) and assemble for a recorded group discussion where final thoughts could be expressed verbally[1]. No such discussions took place in the third playtest.

For the third playtest we increased the mutation probabilities slightly, and altered the voting system from the user only being presented with a single vote screen, to two voting screen, as described in section 5.3. The players would vote on the map first and then the bomb positions. The third test focused on gathering relevant gameplay and player data, to analyze if there were any correlations to the questionnaire data. We recorded votes and ratings to follow the progress of maps and how they evolved. A major change in this last test was the addition of a long-range weapon, a sniper rifle, similar to the ones found in Counter-Strike and Call of Duty. The playtest had 13 participants all playing together (pt3.1) and then later split up into two groups; the skilled players (pt3.2) and the less skilled players (pt3.3) based on the number of points the players scored in the first part of this playtest. An overview of the playtests can be seen in table 6.1.

This final playtest suffered from unidentified network issues, that hindered the progress in some sessions. Because of this, the sessions do not contain as many generations as those of pt2.

Playtest	Shorthand	Skill level	Player count
Second test	pt2.1	Skilled players	4 players
Second test	pt2.2	Medium skilled players	5 players
Second test	pt2.3	Low-medium skilled players	5 players
Third test, preliminary	pt3.1 preliminary	All skill levels	11 players
Third test	pt3.1	All skill levels	13 players
Third test	pt3.2	Skilled players	6 players
Third test	pt3.3	Unskilled players	7 players

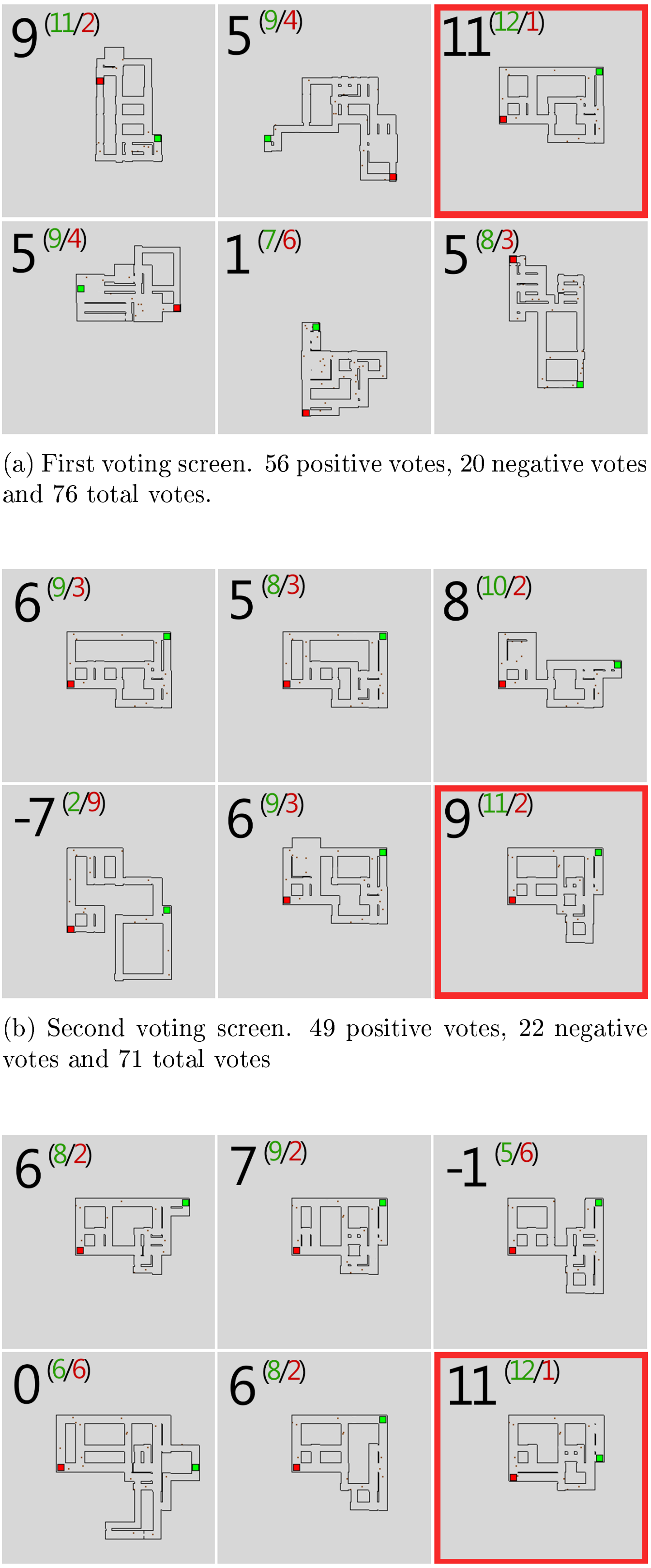
Table 6.1: An overview of all playtests, the names of them, the skills range of each and the player count.

When analyzing the data, correlations and standard deviations were calculated. Standard deviations are calculated using the formula in equation 6.1 where s is the standard deviation, x is the data point and $\overline{x}$ is the mean of the data set.

$$s = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} \qquad (6.1) \qquad\qquad r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{s_x * s_y} \qquad (6.2)$$

The correlation (r) is calculated using the formula in equation 6.2. Here the standard deviation s is used for the two sets of data being compared (x and y).

[1] These recordings can be found at http://www.peterolsted.com/thesis_recordings/



(a) First voting screen. 56 positive votes, 20 negative votes and 76 total votes.

(b) Second voting screen. 49 positive votes, 22 negative votes and 71 total votes

(c) Third voting screen. 48 positive votes, 19 negative votes and 67 total votes

Figure 6.1: Player votes for each map voting screen during pt3.1. In the top left of each picture we see the amount of positive (green) and negative (red) votes. The big numbers are the final scores, it is the sum of the positive and negative votes. The level with the highest score in each group was selected and is marked with a red outline. The maps seen in the first voting screen are randomly generated and have no parent.

6.2 Voting Satisfaction

According to figure 6.10 the average player placed over 100 votes in total, with roughly 58 positive and 46 negative votes throughout pt3. When we look at an example of a session in figure 6.1 we see which levels the players favoured during pt3.1, such as the most popular levels that received 11 points, and the least desired level, which received a score of -7 which is presumably due to its lack of complexity. Since voting data was only gathered during pt3, it is not possible to present any such data from pt2. All other data can be seen in appendix C.2. This alternative voting system provided the user with multiple votes, which should minimize dissatisfaction among voters [Cro97], and with an increase in perceived impact from voting from 4.23 in pt2, to 4.85 in pt3 and 85% of testers agreeing that their votes made a difference in pt3, it appears that voters were more satisfied, despite a bigger group of voters. Something we noticed for the votes on figure 6.1 was that the order remained similar, even if the negative votes were not included, and most importantly the best maps would still have the most votes in either case.

6.3 Fitness from User Preference

An interesting observation was that, despite a wide range in FPS experience, the groups were able to agree on a direction that the level design should take. Players with skills ranging from 2-6 in pt2.2 and 1-4 in pt2.3 still managed to reach consensus, without the players explicitly agreeing on any design. The skill were measured in the range of 1-6, where 6 is highly skilled.

In pt2.2 we saw a clear preference for open spaces in their questionnaire responses (figure 6.3), and in pt2.2's last level (figure 6.2) we clearly see how the level ended up being more open space than corridors, which would fit with their preferences. In fact in the discussions afterwards (at 6:20 minutes) we hear a tester mention the last level being particularly fitting with his preferences. The same happened for the evolution of pt2.3 (figure 6.9), where we see that even though the level did not evolve to something complex that fit well with TGE, it evolved small simple levels. This was line with what the players preferred according to their answers on what constituted a good level (figure 6.5). This would indicate that the evolutionary approach worked and that the group's preference became the better fitness. The last level is even mentioned as a good example *"...the last map we played was a lot of fun..."*, figure 6.5 shows the map evolved.

We see in the questionnaire answers in figure 6.3 and 6.5 that the final levels matched the preferences of players when they had played enough opportunities to evolve the level. Pt3 still matched the descriptions, although not as accurately, possibly due to only a few maps was generated before the test suffered from network issues.

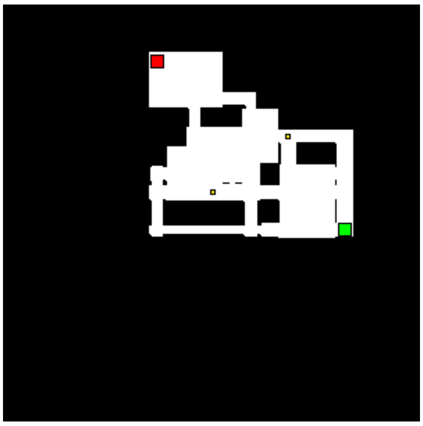


Figure 6.2: Last map of pt2. See appendix B.5 for the full lineage.

What seemed to constitute a good map?
Good mix of **open** and closed spaces, distance between bomb points, multiple paths so gameplay doesn't become monotone.
A good mixture of **open areas** (with some stuff to hide behind), choke points and long corridors. Also multiple ways to get around the map is good for keeping the match from becoming a stalemate
No long corridors. Bomb places placed in opposite sides of the map. Lots of small chunks of block in **open areas** that would allow you to take cover easily.
Not too long pathways to arrive at a bomb site. Only at one certain point, did arriving to bombsite B take way too long path which was difficult to arrive at. **Open spots in the middle of the map** were fun because it created a battlefield rather than just corridors.
Different paths that lead to **big areas**, so you can plan from where to attack the enemy and surprise them. **Big areas** with more covers, to strategically hide and take cover. Bottle necks where you "force" the player to meat the opposite team so that they will play against each other rather than going around trying to find people to shoot at.

Figure 6.3: The answers for pt2.2 on what constitutes a good map. Notice that every tester mentions open areas and how their last level contain a lot of open areas in figure 6.2. See appendix D for the full questionnaire.

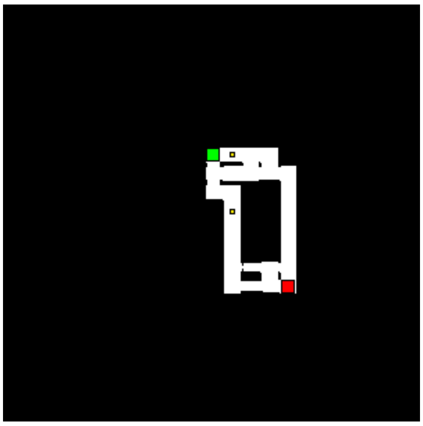


Figure 6.4: Last map of pt2.3. See appendix B.5 for the full lineage.

What seemed to constitute a good map?
I don't personally like the big open spaces. The long corridors and small alleyways are the best
I think to a good map design there are missing some vantage points, especially "lifted" or higher areas, infiltration from above, or from beneath were missing. Everything felt like a very long corridor. (Doom like)
There _were_ long corridors, but they could be avoided. It was more fun the more "enemy contact" we had, therefore **the smaller maps and the maps with more bottle necks were more fun/better.**
Basically speaking **uncomplex maps**. For example the last map we played was a lot of fun and very tense because it was **very small and basically one corridor that went all around the map**. That helped to predict where the enemy players might be. In the more complex levels it was basically running around without plan until you found an enemy

Figure 6.5: The answers for pt2.3 on what constitutes a good map. Notice how their preferences matches with their last level in figure 6.4. Full questionnaire can be found in appendix D

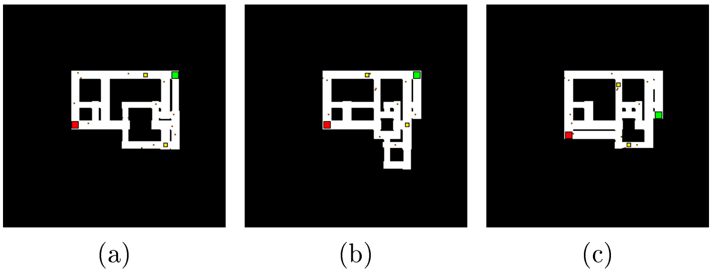
6.4 Complexity

Among the levels evolved for pt3 and pt2 four distinct scenarios can be seen in figure 6.6 to 6.9. The scenarios show different ways the level adapted to the player choices, like making a lot of open spaces, increasing the complexity or greatly simplifying the level.

Figure	Complexity					
6.6 (pt3.1)	17	18	18			
6.7 (pt2.2)	8	16	17	17	17	17
6.8 (pt3.3)	10	13	13	16		
6.9 (pt2.3)	21	18	19	17	13	9

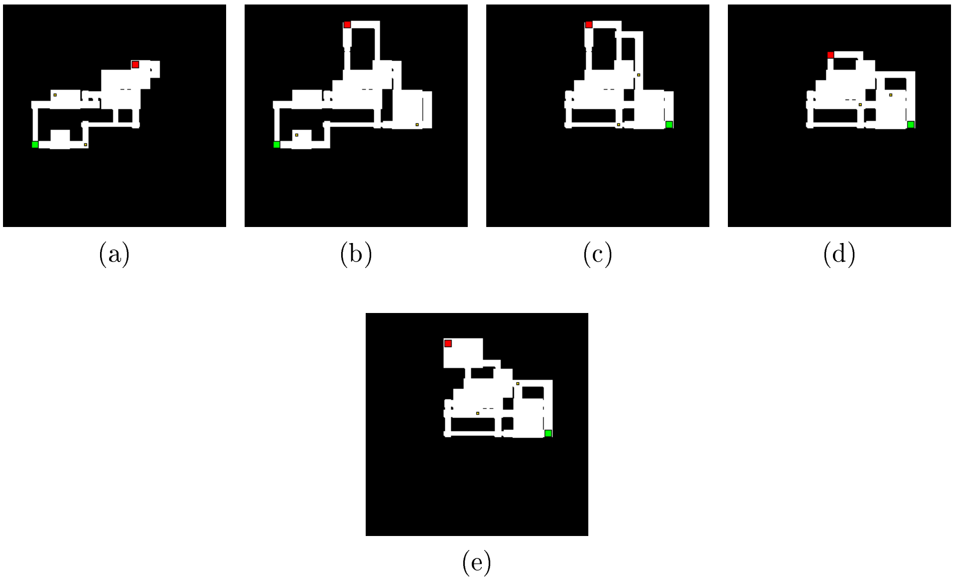
Table 6.2: The complexity for the evolved levels of figure 6.6 to 6.9.

As can be seen in figure 6.2, the map complexity (see section 5.5.3) for pt3.1 with 13 players remained steady throughout the test. Most others saw some change in complexity as they evolved to fit with the overall preference of the players. For pt2.2 we see a short increase in fitness after which it settles at 17. What we however can not see from these numbers, is the increase of open areas in this particular playtest, because the complexity metric only takes into account the amount of three-way connection in a level. The complexity of open areas is measured by the number of entrances, meaning that open areas do not necessarily change complexity significantly. For pt3.3 and pt2.3 we see a continuous change throughout the sessions, with respectively upwards and downwards trend as seen in figure 6.8 and 6.9.



(a) (b) (c)

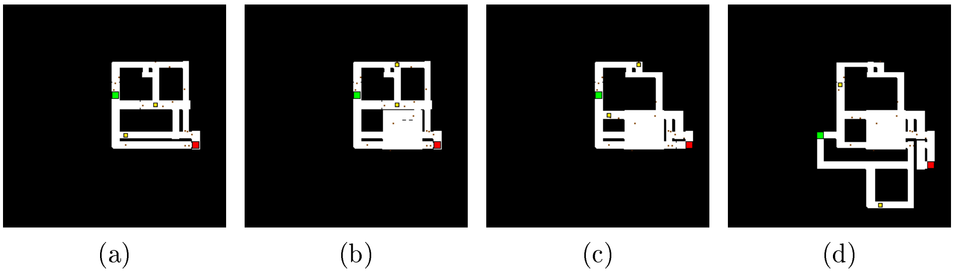
Figure 6.6: Maps for playtest three, group of 13 players. Note the relatively stable complexity (pt3.1).



(a) (b) (c) (d)

(e)

Figure 6.7: Maps for playtest two, second group with 5 players (pt2.2). Note the increase of open areas.



(a) (b) (c) (d)

Figure 6.8: Maps for playtest three, seven unskilled players (pt3.3). Note the increasing complexity.

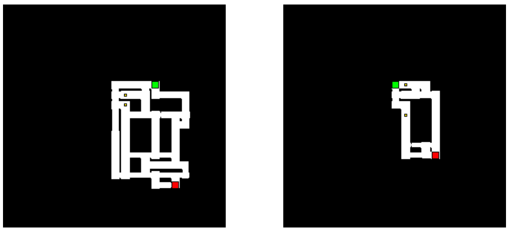


Figure 6.9: The first and last map of playtest two without the three intermediate maps, third group of five low-skilled players (pt2.3). Note the decreasing complexity.

Question	Average	Median	Standard Deviation	How would you rate your overall gaming skills?	How would you rate your FPS skills?	How would you rate your skills in this game?	Did you enjoy the game?	KD Ratio
How many players were in your playtest?	6,54	7	0,50	-0,28	-0,50	-0,37	-0,53	-0,59
How would you rate your overall gaming skills?	4,62	5	1,27		0,46	0,45	0,46	0,25
How would you rate your FPS skills?	3,15	4	1,56	0,46		0,89	0,53	0,81
How would you rate your skills in this game?	3,46	4	1,78	0,45	0,89		0,61	0,88
Do you enjoy FPS games?	3,62	3	1,64	0,33	0,68	0,69	0,43	0,52
Did you enjoy the game?	4,54	5	1,39	0,46	0,53	0,61		0,58
Did the game accurately portray an FPS game?	100%	1	0,00	0	0	0	0	0,00
Was the movement in the game what you would expect of an FPS game?	85%	1	0,36	0,04	0,18	0,47	0,01	0,33
Was the shooting in the game what you would expect of an FPS game?	92%	1	0,27	0,14	0,21	0,40	-0,10	0,22
Did you use the bomb planting/defusing?	2,54	3	1,28	-0,11	0,15	0,23	-0,08	0,06
Did you feel the game was fair?	69%	1	0,46	0,58	0,49	0,45	0,38	0,36
How sure were you of your objective?	3,38	3	1,90	-0,16	0,26	0,17	-0,17	-0,03
How easy was it to find your way through the levels?	4,31	5	1,38	0,37	0,30	0,47	0,55	0,37
Was the game experience what you expected, based on the picture preview?	4,62	5	1,55	0,39	0,57	0,71	0,45	0,54
Did you use the map-view to help find your way?	46%	0	0,50	-0,21	0,01	0,19	0,31	0,24
How varied did you feel the generated levels were?	4,00	4	1,36	0,36	0,36	0,54	0,69	0,42
How would you rate the overall quality of the maps?	4,31	4	0,91	0,43	0,40	0,43	0,60	0,45
Did the quality of the levels improve as the test session progressed?	77%	1	0,42	0,41	0,64	0,55	0,87	0,48
Did the quality of the levels decreased as the test session progressed?	0%	0	0,00	0	0	0	0	0,00
Did you feel voting could impact the bomb positions?	85%	1	0,36	0,71	0,45	0,47	0,78	0,30
Did you feel voting could impact the spawn positions?	69%	1	0,46	0,58	0,39	0,45	0,38	0,25
How much did the map's layout impact your enjoyment of the game?	4,46	5	1,34	0,29	0,08	0,07	0,61	0,05
How varied did you feel the evolved levels were?	3,85	4	1,17	-0,35	0,10	0,11	0,00	0,01
How much impact did you feel the voting system had on the overall quality of maps?	4,85	5	1,29	0,29	0,01	0,10	0,56	0,05
How would you rate the best level that you played?	5,00	5	0,78	0,39	0,13	0,33	0,70	0,10
Did you win most of the rounds played?	2,92	3	1,27	0,36	0,67	0,80	0,76	0,85
Did you feel your votes made a difference?	85%	1	0,36	0,71	0,45	0,47	0,78	0,30
Kills	30,85	28	14,73	0,21	0,78	0,84	0,54	0,96
Deaths	29,69	30	5,38	-0,22	-0,71	-0,72	-0,58	-0,75
KD Ratio	1,13	0,82	0,69	0,25	0,81	0,88	0,58	
Sum of the map complexity voted on positively	885,15	910,00	222,72	0,01	0,22	0,42	0,02	0,48
Number of positive votes	58,08	59,00	14,49	-0,03	0,18	0,38	-0,04	0,44
Average complexity of maps voted on	15,24	15,17	0,40	0,31	0,25	0,21	0,50	0,27
Sum of the map complexity voted on negativly	694,92	684,00	220,23	0,47	0,06	-0,08	0,49	-0,16
Number of negative votes	45,62	47,00	14,24	0,48	0,06	-0,06	0,48	-0,16
Average complexity of maps voted negativly on	15,23	15,36	0,51	-0,03	-0,01	-0,13	0,10	0,00

Figure 6.10: Quantitative questionnaire answers from pt3. Includes recorded kill-death ratio and votes. Pt2 answers can be seen in appendix section B.

6.4.1 Players

Player Satisfaction

In pt2 we only asked whether or not the maps quality increased over the sessions, where we in pt3 also asked if they got worse. Not a single participant felt the levels got worse. A majority of players stated that the quality improved, despite having as many as 13 simultaneous players voting, and that is backed up by the rating data seen in table 6.3, where all level received a majority of positive ratings. These positive ratings suggest that progressively likeable levels were created, even with many people voting at the same time.

	Votes	Better	No difference	Worse	Average	Better %	Worse %
Big	14	7	3	4	0,21	50%	29%
Skilled	8	7	0	1	0,75	88%	13%
Less-skilled	9	3	5	1	0,22	33%	11%
	8	5	2	1	0,50	63%	13%

Table 6.3: Ratings from pt3. They rated if the levels improved or worsened. Due to a networking issue, there is unfortunately an extra unidentifiable rating in each line, resulting in a slightly skewed result. A vote is respectively 1, 0 or -1 depending on if its 'Better', 'No difference' or 'Worse'. Hence, 'Average' is the average of the ratings.

Player Skill

The first questions in the questionnaire are regarding the quality of the game itself, and not the maps. To make sure the game quality did not result in negative answers, we need to know that their opinion of the game mechanics. As we can see in the answers in figure 6.10 and appendix C.2, all but one player of the 19 unique participants across pt2 and pt3 had their expectation met for an FPS game, and all but one player agreed that the controls and shooting mechanics were up to par. This makes it unlikely that the played has a negative bias towards the levels based on the quality of the gameplay.

Along with game quality data, the skill level of our players affect their opinion of the level and game. Looking at their own perceived skill levels as well as their measured skill level, allows us to look for correlations in the answers in relation to fun and skill.

Throughout pt2 and pt3, the skill levels ranged from the minimum of 1 to the maximum of 6, with a high correlation to the general enjoyment of FPS games. For our questionnaire data, we have included the correlations with their skill levels, a full overview can be seen in figure 6.10]. It is clear that there is a correlation between the skills of the players, in the form of kills in the game, and their answers about their skill level. We also was a difference in round duration based on skill, which can be seen in table 6.4, where pt3.2, that consisted of much more skilled players than pt3.3, saw a lower average round duration. We can also see that the preliminary rounds had much longer rounds

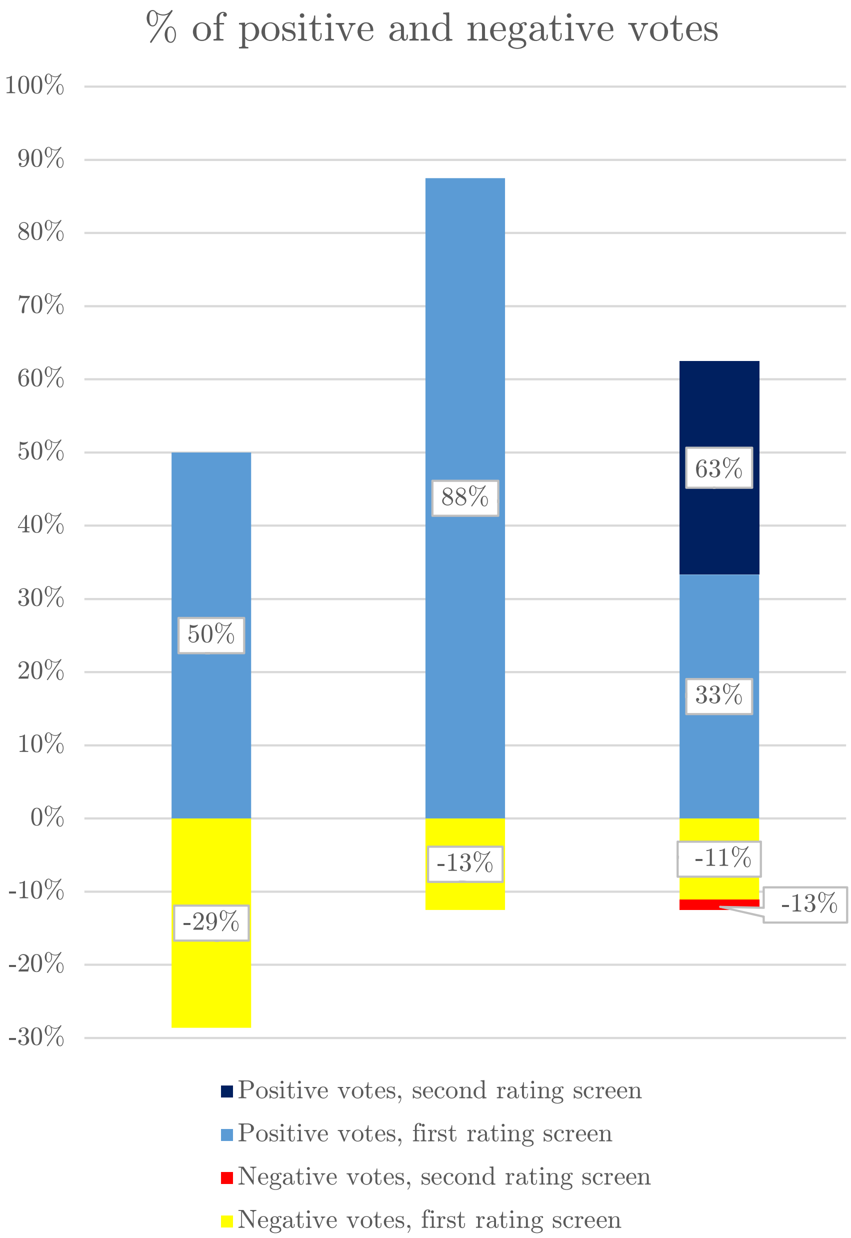


Figure 6.11: Player ratings for playtest 3. In order, big group, skilled, unskilled players. Third column shows the two rating screens on top of each other (not the sum) while the first two only reached a single rating screen in the playtest.

Playtest	Average round duration
pt3.1 preliminary	72 seconds
pt3.1	54 seconds
pt3.2	38 seconds
pt3.3	43 seconds

Table 6.4: Average round time during playtests.

than pt3.1, despite having fewer player, due to players not being familiar with the game at this point.

For pt3 we see in the answers that those who did not consider the game fair stated it was due to their own skill level, and made no mention as to the game itself being unfair. See appendix E for the full answers. Based on these observations, it is unlikely that the gameplay and controls caused a negative bias towards the ratings and enjoyment of the game. It appears that less skilled players enjoyed the experience less due to playing against people of greater skill or simply not enjoying FPS games, which had an impact on their answers in the questionnaire, and possibly their behaviour during voting.

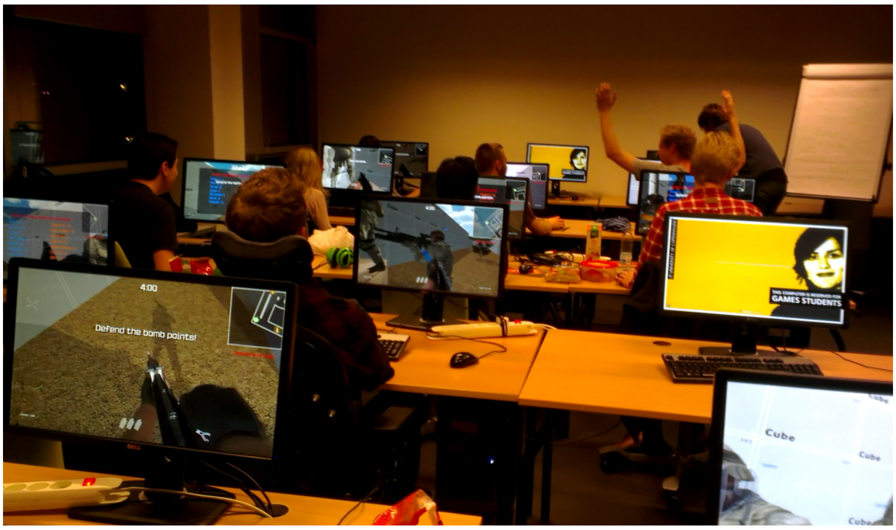
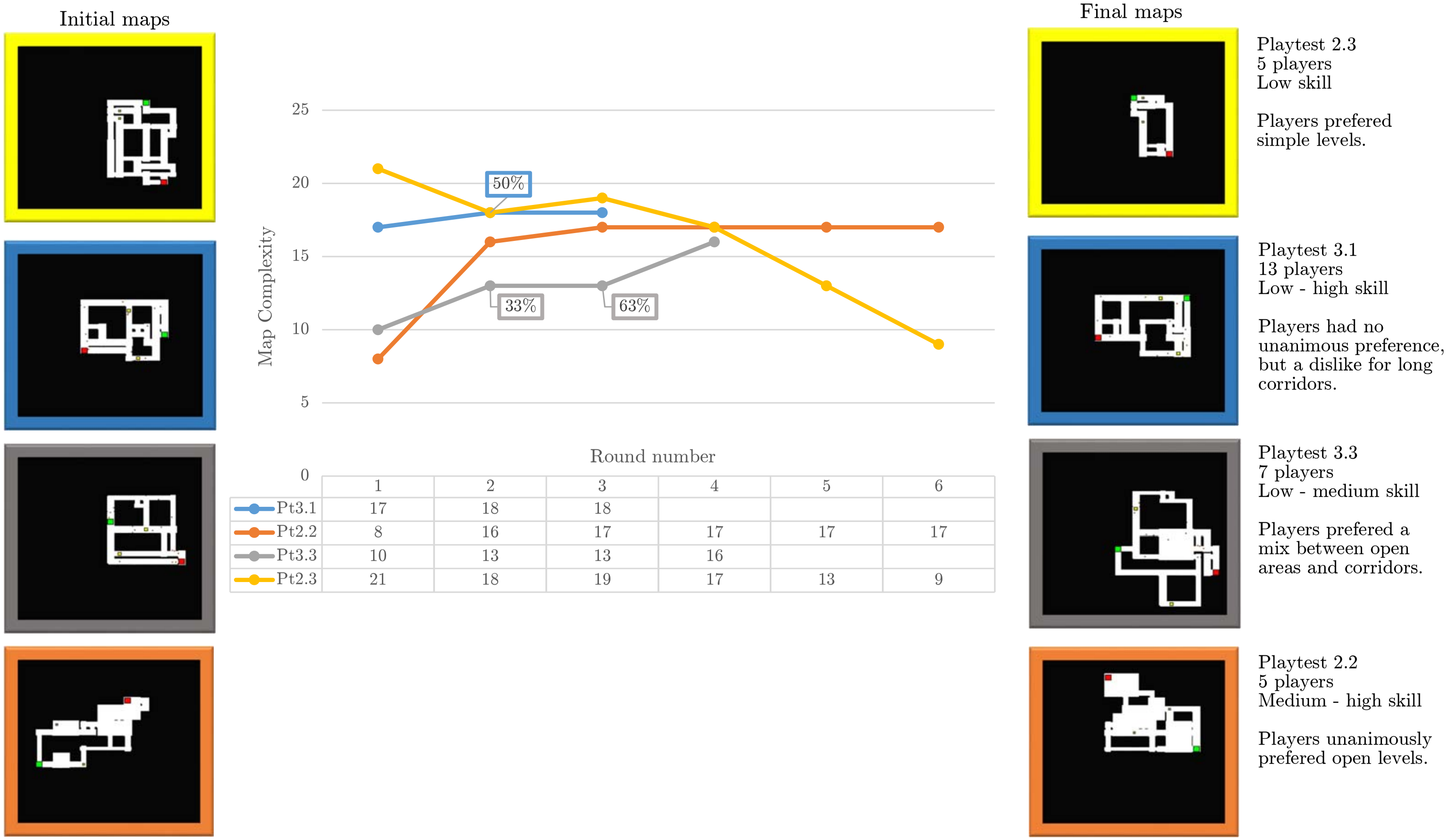


Figure 6.12: Group of 13 players during playtest 3.

Figure 6.13: Figure showing the change in complexity for four playtests. Pictures on the left and right hand side shows the first and last map of each playtest respectively. The marked percentages shows the agreement among players that this map was better than the previous. The lack of labels on the other lines is due to missing data. The main result is that groups of players were able to evolve their maps to match their preferences.



	1	2	3	4	5	6
Pt3.1	17	18	18			
Pt2.2	8	16	17	17	17	17
Pt3.3	10	13	13	16		
Pt2.3	21	18	19	17	13	9

Playtest 2.3
5 players
Low skill

Players prefered simple levels.

Playtest 3.1
13 players
Low - high skill

Players had no unanimous preference, but a dislike for long corridors.

Playtest 3.3
7 players
Low - medium skill

Players prefered a mix between open areas and corridors.

Playtest 2.2
5 players
Medium - high skill

Players unanimously prefered open levels.

7. Discussion

7.1 Map Quality

One of the main questions for these tests were whether or not the levels improved over time. This means that the players' choices should give them better suited levels, and according to the numbers for pt2 and pt3 this was indeed the case. The overall quality was an average of 4.07 and 4.31 in the two tests on a scale of one to six, with a standard deviation of 1.07 and 0.91 respectively. 61% and 77% agreed that the levels increased in quality during the test sessions. In pt3 we also asked whether or not the quality decreased and not a single of the testers stated it decreased in quality. The best levels were rated as 4.8 and 5 on average respectively ($sd = 0.97$ and 0.78). Looking at the qualitative answers and the maps developed, see section 6.3, we deem that the quality of the levels did increase over time.

In the qualitative answers players expressed that there had been a lack of verticality in the levels, and that they felt quite flat, unlike levels in Counter-Strike and Call of Duty, where the environment is not entirely flat. Since the approach for our levels can not create vertical features, some players had not had their expectations met completely, where we might have seen higher ratings of maps if vertical traversal was also implemented as part of the level layout.

7.1.1 Selection

The voting system for pt2 and pt3 changed between the playtests as explained in section 5.3. Instead of a single voting screen determining the layout, bomb and spawn positions, the bomb positions are now on a second voting screen. Comparing the results of pt2 and pt3 we see that for the question 'Did you feel voting could impact the bomb positions?', most players in both tests agreed. In pt2 23% of players did not feel they could impact the bomb positions, while only 15% in pt3 felt the same way. We expected that all players would feel their vote could had an impact on the bomb positions. The fact that not all players agreed could be due to the six variations presented where too similar to one another. As the bomb positions had to be placed closer to the defenders, it did only leave about half the map for the bomb positions, and thus did not leave a lot of possibility for variations.

7.1.2 Player Enjoyment

In pt3, figure 6.10, we can see a strong correlation between the skill levels and enjoyment of the game. Correlation values range from 0.46 to 0.61, and the personal perceived impact with correlation values between 0.45 and 0.71. It is possible that a bad experience with the game is reflected the perceived impact when voting. When we look at the individual answers, we see that it is the same players that respond negatively to impact on spawn points and "Did you feel your votes made a difference?". These testers in pt3 also had the lowest FPS skill levels of 1 out of 6, which was not something exclusive to pt3, but playing in the bigger group with 13 players in pt3.1, might have been a negative experience since there were multiple skilled FPS players participating. It is suspected that mixing players with extremely low skill levels and high skill levels provide less enjoyment for the weaker players, because the numbers for enjoyment and skill show no correlation in pt2 (appendix D), where the range in skill and group sizes were smaller.

7.2 Additional Test Result Factors

Altering Gameplay

For pt3 we made the decision of adding a long range weapon. This however might have increased the difficulty of the game, since experienced players now had more tools at their disposal, while unseasoned players were still left struggling with the default weapon, resulting in a bigger skill gap. From the responses in appendix D, most players perceived levels which contained long corridors as inferior, where only a single pt3 tester made any mention to this. It is possible that the inclusion of the long ranged weapon gave new purpose to these long corridors, that had none in pt2, and this change in game design meant that the preferences of the players changed. This was included due to the number of players in pt2 complaining about long corridors, which is something that is not necessarily considered negative in Counter-Strike and Call of Duty. Had this not been included due to sniper rifles being standard in almost all FPS games, we might not have included it in pt3, since it is unknown what other implications this change could have had.

7.2.1 Player Satisfaction

In terms of overall player satisfaction, the numbers appear to be quite positive. Almost all players agreed that the game was representative of an FPS, and the enjoyment of the game was high in pt3 and pt2, with an average of 5.15 and 4.54 in the two tests on the scale of 1-6. These data had a standard deviation of 0.66 and 1.39 and a median of 5, which means that players tended to rate their enjoyment on the positive end of the scale. When we look at the correlation with other questions, we can see where enjoyment had an effect on the data, which could tell us the factors that influenced lower ratings in enjoyment.

First off we can see that there is a clear link between the performance of the player, and the enjoyment of the game. Interestingly we can also see a clear correlation between the enjoyment and the quality of levels. In the overall quality, best level and whether or not the quality improved, all show very high numbers for people who enjoyed the game.

7.2.2 Map Complexity

As we can see in figure 6.10 the average map complexity voted on was 15.24, with complexity referring to the amount of three-way intersections in a level. Although simply counting three-way connections does not fully encompass the complexity of a level, we do see a correlation between the skills of the player and the complexity of the levels they voted on, which suggests that more experienced players prefer more complex layouts. Surprisingly though the biggest correlation was between complexity and enjoyment, and not the performance in the game, which we can find no reason for. This could mean that understanding the mechanics of the game, which might be indicated by enjoyment, encourage complex levels, which again is a matter of skill, except we did not see the same correlation for any skill measurements.

7.3 Playtest Approach

We sometimes see widely different results between pt2 and pt3 in places such as enjoyment of the game, and in some of the correlations. It is difficult to conclude why, but we suspect that our approach to the two playtests could have had an impact. For example in pt2 the three groups were all separated, unlike pt3 where all players would first play as a group. During pt3.1 that in part was used to determine the players' skill level, an imbalance appeared early on in the team arrangements. These imbalanced teams could have resulted in a bad experience for the weaker players who would lose more often. We can see that the round duration (seen in table 6.4) was longer in pt3.1 and pt3.3, and that means that players that died early had to wait for longer periods of time. This along with a lower average FPS skill level and bigger groups, could result in less enjoyment for the weaker players, as they had to wait for the more skilled players.

Another big difference in the two playtests was the delivery of instructions. For pt2 we had printed instructions, and placed them at each computer. In pt3 we had instead visual instructions in the menus of the game, which some players missed when entering the game. The instructions for the voting screens were delivered orally in both tests, but having a larger crowd of people was more difficult than anticipated, with some testers having overheard important information.

7.3.1 Sample Size

In pt2 we had groups of four to five people playing, each divided in two teams, where pt3.1 had 13 players split into two groups. While this provided us some good conditions to test the voting system, it might have changed the experience during gameplay. As mentioned, the round duration was longer in pt3.1 and pt3.3 due to their inclusion of more, and less skilled player, but we see that the durations (seen in table 6.4) in pt3.1 and preliminary pt3.1 are substantially higher than those in later tests.

In terms of levels, when we look at the generated levels for pt3, they have noticeably fewer generations per session. This makes it harder to evaluate the evolutionary progress and patterns in the lineage, as the testers did not get the same amount of opportunity as in pt2 to vote. Had it not been for network overload issues during the playtests, it is possible that we might have seen better levels, in line with what we saw during pt2, since increasingly as the lineages progressed, but the lineages in pt3 were shorter.

A higher sample size would have been preferable to further verify the results with more than the 19 unique participating players. We do however deem that the sample size was large enough to verify that the presented approach resulted in good and improving levels.

7.3.2 Test Audience

In both tests we invited a mixture of people with different backgrounds and skill levels with all participating players having an interest in games. A few players had very little interest in FPS games in general, but everyone were familiar with how to play one. During the play test we only distinguished between skilled and low-skilled players. Some players had a background in game development and some with level design experience. Some testers were skilled FPS players but did not have any game development background, while others had a level design background with poor FPS skills. Some testers were also returning from pt2, and others had only tested the game once. Despite a mixture of player skills, the results still showed that any agreements on level preferences that players had would eventually be achieved, but tests with very specific skill levels or preferences might have yielded much more rapid evolutions and perhaps extreme levels.

7.3.3 Exclusion of an outlier

After pt2 we decided to exclude a single tester's results on the basis of being deliberately dishonest. The tester, who had been invited due the person's skill and experience with gaming had filled out the questionnaire dishonestly. This could be seen as the person had responded with having the lowest possible skill levels, and all other answers were scored as negatively as possible. Textual answers were also mostly irrelevant to the questions. Had this person's actual gaming skills not been known to us beforehand, we would still exclude the data point due to the complete lack of FPS experience expressed, since this person would not be able to draw any comparisons to other FPS games, and a prerequisite for testing was having previously played FPS games. Another tester also rated his/her own skills at the minimum level, but through speaking to the tester who had experience with FPS games, we did not deem the answers untruthful, since the tester showed low performance during the playtest, which was not the case with the outlier.

8. Future Work

Although we have been able to come to a conclusion on our initial questions, we still have unexplored approaches, and many new questions arose during research and development. This chapter is dedicated to how our work can be built upon in the future.

8.1 Abstract Rule-based Generation

As mentioned in chapter 2, formal language (2.3.1) could be an alternative approach to create meaningful levels. This was not used due to the difficulty of generating cyclic levels that would translate into valid three dimensional geometry. If such an approach could be realized, it would be possible to better reason about how good levels are composed. Using online crowds, large data sets could be gathered and analysed for patterns, which in turn could be used for machine learning.

8.2 Breeding

A different direction our evolutionary process could have taken, was to evolve levels based on multiple parents, instead of simply selecting one. While cross-breeding is not a new approach, it could be interesting to look at the satisfaction of maps if all votes have an influence on the evolutionary progress, and not simply a winner-takes-all solution. It could possibly better balance the different player preferences.

8.3 Evolution using Huge Crowds

In the end our prototype was only ever tested using local networks. It could have been interesting to look at a similar problem utilizing a larger user base; the internet. Not only could this garner different results due to the lack of physical presence with other people in the crowd, but could also provide a substantially larger data set for further analysis. The lineage of maps evolved was quite shallow, with the longest lineage being six maps. Playing online could evolve a map more times than it would be possible be during local play. Here the evolution would also have to deal with a potentially changing group of players, where the overall preference might change from round to round.

8.4 Automatic Contextual Evolution

An idea for the project, we never got time to look at was the option of determining the maps' fitness solely on the performance and statistics of the level. This also tied in with using online crowds, to generate better statistics, that could then be used to evolve and improve the maps. This would remove the players' knowledge of level design from the equation. This would especially be suited for games where The Good Engagement is the goal, and players can not be expected to have sufficient knowledge of level design.

9. Conclusion

We have presented a novel approach to creating high quality multiplayer FPS maps using collaborative interactive evolution and procedural content generation which adapts quickly to a group of players' preferences. The approach consists of building a high number of maps using a novel approach and selecting a small sample of the maps with the highest fitness according to our rules based on The Good Engagement, for the players to choose between. By rating the levels using simple votes, a group of players can rank the levels, and provide players with the map that most players would prefer. The players would play the map and vote on a new map from a set of mutated levels based on the just played map. Results indicated that players considered their voting impactful, even in groups of up to 13 voters.

Interesting levels structures that facilitates good engagements are accomplished by placing L-shapes on a map, fitting them into a grid and combining nearby sections of the L-shapes with each other. This creates a grid of connections between the cells. The connections are then used to carve out corridors onto more detailed features are added. Map, spawn- and bomb positions are placed on the final level structure.

Across a wide range of player skill levels, almost all players reported the levels were of higher than average quality and increased in quality as they played the game. We found evolved level matched closely the player preferences. The players expressed enjoyment with the game, though it is unlikely that this approach will make non-FPS players want to play it extensively since we found a strong correlation between their enjoyment of this game and FPS games in general.

Our approach provided the different groups of players with distinct types of maps. The structures ranged from open to closed spaces and from simple to complex levels. Almost every player agreed that the maps improved in quality and no-one stated that the maps decreased in quality as they played.

For future research, machine learning could be an interesting direction to take for the generation of levels. This could possibly be combined with a game suitable for online play that would generate a much larger amount of data. This could also display how well the system adapts to a changing group of players that can enter and leave the game throughout a session.

Bibliography

[CE12] Valve Corporation and Hidden Path Entertainment. *Counter-Strike: Global Offensive*. Valve Coorporation. 2012.

[CLL11a] Luigi Cardamone, Daniele Loiacono, and Pier Luca Lanzi. "Interactive Evolution for the Procedural Generation of Tracks in a High-End Racing Game". In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM. 2011, pp. 395–402.

[CLL11b] Luigi Cardamone, Daniele Loiacono, and Pier Luca Lanzi. "Interactive evolution for the procedural generation of tracks in a high-end racing game". In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM. 2011, pp. 395–402.

[CS12] Xiao Cui and Hao Shi. "An Overview of Pathfinding in Navigation Mesh". In: *IJCSNS* 12.12 (2012), p. 48.

[Car+11] Luigi Cardamone et al. "Evolving interesting maps for a first person shooter". In: *Applications of Evolutionary Computation*. Springer, 2011, pp. 63–72.

[Cor04] Valve Corporation. *Counter-Strike: Source*. Valve Coorporation. 2004.

[Cor98] Valve Corporation. *Half-Life*. Sierra Entertainment. 1998.

[Cro97] David Wallace Croft. *A Proposed Voting System Alternative: Multiple-Selection, Winner-Take-All*. `http://alumnus.caltech.edu/~croft/research/government/approval/voting.html`. 1997.

[Ext] Digital Extremes. *The Evolution of Warframe*. `https://warframe.com/news/evolution-warframe`.

[Ext13] Digital Extremes. *Warframe*. Digital Extremes. 2013.

[Gam11] Headup Games. *The Binding of Isaac*. 2011.

[Hin10] Philip Hingston. “A new design for a turing test for bots”. In: *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE. 2010, pp. 345–350.

[Int12] IO Interactive. *Hitman: Absolution*. Square Enix. 2012.

[Juu11] Jesper Juul. *Half-real: Video games between real rules and fictional worlds*. MIT press, 2011.

[LSS14] Antonios Liapis, Gillian Smith, and Noor Shaker. “Mixed-Initiative”. In: *Procedural Content Generation in Games*. 2014. Chap. 11, pp. 191–209.

[LYT13] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. “Sentient Sketchbook: Computer-Aided Game Level Authoring”. In: *FDG*. 2013, pp. 213–220.

[ME13] Anders Mousten and Michele Ermacora. “Procedurally Generated Levels for a AAA Game”. MA thesis. IT-University of Copenhagen, 2013.

[McM11] Edmund McMillen. `http://edmundmcmillen.blogspot.co.uk/2011/09/binding-of-isaac-gameplay-explained.html`. 2011.

[Moj11] Mojang. *Minecraft*. Mojang AB. 2011.

[Pat10] Amit Patel. *Polygonal Map Generation for Games*. `http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/`. 2010.

[Pik07] Allen Pike. *Modeling Plants with Lindenmayer Systems*. `http://www.allenpike.com/modeling-plants-with-l-systems/`. 2007.

[Sak] Alex Sakharov. *Formal Language*. `http://mathworld.wolfram.com/FormalLanguage.html`.

[Sec+08] Jimmy Secretan et al. “Picbreeder: Evolving Pictures Collaboratively Online”. In: *In Proceedings of the Computer Human Interaction Conference* (2008).

[Smi] “PCG-Based Game Design: Enabling New Play Experiences through Procedural Content Generation”. In: *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*. ACM. 2011, p. 7.

[TSN14] Julian Togelius, Noor Shaker, and Mark J. Nelson. “Introduction”. In: *Procedural Content Generation in Games*. 2014. Chap. 1, pp. 1–15.

[War03] Infinity Ward. *Call of Duty*. Activision. 2003.

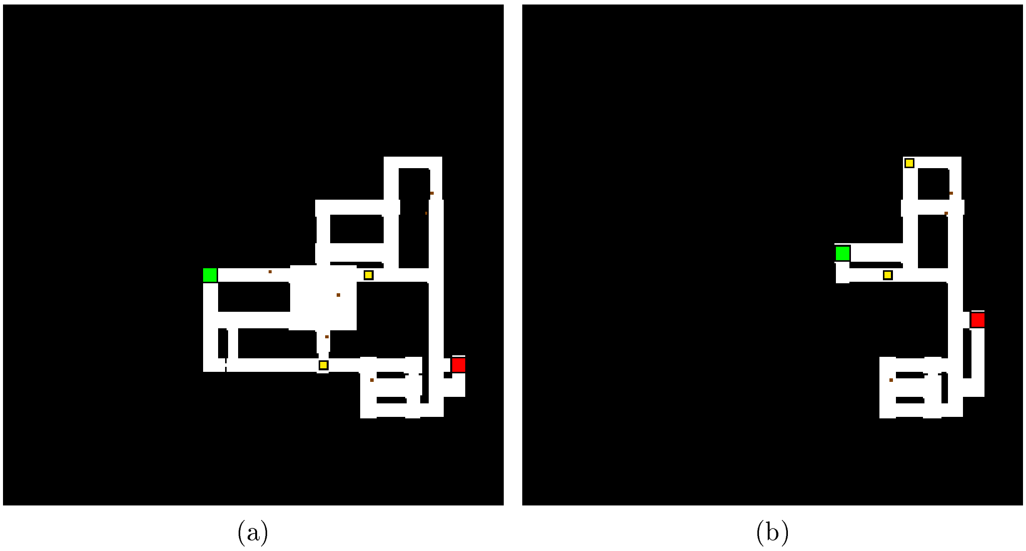
[War07] Infinity Ward. *Call of Duty 4: Modern Warfare*. Activision. 2007.

[War09] Infinity Ward. *Call of Duty: Modern Warfare 2*. Activision. 2009.

[Yu08] Derek Yu. *Spelunky*. 2008.

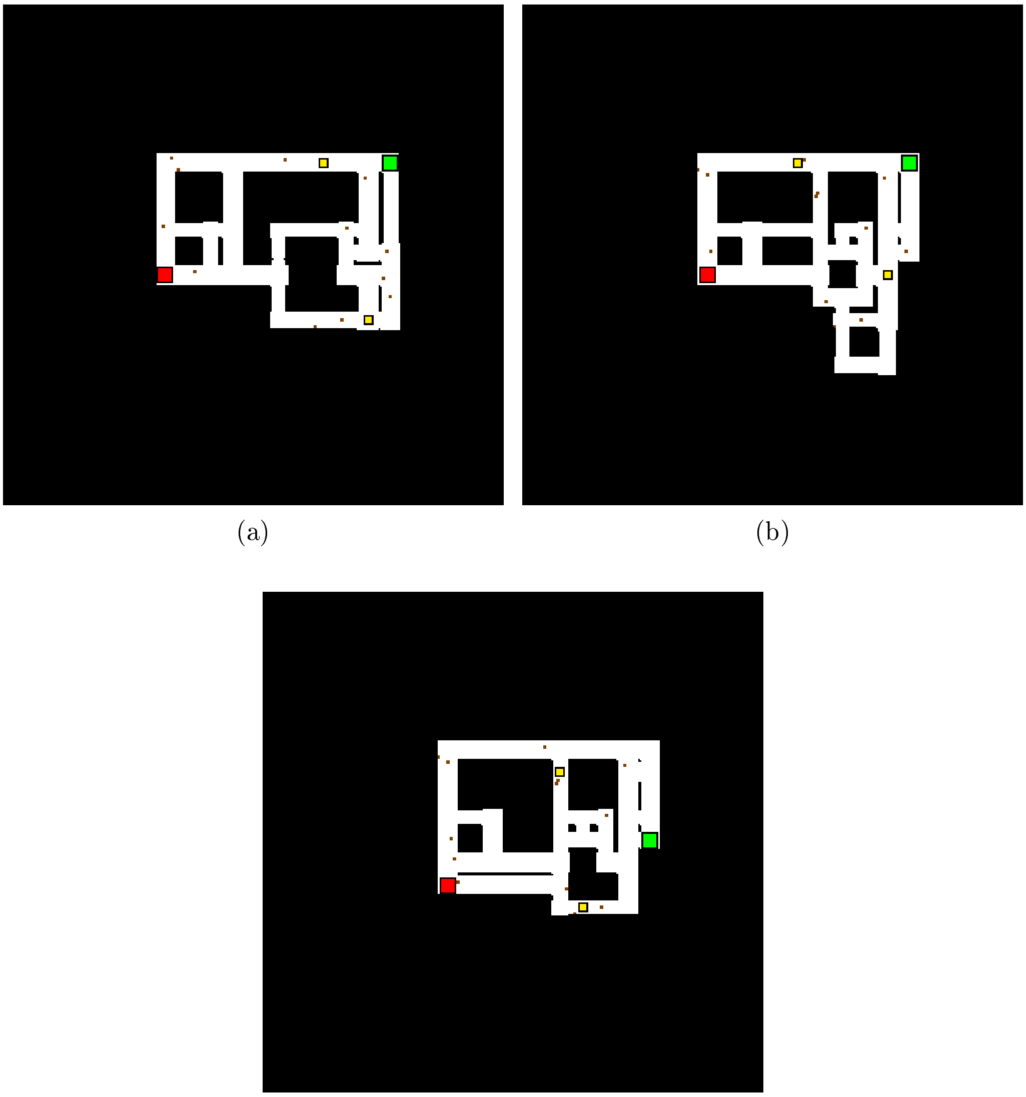
[lM12] Peter Ølsted and Benjamin Ma. "God Algorithm for Living Artificial Physics-based Adapting Creatures with Omnipotent Scalability". MA thesis. IT-University of Copenhagen, 2012.

A. Third Playtest Maps


(a)

(b)

Figure A.1: Lineage of maps for playtest three, preliminary. (pt3.1 preliminary)

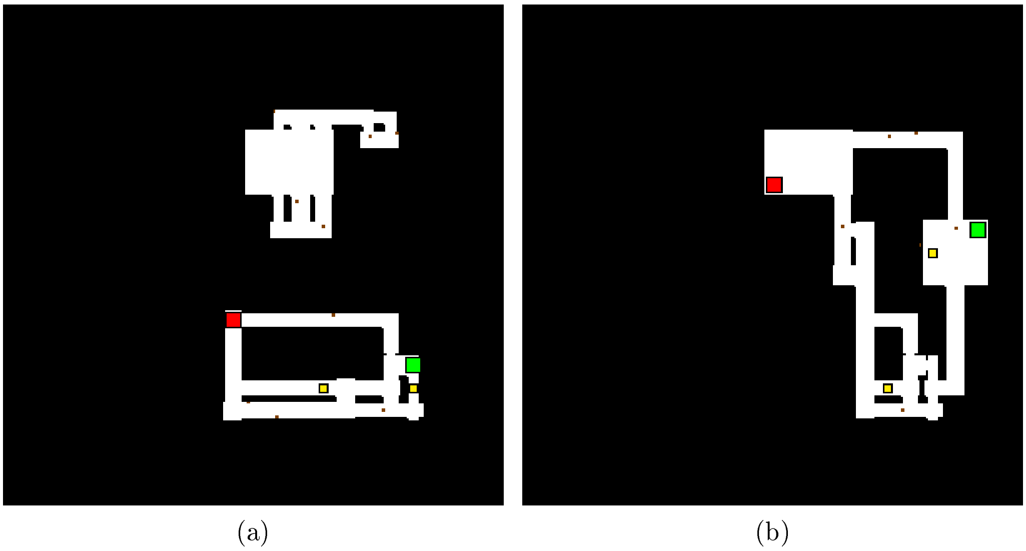


(a)

(b)

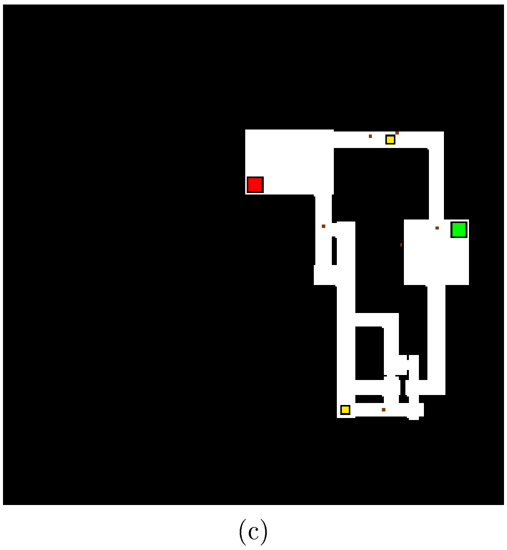
(c)

Figure A.2: Lineage of maps for playtest three, big group. (pt3.1)



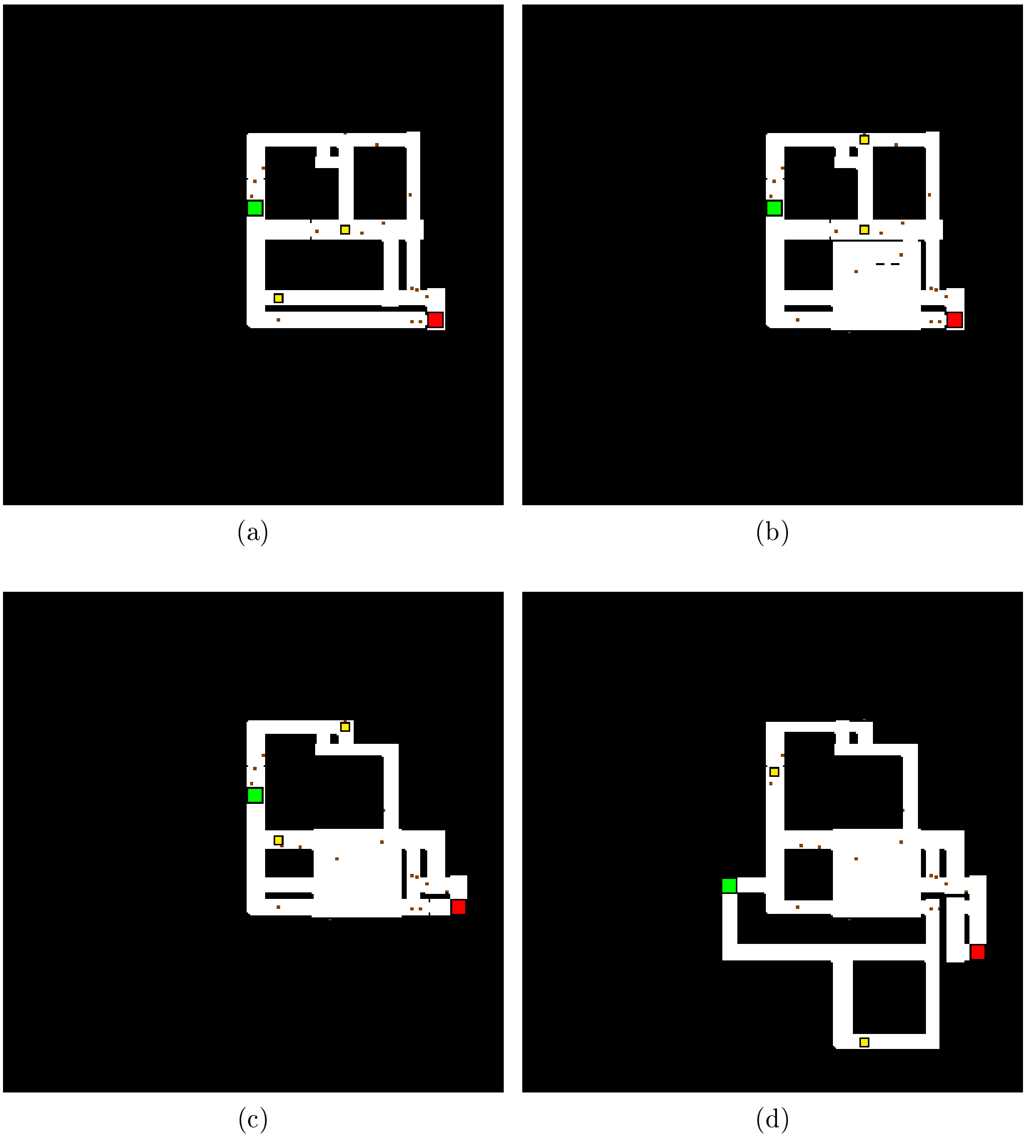
(a)

(b)



(c)

Figure A.3: Lineage of maps for playtest three, skilled players. (pt3.2). Note how part of the map is not reachable.


(a)

(b)

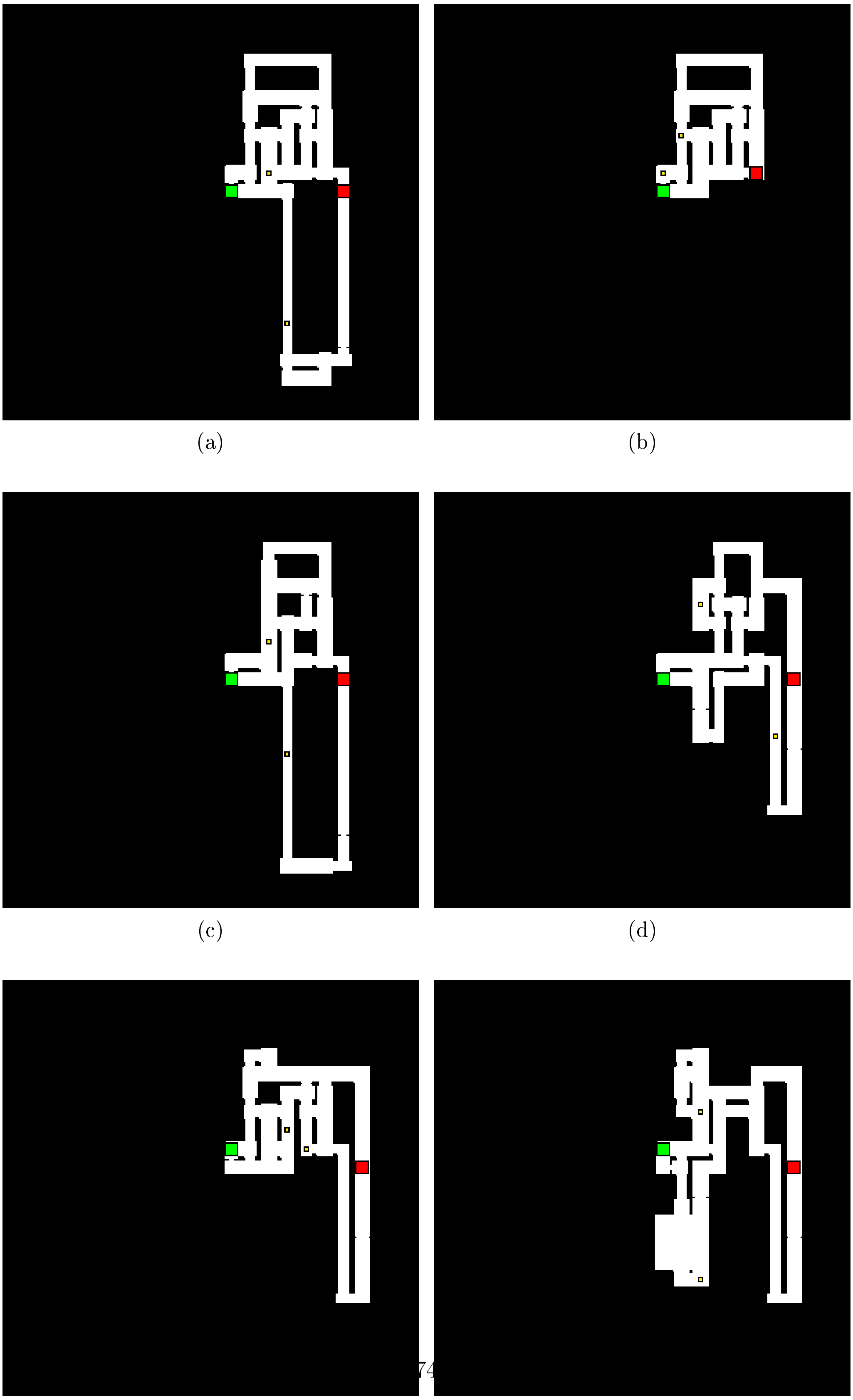
(c)

(d)

Figure A.4: Lineage of mapsaps for playtest three, unskilled players. (pt3.3)

B. Second Playtest Maps

This page is intentionally left mostly blank



(a)

(b)

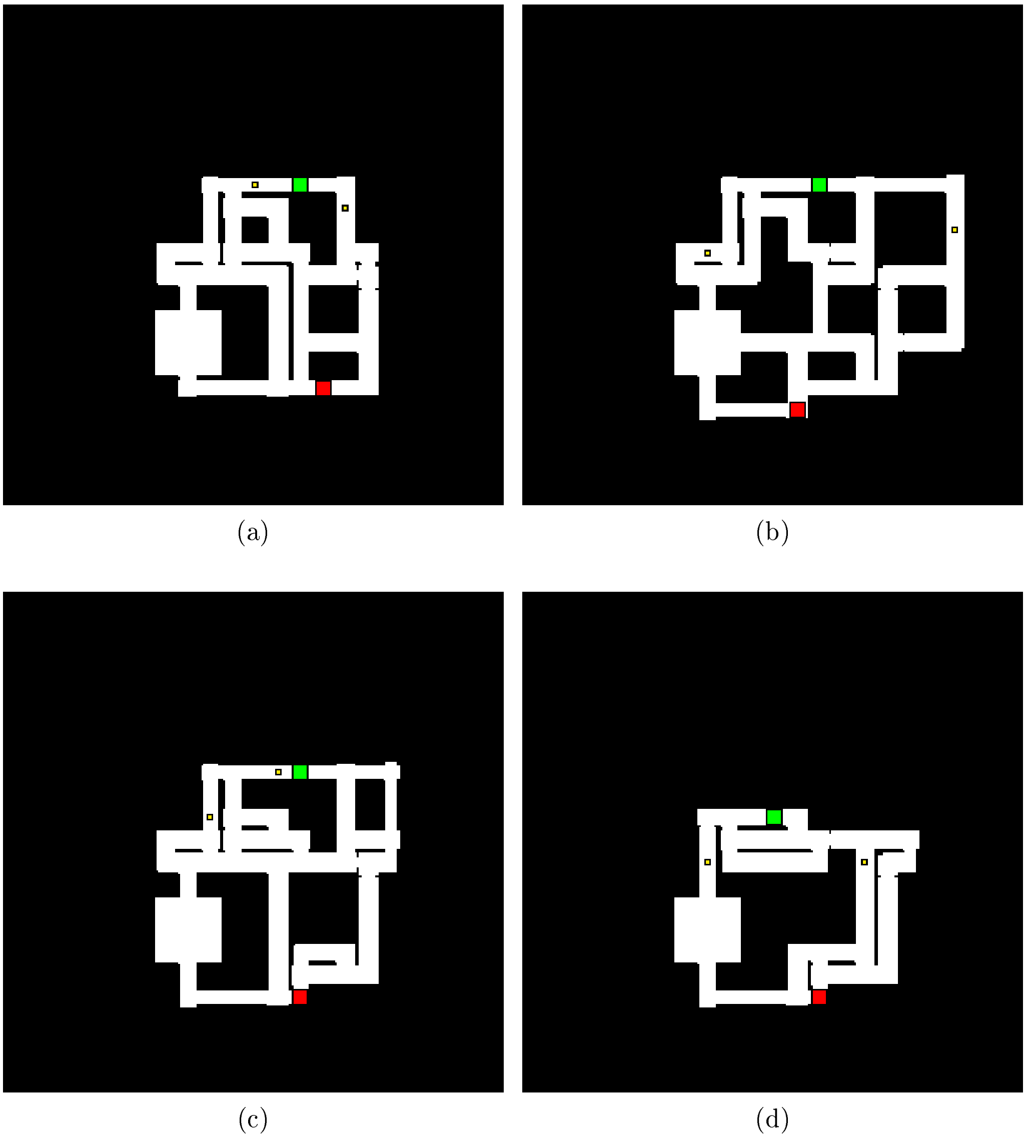
(c)

(d)

(e)

(f)

Figure B.1: Lineage of maps for playtest two, first group. (pt2.1)



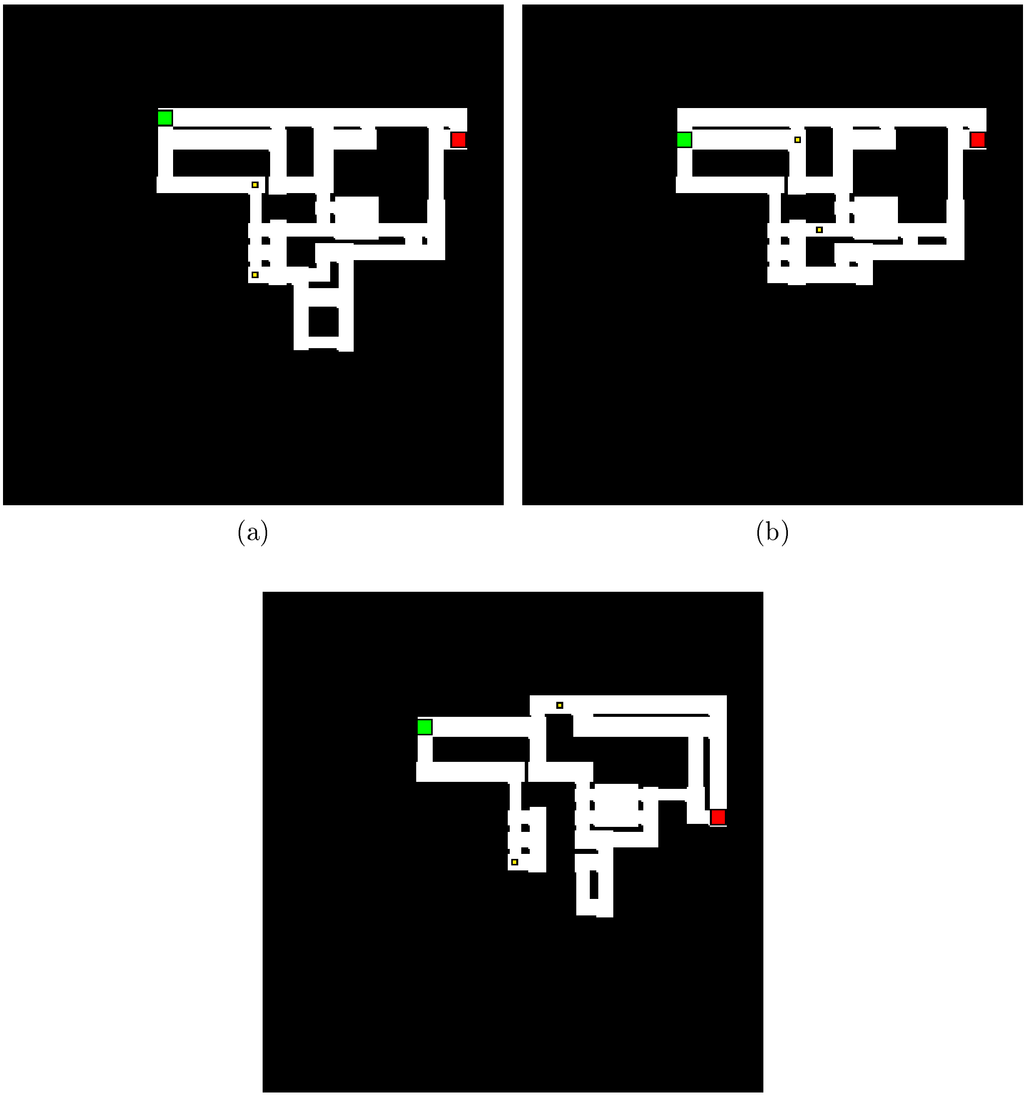
(a)

(b)

(c)

(d)

Figure B.2: Lineage of maps for playtest two, first group. (pt2.1)

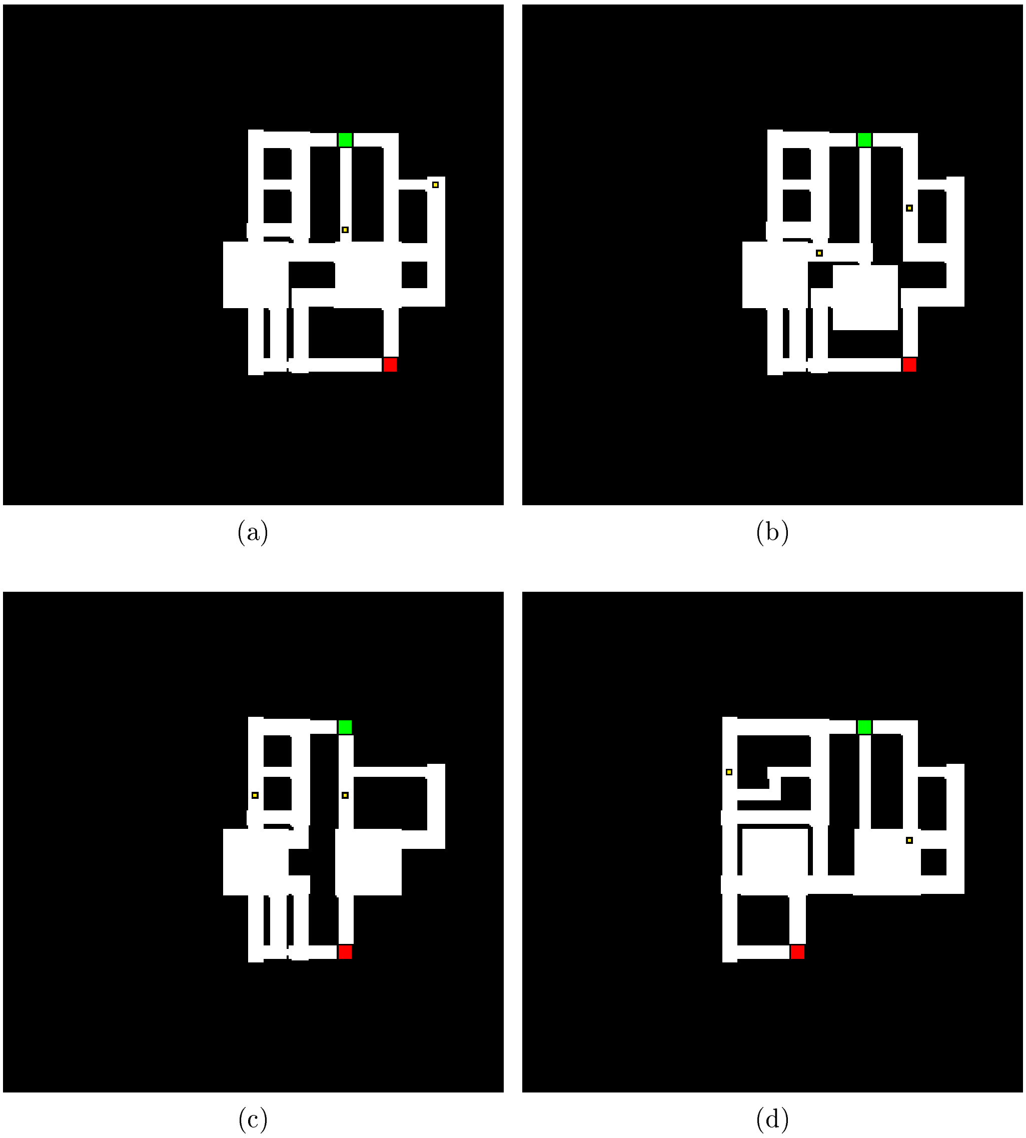


(a)

(b)

(c)

Figure B.3: Lineage of maps for playtest two, second group. (pt2.2)

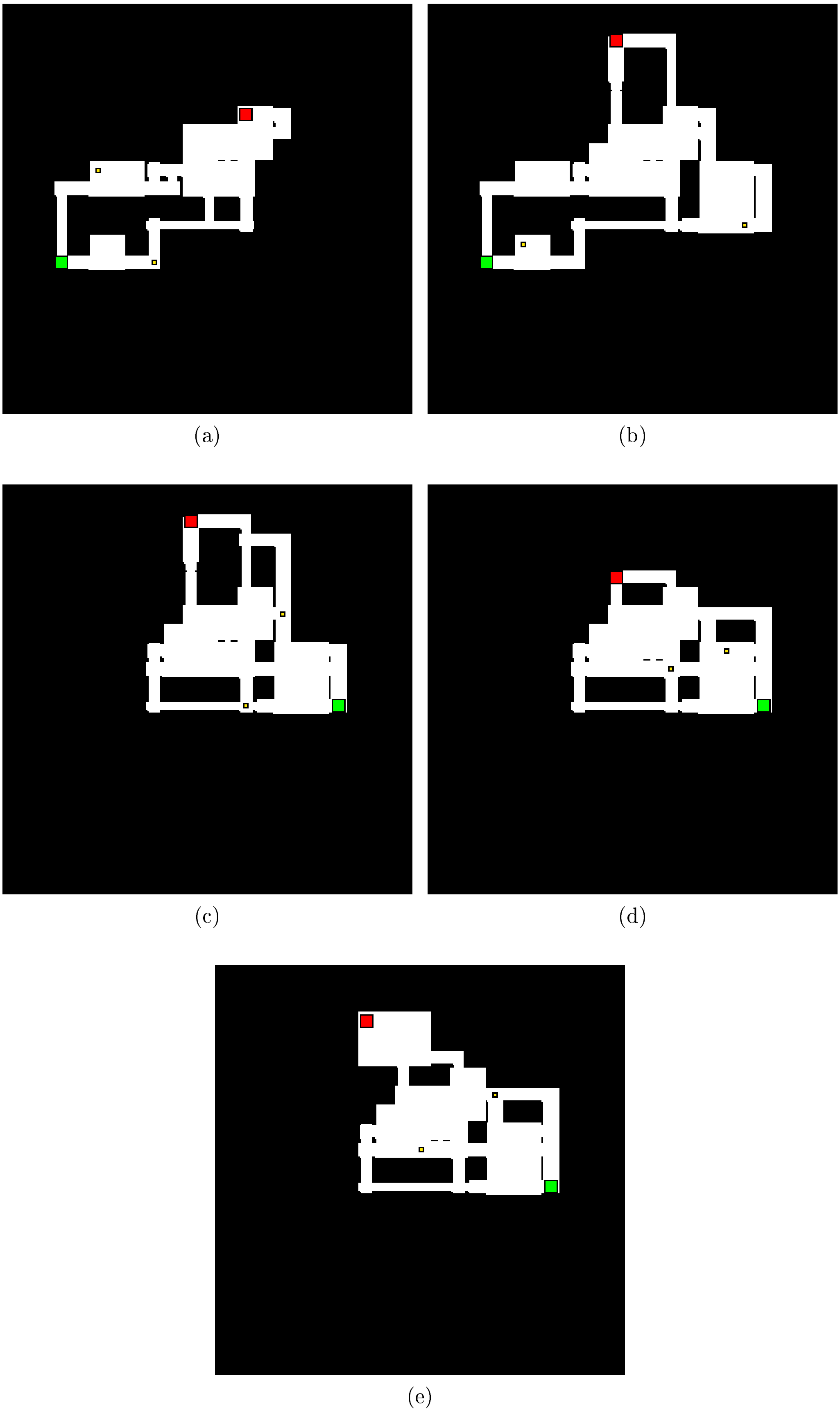

(a)

(b)

(c)

(d)

Figure B.4: Lineage of maps for playtest two, second group. (pt2.2)



(a)

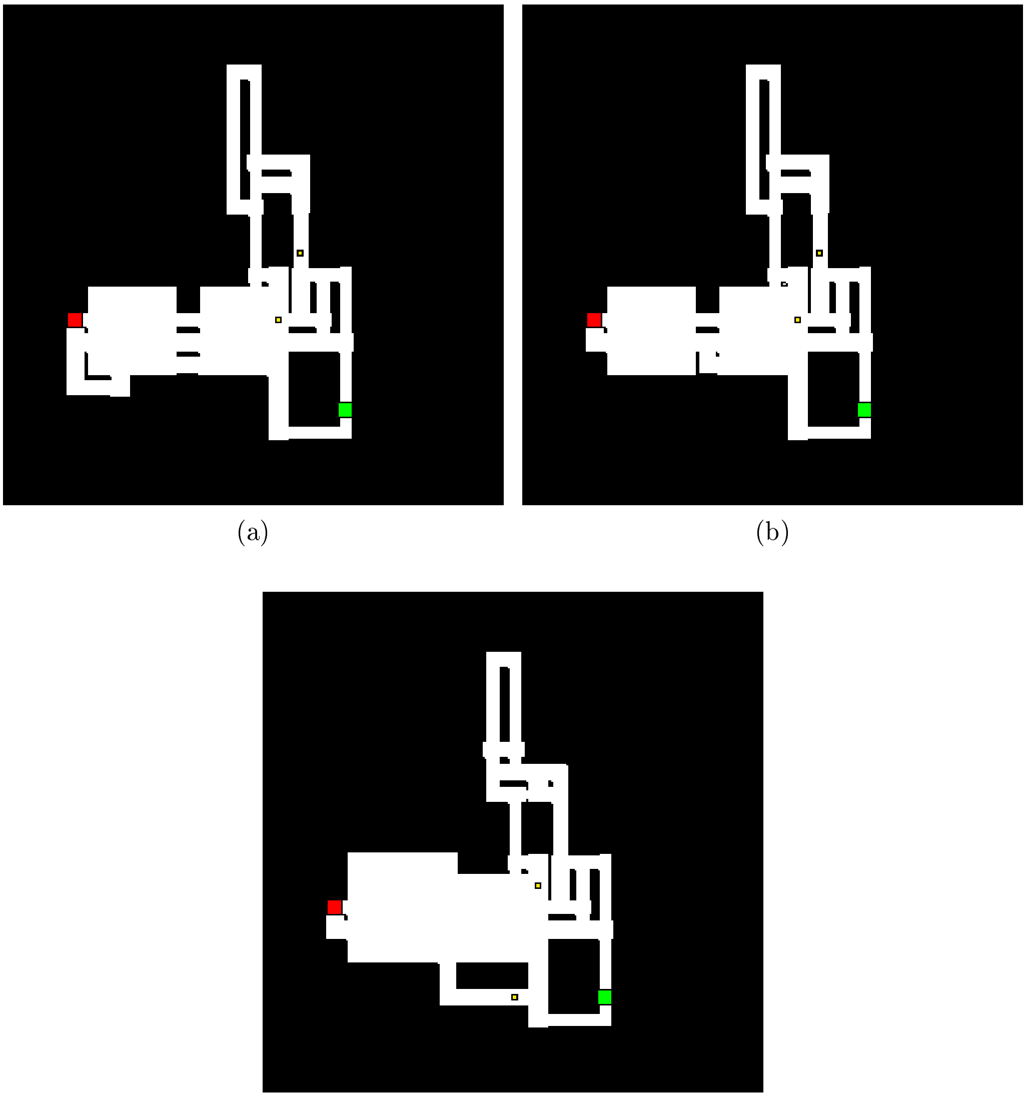
(b)

(c)

(d)

(e)

Figure B.5: Lineage of maps for playtest two, second group. (pt2.2)

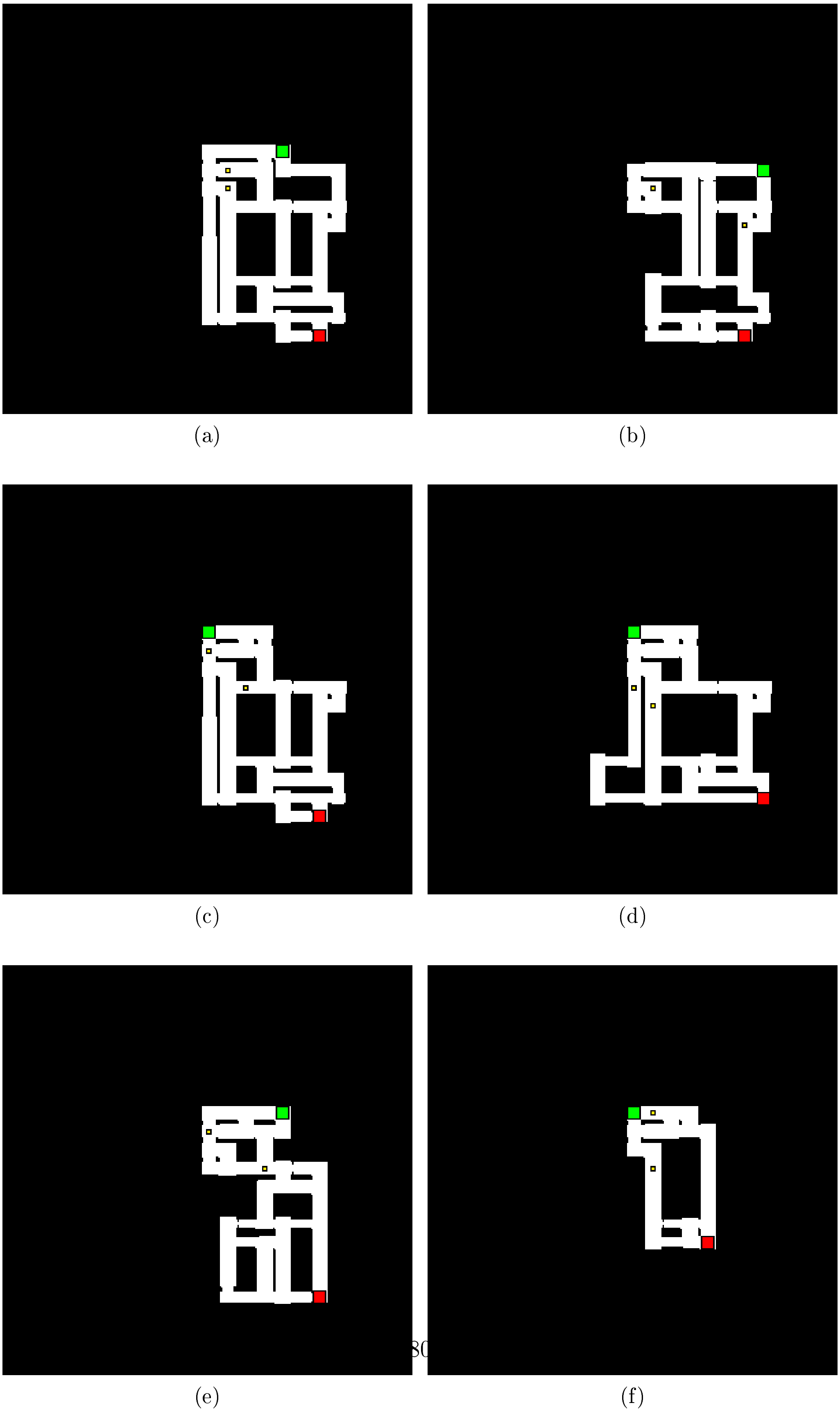


(a)

(b)

(c)

Figure B.6: Lineage of maps for playtest two, third group. (pt2.3)



(a)

(b)

(c)

(d)

(e)

(f)

Figure B.7: Lineage of maps for playtest two, third group. (pt2.3)

C. Questionnaire

The columns on the right are respectively, the average rating, the median rating, the correlation between the player general skill and the rating & lastly the correlation between the player FPS skill and the rating. 0 means no correlation at all, 1 means perfect correlation, -1 means perfect negative correlation.

Question	Average	Median	Standard Deviation	How would you rate your overall gaming skills?	How would you rate your FPS skills?	How would you rate your skills in this game?	Did you enjoy the game?	KD Ratio
How many players were in your playtest?	6,54	7	0,50	-0,28	-0,50	-0,37	-0,53	-0,59
How would you rate your overall gaming skills?	4,62	5	1,27		0,46	0,45	0,46	0,25
How would you rate your FPS skills?	3,15	4	1,56	0,46		0,89	0,53	0,81
How would you rate your skills in this game?	3,46	4	1,78	0,45	0,89		0,61	0,88
Do you enjoy FPS games?	3,62	3	1,64	0,33	0,68	0,69	0,43	0,52
Did you enjoy the game?	4,54	5	1,39	0,46	0,53	0,61		0,58
Did the game accurately portray an FPS game?	100%	1	0,00	0	0	0	0	0,00
Was the movement in the game what you would expect of an FPS game?	85%	1	0,36	0,04	0,18	0,47	0,01	0,33
Was the shooting in the game what you would expect of an FPS game?	92%	1	0,27	0,14	0,21	0,40	-0,10	0,22
Did you use the bomb planting/defusing?	2,54	3	1,28	-0,11	0,15	0,23	-0,08	0,06
Did you feel the game was fair?	69%	1	0,46	0,58	0,49	0,45	0,38	0,36
How sure were you of your objective?	3,38	3	1,90	-0,16	0,26	0,17	-0,17	-0,03
How easy was it to find your way through the levels?	4,31	5	1,38	0,37	0,30	0,47	0,55	0,37
Was the game experience what you expected, based on the picture preview?	4,62	5	1,55	0,39	0,57	0,71	0,45	0,54
Did you use the map-view to help find your way?	46%	0	0,50	-0,21	0,01	0,19	0,31	0,24
How varied did you feel the generated levels were?	4,00	4	1,36	0,36	0,36	0,54	0,69	0,42
How would you rate the overall quality of the maps?	4,31	4	0,91	0,43	0,40	0,43	0,60	0,45
Did the quality of the levels improve as the test session progressed?	77%	1	0,42	0,41	0,64	0,55	0,87	0,48
Did the quality of the levels decreased as the test session progressed?	0%	0	0,00	0	0	0	0	0,00
Did you feel voting could impact the bomb positions?	85%	1	0,36	0,71	0,45	0,47	0,78	0,30
Did you feel voting could impact the spawn positions?	69%	1	0,46	0,58	0,39	0,45	0,38	0,25
How much did the map's layout impact your enjoyment of the game?	4,46	5	1,34	0,29	0,08	0,07	0,61	0,05
How varied did you feel the evolved levels were?	3,85	4	1,17	-0,35	0,10	0,11	0,00	0,01
How much impact did you feel the voting system had on the overall quality of maps?	4,85	5	1,29	0,29	0,01	0,10	0,56	0,05
How would you rate the best level that you played?	5,00	5	0,78	0,39	0,13	0,33	0,70	0,10
Did you win most of the rounds played?	2,92	3	1,27	0,36	0,67	0,80	0,76	0,85
Did you feel your votes made a difference?	85%	1	0,36	0,71	0,45	0,47	0,78	0,30
Kills	30,85	28	14,73	0,21	0,78	0,84	0,54	0,96
Deaths	29,69	30	5,38	-0,22	-0,71	-0,72	-0,58	-0,75
KD Ratio	1,13	0,82	0,69	0,25	0,81	0,88	0,58	
Sum of the map complexity voted on positively	885,15	910,00	222,72	0,01	0,22	0,42	0,02	0,48
Number of positive votes	58,08	59,00	14,49	-0,03	0,18	0,38	-0,04	0,44
Average complexity of maps voted on	15,24	15,17	0,40	0,31	0,25	0,21	0,50	0,27
Sum of the map complexity voted on negativly	694,92	684,00	220,23	0,47	0,06	-0,08	0,49	-0,16
Number of negative votes	45,62	47,00	14,24	0,48	0,06	-0,06	0,48	-0,16
Average complexity of maps voted negativly on	15,23	15,36	0,51	-0,03	-0,01	-0,13	0,10	0,00

Figure C.1: Quantitative questionnaire answers from pt3. Includes recorded kill-death ratio and votes.

Question	Average	Median	Standard Deviation	How would you rate your overall gaming skills?	How would you rate your FPS skills?
How many players were in your playtest?	4,69	5	0,46	-0,36	-0,34
How would you rate your overall gaming skills?	4,46	5	1,01		0,59
How would you rate your FPS skills?	3,54	4	1,39	0,59	
Do you enjoy FPS games?	5,08	5	0,83	-0,04	0,63
Did the game accurately portray an FPS game?	0,92	1	0,27	0,13	-0,10
Was the movement in the game what you would expect of an FPS game?	1,00	1	0	0,00	0,00
Was the shooting in the game what you would expect of an FPS game?	1,00	1	0	0,00	0,00
Did you use the bomb planting/defusing?	3,92	4	1,38	0,58	0,26
Did you feel the game was fair?	0,85	1	0,36	-0,44	0,01
How sure were you of your objective?	4,77	5	1,25	0,08	0,47
How easy was it to find your way through the levels?	4,77	5	1,05	-0,34	-0,13
Was the game experience what you expected, based on the picture preview?	4,77	5	1,05	0,54	0,56
Did you use the map-view to help find your way?	0,46	0	0,50	0,65	0,09
How varied did you feel the generated levels were?	3,85	4	0,77	-0,40	-0,43
How would you rate the overall quality of the maps?	4,08	4	1,07	-0,25	-0,29
Did the quality of the levels improve as the test session progressed?	0,62	1	0,49	-0,27	-0,26
Did you feel voting could impact the bomb positions?	0,77	1	0,42	0,43	0,21
Did you feel voting could impact the spawn positions?	0,54	1	0,50	-0,04	0,14
How much did the map's layout impact your enjoyment of the game?	4,00	4	1,24	0,55	0,67
How varied did you feel the evolved levels were?	3,46	3	1,08	-0,05	0,19
How much impact did you feel the voting system had on the overall quality of maps?	4,23	4	1,05	-0,10	-0,35
How would you rate the best level that you played?	4,77	5	0,97	-0,13	-0,36
Did you win most of the rounds played?	3,31	3	0,99	0,55	0,44
Did you enjoy the game?	5,15	5	0,66	0,01	-0,17

Figure C.2: Quantitative questionnaire answers from pt2.

D. Second Playtest Questionnaire

Questionnaire follows on the next page.

How many players were in your playtest?	4	4	4	4	5	5	5	5	5	5	5	5	5
How would you rate your overall gaming skills?	5	5	6	4	5	6	4	4	3	3	3	5	5
How would you rate your FPS skills?	3	4	6	4	4	6	4	2	3	3	2	1	4
Do you enjoy FPS games?	4	4	6	6	5	6	6	4	6	5	5	4	5
Did the game accurately portray an FPS game?	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Was the movement in the game what you would expect of an FPS game?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Was the shooting in the game what you would expect of an FPS game?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Did you use the bomb planting/defusing?	4	3	4	4	6	6	5	2	3	2	3	6	3
Did you feel the game was fair?	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes
If you have answered 'No' to any of the above questions, please explain?	Fair? Yes - depending on the levels though					Headshots very too effective and hence not fair			Iȷd like to have more feedback on grenades, if i die from them especially. because i never understand if someone shot me or i exploded.			The ways leading to the bomb places were sometimes very easy to predict and therefore it was kinda easy to pick the other team off.	
How sure were you of your objective?	6	4	5	5	4	6	6	2	5	4	6	3	6
Did anything else hinder you in playing the game?	Some weird map layouts - the picture previews were sometimes not the same on all clients			Small connect/setup server buttons. Error, when pressing "These maps suck" one player did not have the same maps as mine:)		T - I dont wanna change team on T !	Switching between primary weapon and secondary one should be faster. Having more unique decorations in the levels could have helped me to find my way around better.		Did not know i had two sets of weapon choices. randomly found out :D still grenade kills feedback more clear. Also headshot kills.			No	
How easy was it to find your way through the levels?	5	4	3	6	4	6	4	3	5	6	6	5	5
When playing a level, was the experience what you expected, based on the picture preview?	6	4	5	6	5	6	5	4	5	3	3	4	6
Did you use the Map-view (M) to help find your way?	Yes	No	Yes	No	No	Yes	No	Yes	No	No	No	Yes	Yes
What did you think of the bomb positions?	No. But the "fairness" depends on the voting so... i guess they are fair when the teams switch positions right..	Yes	Itȷs good that theyre placed closely to the defending teams spawn, but they should not be placed either too close to each other or in such a way that you have to go through one to get to the other. Also, bombsites in the middle of long hallways are pretty bad for gameplay, as youre sort of fucked if you want to defuse it.	avg. ok, a few maps had them very close to each other. But having multiple maps to chose from resolved the issue quickly:)	Pretty generic, often. It would be more fun if they were more variably placed relative to choke points, open spaces, and covers.	Fairly	Pretty much. Sometimes it was more difficult to get to one than the other. So you would always go for the same position.	They were often placed in close range to defending team, and thus providing a fair spot, as well as a more challenging spot which hindered camping at just one spot.	Sometimes they were giving more advantage to either the defend team or the attack team. Iȷd like to have them in a position where is both fair for each team to reach them, defend them and find nice solutions to attack.	Yes	I actually never played the bomb placing team, if i have played it then i actually totally missed that part. I concentrated on shooting	They were too close to the defenders most often. Since my team had more players than the other, and we were quite often defending, the positions became unfair (caused by the amount of players).	Yes I think so because in most cases they have been placed closer to the spawnpoint of the defending team. In some cases they were in the middle of both spawnpoints which was harder for the defending team i think. Well, at least if the other team tried to place the bomb and not just killing people.
What did you think of the spawn points?	Same as above	I guess so.	good. placed closer to the bombsites for the defending team than the attacking, so you can get there in time.	Same as bomb points, But generally very good:)	Usually pretty good, although sometimes you could reach the other team immediately or too fast.	Not always - We had one instance where you could see directly from spawn to spawn and open fire from the first second of gameplay	Yes. Expect for one instances where we spawned at the end of a long corridor and we were able to shoot eachother as soon as we started the round.	They were alright	there was one map where the spawn points were in front of each other, so the fastest person to spawn would easily kill the opponent. I liked having the spawn point close to the defend base, and a bit further when attacking.	Yup	I cant say anything particular. This is an important point of level desgin, very much related to the map design questions. The spawing point were okay, as there was never really a different outcome in the map, through the map design.	Yup :)	Yep very fair.
How varied did you feel the generated levels were?	3	4	3	4	4	3	5	4	4	5	3	5	3
How would you rate the overall quality of the maps?	5	4	2	4	2	5	5	4	5	5	3	5	4

What seemed to constitue a good map	not too big of a level, so people didnt run all over not finding each other. Long hallways seemed... bad	Quick encounters within short time frames.	Independently placed bombsites (aka bombsites arent in view of eachother) with good covering angles and a limited amount of chokepoints entering the bombsite.	A map without more than one or two long corridors/hallways, with spawn and bomb points not to closely set. The smaller maps where generally better	Good mix of open and closed spaces, distance between bomb points, multiple paths so gameplay doesn't become monotone.	A good mixture of open areas (with some stuff to hide behind), choke points and long corridors. Also multiple ways to get around the map is good for keeping the match from becoming a stalemate	No long corridors. Bomb places placed in opposite sides of the map. Lots of small chunks of block in open areas that would allow you to take cover easily.	Not too long pathways to arrive at a bomb site. Only at one certain point, did arriving to bombsite B take way too long path which was difficult to arrive at. Open spots in the middle of the map were fun because it created a battlefield rather than just corridors.	Different paths that lead to big areas, so you can plan from where to attack the enemy and surprise them. Big areas with more covers, to strategically hide and take cover. Bottle necks where you "force" the player to meat the opposite team so that they will play against each other rather than going around trying to find people to shoot at.	I don't personally like the big open spaces. The long corridors and small alleyways are the best	I think to a good map design there are missing some vantage points, especially "lifted" or higher areas, infiltration from above, or from beneath were missing. Everything felt like a very long corridor. (Doom like)	There `were` long corridors, but they could be avoided. It was more fun the more "enemy contact" we had, therefore the smaller maps and the maps with more bottle necks were more fun/better.	Basically speaking uncomplex maps. For example the last map we played was a lot of fun and very tense because it was very small and basically one corridor that went all around the map. That helped to predict where the enemy players might be. In the more complex levels it was basically running around without plan until you found an enemy
Any other notes on the quality of generated maps?	Holes in the outer walls of some maps, making an escape from the level possible		Generally any bombsite should have an almost-fixed amount of chokepoints so that the defending team can spread out and effectively cover bombsites with fewer players than it takes to attack it. There needs to be crates and things to hide behind so as to not be completely vulnerable when waiting for the opposing team. Generally, I feel its best when a bombsite has a chokepoint leading towards the defending teams spawn as well as 1-3(rarely 3, but it happens) pointing towards either the attacking spawn or somewhere mid-map where both teams have a chance of coming from. This leads to interesting attack-strategies as the defending team can feel relatively secure from the defending-spawn-chokepoint but not so much that they will never be attacked from there.	n/a		Crates were sometimes spawn half within the walls			Depends on how many people are playing, maybe if there aren't many, the maps should be smaller and be larger if there are more. I had to choose between some maps at the beginning so i was always going for the smaller ones because we were only 5 ppl playing.	Quite good.		The crate were sometimes placed in a cramped way. I liked the niches where one could hide. I really liked the bottle necks. There were often two or three different ways, which was really nice, made it less predictable.	Nah i think it was ok.
What were the deciding factors, when voting for a new level?	to me: Simplicity. Generally: split spawnpoints and variety i think	Tried to choose many different variations of levels.	Because they fixed problems with the prior level.. by moving a bombsite to a more interesting location or by removing some pointless paths.	It looked better then the previous one. It took playing a few different maps to figure out what constituted a "good" map. But all the maps functioned without problems.	Looking for very dense maps, maps with distance between bomb points, maps without separations between areas with only one path between them.	They looked "fun" - meaning that they had the features I would think makes a map good	Whether it felt like it was to complicated to find my way around. + whether I was able to actually get to place or defuse the bomb in time before everyone else died.	They looked different from previous level. Deciding was just based on the thumbnail, such as interesting shapes of the level or fun spawn points.	Size of the map, placement of spawn points compared to bomb. comeplexity of the map. (narrow corridors vs openspaces and how to reach the objectives)	I went for the levels without big open spaces, because I'm such a noob with fps games that it's too hard for me to hide.		Levels down the road seemed to enhance existing features (like long corridors). Most levels had a really unfair bomb site placement, so I picked the least unfair ones.	I tried to vote for simpler maps. Because they were more fun
Did the quality of the levels improve as the test session progressed?	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Did you feel voting could impact the bomb positions?	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Did you feel voting could impact the spawn positions?	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No
How much did the map's layout impact your enjoyment of the game?	6	3	5	4	5	6	4	2	4	4	2	3	4
How varied did you feel the evolved levels were?	3	3	3	4	5	4	5	5	4	2	3	2	2
How much impact did you feel the voting system had on the overall quality of maps?	6	4	3	4	6	3	5	4	4	5	3	5	3

Any other notes on the voting and evolution of maps?	Didnt seem to work all the time.. bugged		Side by side comparisons might be nice? An image of the old map or something so that its apparant what was changed.	n/a		The system made sense, but the way the interface was made did cause some confusion at first		The thumbnails were 2 dimensional and not very representative. I feel like adding dimensions could help decide which map to vote for. At the end of games, the voting system was unclear in the sense that bad represented worse, and likewise good represented better rather than good. I also felt a comparison to previous level was hard. Maybe random generating a name associated with the thumb could ease the association.			I actually didnt really cared about the voting.	The generated maps were too similar to the old map. Maybe add more randomization (or corridor recognization to eliminate those maps beforehand). The really small doorways were nice sources of surprise. The decals(barrels, boxes) were well placed on open spaces.	
If you have played Counter-Strike, how did the generated levels compare?	A lot i guess	Haven't played it.	Poorly. The only similiar thing was the proximity from defenders spawn to the bombsites.	smaller, but just as fun	Not quite on par with good maps created by experienced human level designers.	These maps here were simpler, but otherwise fairly similar	The main benefit I can see in having the levels generated is that everyone has an even chance at winning, whereas in counter+strike, the better you know the map you're playing, the higher your chances of staying alive are.	Flat, but structurewise quite good			Fast paced narrow corridors with few open spaces.	Only played once, didnøt like it. Therefore your levels are waaaaay better ;)	The counter-strike levels are better designed as far as i remember. On the generated maps chokepoints were present like in the counter-strike maps but not used very often because the whole map is basically a corridor so you could use everything as a chokepoint if you wanted. Also one thing that I was missing were height differences. You cannot take advantage of a high position especially when you are defending that would be nice.
If you have played Dall of Duty, how did the generated levels compare?	Not very much, but then again i dont play CoD	Haven't played it.	more closed off. I feel CoD has a lot of open areas compared to Counter-Strike (and this).	Smaller, and better. CoD levels are much bigger and the greater variation in weapons does make the bigger maps useful in CoD.	Same.	These maps here had less objects to hide behind and felt more arcade-like					i dont see a similarity here. It felt from the first spawn like CounterStrike.	DALL of duty haha	Same as above basically
If you have played other similar multiplayer FPS games, how did the generated levels compare?	aside from visuals i think heightmaps should be altered to really get good vantage points	Much smaller compared to other FPS games I play. No way to change altitude or go a floor down/up.		Good	Same, again.	The maps here are in general smaller I would say			I thik the experience that you gain from other fps game have more thought into it. These maps were doing good, but sometimes i d like for a really thought experience. Level hights maybe, not only horizontal plane.	I've played left for dead, which is ..kinda the same, but the levels are much more varied. Jeg skriver lige på dansk, kællinger! - Der er ligesom meget mere variation i lfd - mest fordi alle elementer ligesom er med. Der er smalle gader, brede gader, store veje, buske og planter. Mulighederne for at gemme sig, og sådan snige sig rundt og udnytte miljøet omkring én var ret begrænsede.	I already made the comparison, but it had soem doom flair. Narrow, corridors with suddenly appearing enemies. It almost could be a feature to add random enemy champions against both teams.	Not as much variance, which is to be expected.. There is no reason to jump in the generated levels. The bomb sites were always sheltered, in other fps they are sometimes in the open	Same
How would you rate the best level that you played?	6	4	3	6	4	5	6	5	5	4	4	6	4
Did you win most of the rounds played?	4	2	5	4	3	5	2	2	3	3	3	4	3
Did you enjoy the game?	5	4	4	6	5	6	5	5	5	5	5	6	6
Tell us why you did or didn't enjoy it	Mostly the "LAN-ish" environment. The game in itself is pretty standard but works	The explosives had a too wide radius. Could shoot a second or so after I stopped sprinting. No use of using aiming down targets in my opinion.	Pretty fast-paced and it sort of felt like you hit what you aimed at... Roughly.. Iron sights suck though ;)	Nice CS type game:), Great for LAN and the like.	Much satisfying to shoot people. And blow them up. And sneak up on them. Lols.	I was winning! Winning is fun! - But we also had fun as a group playing & interacting, which made it more fun.	The rounds didn't last too long and the controls were very familiar.	Cause I'm high on cocaine.	I enjoy the players :D some situation it created. liek when i got blocked by people throwing bomb on each side i was trying to escape.	I get pretty sick from playing fps games, no idea why. But that has nothing to do with the quality of the game.	Playing with friends.	Fun! And PCG yeah	Well shooting people is always fun if you are in one room with people you know. But in short becasue its a competitive multiplayer game.

Do you have any final thoughts?			Iron sight aiming slows down play and generally noobifies the whole experience.	n/a	Needs more cowbell.	Can I play again?		Nicely done	Was fun, but i canȷt think anymore lol ¡3	The game was better than I had expected. Actually way better. Men ϒdet bliver uklart hvad spillet egentlig handler om, og hvornår spillet stopper. Er det f.eks fordi man har slået alle modstandere ihjel, eller er det fordi man har droppet den der bombe? For der sker ikke rigtig noget med bomben når den bliver smidt, og så virker det mere som om man egentlig bare skal skyde sine modstandere ned, og det er egentlig måske også det sjoveste.	Go on. The concept is great if you can bring it to another level, more randomness to genereate, different assets, AI*s maybe, heights, tunnels, secret rooms etc.	I know you're done, but if you would use multiple levels as seed it might increase level quality. If players could vote on more than 6 levels before playing the first level, the generator might learn more. Ex: All players vote one of six maps, then again and again, and then play the third generated map	Even though it does not change the maps a lot at the moment i think it works very well. I also liked that it recocnizes the votings and adjusts the maps to the previous map. One problem with that was that after the 5th vote or something basically all the maps looked the same.

E. Third Playtest Questionnaire

Questionnaire follows on the next page.

How many players were in your playtest?	6	6	7	7	6	6	7	6	7	7	7	7	6
How would you rate your overall gaming skills? (1-6	5	6	6	5	4	6	1	5	5	4	5	4	4
How would you rate your FPS skills? (1-6)	4	5	1	4	4	6	1	3	1	4	4	2	2
Do you enjoy FPS games? (1-6)	3	5	4	3	5	5	2	3	1	6	6	1	3
How would you rate your skills in this game? (1-6)	5	5	2	4	6	6	1	2	2	4	5	2	1
Did the game accurately portray an FPS game?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Was the movement in the game what you would expect of an FPS game?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Was the shooting in the game what you would expect of an FPS game?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Did you use the bomb planting/defusing? (1-6)	1	4	3	5	3	2	4	1	2	3	3	1	1
Did you feel the game was fair?	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	No	Yes
If you have answered 'No' to any of the above questions, please explain?							Cause i suck	It felt at bit slow at times. It was straining to hold down the shift button all the time to run. I never figured out how to plant bombs.	The bomb sites weren't placed at equally strategical positions. They would very often be placed at a high disadvantage for one team, skewing the game experience in favor of the other team	Sniping seemed really op compared to other weaponz, never tried the hand gun. But still fun :D	We didn't use to plant/defuse the bomb that much because most of the time we would kill eachother really fast.	I didn't feel the game was fair due the fact that one team sucked horrible and the other one owned the entire way through.	I feel that the motion of movement could have been more concise, I didn't feel able to move freely. sideways and backwards while pressing more keys. I didn't get the accuracy of the main weapon. It mostly felt very random if I actually hit my target. If too far or close, it felt like I hit nothing but thin air. The sniper felt a lot more fair.
How sure were you of your objective? (1-6)	3	6	2	6	2	2	6	4	1	3	6	2	1

Did anything else hinder you in playing the game?	Network issues, not for me though	Nope, not really	Rules were not explained too much. Understood placing bombs and defusing bombs much later. Sometimes, objective did not show. It should be on the screen all the time.		UI: Hard to know if you were attacking team or defending. Network: We had some problems with the voting system, but none ingame.	Network issues were abundant. After the first map the game had problem with the voting. There were also a couple of times where the physics bugged out and threw people off the map.		I was straining my hand a lot towards the end, because I wanted to run most of the time and had to hold down the shift button. Bug with being thrown off the arena :P I never noticed wether I had the bomb or not.	network issues, ITU network sucks	I knew that i had to defuse or plant abomb, but I was more into the killz! The tension of surviving! But if I felt safe enough I would consider the planting the bomb	The sniper rifle was too powerful I believe.	stuck on bokses and jumping behind walls O.o	Never got a good feel of the range and impact of granades. The UI was simple, which I don't mind at all. Although, it rarely felt clear how much life I had and if I had the bomb. All I noticed, was granades, defuse kit and if I took the time, who killed who (not teamdefined tho, so it didnt give me a sense of team progression).
How easy was it to find your way through the levels? (1-6)	4	6	5	4	5	5	4	4	6	5	5	1	2
When playing a level, was the experience what you expected, based on the picture preview? (1-6)	6	5	5	6	6	6	4	5	4	4	6	1	2
Did you use the Map-view (M) to help find your way?	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No	No	No	No
What did you think of the bomb positions?	We got to chose the placement we wanted so... they were good. They totally worked. Seemed like people agreed on stuff.	I liked it better when the bomb position are both further apart and further from the starting positions	We voted for the more interesting ones. Placement was good.	Best when far away, but that was not always an option during voting.	Sometimes there wasn't much difference between 3-4 of the 6 options.	The bomb positions had a tendency to be on long halls, which made them very easy to defend. There was rarely a good opportunity to plant the bomb without getting shot by the opponents	ok, sometimes they spawned too close to teams	I didn't really notice them to be honest. I focused on the killing the other team instead.	weren't placed well (see answer before). An incredibly good thing: the separate voting on bomb sites and level structure	I liked when they were positioned in places where it was hard to guard them. It made the game more challenging. Having 3 openings.	There was a decent variety in chosing the bomb positions. This was a good thing because I could choose whichever seemed more fare for both teams.	I rarely took notice of the bomb positions as I had problems surviving :P	The bomb mode itself seemed unnecessary. I'd rather a more polished version, with no bombs. Simply team vs. team.
What did you think of the spawn points?	seemed to be placed fair	They do generate bottlenecks which are easily deductable based on the speed of the players. That's how you would know where to go and what weapon to use depending on the amount of players left. In general they were OK. The more space and corners and pathways between them, the better in my honest opinion.	They were fair.	Good.	Sometimes there wasn't much difference between 3-4 of the 6 options.	The spawn points seemed to be placed in good spots. They allowed quick movement around the map.	ok	Great that they were placed in opposite corners of the map, and that there were several ways to get to each other. Sometimes I would try to get behind the other team	OK, were nearly always placed well, at opposing level parts. Different placement methods could have resulted in more interesting gameplay	Spawn point were nice when they were not close to each other.	I believe they were place in approriate positions on the map because most of the times they were in opposite sides and required you to walk for a while before encountering an enemy.	I couldn't really figure out where I was on the level compared to what I was voting for.	a bit clumsy, since people got flung up onto the walls. Besides from that, very plain and simple (which was nice).
How varied did you feel the generated levels were? (1-6)	6	4	4	4	4	5	3	2	5	4	6	1	4
How would you rate the overall quality of the maps? (1-6)	6	4	4	4	5	5	3	5	4	3	5	3	5

What seemed to constitute a good map?	A map where some parts were close combat, some open hallways (for snipers) some crates for variation and multiple possible paths to the opposite spawn	A balanced map with multiple paths to victory.	Interesting choices for the player, some mazes, some bigger area. Well placed bomb points.	Having a certain number of paths between key points (spawns and bomb spots), probably 2-4. Having few or no areas where you have to walk a lot without anything happening (ie long corridors).	Flanking options, combinations of long stretches for sniping and smaller distances.	Easy access to the bomb spots. Few hallways and a couple of open areas that allowed fun combat.	space before meeting people	Being able to go around in a circle on the entire map - but still not having the map be too simple, so you could take alternate routes.	The bomb sites, the spawn points, but most importantly the spawn point connections. Bottlenecks and short ways between the spawn points were fun, as well as "labyrinths" between the spawn points	A good balance between open areas and narrowed places. Easy to get to the target whe in open areas but feel exposed, longer way in narrow corridors but safer	Spawn points in opposite sides of the map, bomb positions placed that are easily reachable by both team, and a balance between open spaces, corridors and cover elements (crates etc)	I think it's hard since all the walls looked a like, and the only thing that varied it for me was the boxes and the occasional shrubbery.	smaller maps with less endless labyrinths.
What seemed to constitute a bad map?	maps with too many paths and too many unnecessary paths. And the opposite of all the above	A map that has very predictable bottlenecks, or a straight line between both spawn points.	Too small or too big.	The opposite (sorry, shouldve put some of the above here)	Not enough variety.	The maps with a lot of hallways usually led to the game devolving into a snipe shoot-out. The bomb was rarely planted because of this.	tight rooms	If it was too simple, so everyone had to run down the same roads every time. Harder to hide from the other team. Also if there was only one rotue to the other teams base.	Where one team had a distinct advantage/disadvantage over the other team. Also, the crates sometimes blocked the way and didn't serve as cover > a blessing and a bane	Sometimes the props around the level were blocking corridors, which was a bit frustrating cause i didn't really know if i could pass through them or not.	Spawn positions crossing eachothers line of fire, bomb positions that are too close to the spawn points and a lack of cover areas.	When the spawning and bomb positions where too close to the "protecting" team?	big complicated levels, with lots of deadspace. (places placed far, where no one would go).
What were the deciding factors, when voting for a new level?	Multiple different paths, variation in paths	Distance between spawn points. Having as many corners as possible between them.	How big the map was, how many mazes it had.	No redundant areas (not being paths between spawns or open rooms), not having too close spawns, not having too many non-branching corridors.	The variety of the presented options, that means if the map looked like it had different distances and options for flanking and such.	Finding a map with a couple of pressure points. This allowed fun gameplay.	i picked at random	When the overall map was constructed as a circle - but still complicated enough that you could take several routes through the map.	The size of the map, the connections between spawnpoints. Expectations on how much fun/how many confrontations a level would elicit	Goal had to be in between the 2 spawn points. With nice balance, open areas and sneaky corridors	Mostly the layout of the map. Meaning a balance between corridors, open spaces and cover points. Besides that also placement of bomb positions that were not too close, either to far away from the spawn points.	is it compact? No long corridors.	My preference was to go for simple smaller maps, with a bit of variation to keep it fresh, and opportunities open.
Did the quality of the levels improve as the test session progressed?	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes
How so?	The last map was killer, the first one... had an island..	They felt more balanced over the play session.	I'm not sure if they did. Playing seemed similar.	Not quite sure if they did, but there were fewer retarded options among the maps to vote between.	I chose yes, but really I thought that it was too hard to tell if the maps really evolved. We had 3 map evolvements, but I couldn't really judge if the evolvement was better or worse from the period of time playing them.	It took the fun levels we picked and mutated them. Usually this mean changing how the hallways were generated which helped a lot with the problematic long hallways. The addition of a crate here and there also helped.	I didnt notice a significant difference	They seemed bigger. And I also liked that there was a huge open space located somewhere on the map. So you couldn't just hide in the narrow roads.	The first levels were too big, had too many corridors. Maybe I just became better.	More corridors that had multiple ways to sneak on people, smaller area with smaller amount of players	My votes mostly corresponded to the ones of the majority and it seemed like maps were changing in a better way.	It wasn't really something I paid attention too.	1. Cause people had the power to vote, so naturally, the majority were satisfied. 2. Cause the idea of an evolving map bending to the players votes over time was unique and fun.
Did the quality of the levels decrease as the test session progressed?	No	No	No	No	No	No	No	No	No	No	No	No	No
If yes, how so?		Wrote above.					answered before					Again, it wasn't something I paid attention too. But I did appreciate the shrubberies.	
Did you feel voting could impact the bomb positions?	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes
Did you feel voting could impact the spawn positions?	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No
Did you feel your votes made a difference?	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes

How much did the map's layout impact your enjoyment of the game? (1-6)	6	4	5	4	5	4	3	6	5	5	4	1	6
How varied did you feel the evolved levels were? (1-6)	6	2	3	4	4	3	4	3	2	5	5	4	5
How much impact did you feel the voting system had on the overall quality of maps? (1-6)	6	4	6	5	5	5	4	5	6	5	5	1	6
Any other notes on the voting and evolution of maps?		Later generations seemed more as an iteration of the previous map if it was a successful map.			A lot of the voting options seemed to be the same. Sometimes 3-4 of the 6 options were close to identical.	It would have been nice if there was a visual difference in the debris on the map. It wasn't obvious whether the debris you placed was a big crate that would block line of sight, or merely a small piece of trash. It would also help if they were easier to see on the generated maps. Small brown dots were hard to distinguish.		Nope.				The overall time used on each map was not enough to gain enough familiarity with the level.	It's a great concept, and I see much potential for it. Another thing that makes it great is, that no one knows the map better than others.
If you have played Counter-Strike, how did the generated levels compare?	the evolved one in the end kinda worked similar to a map from CS	They seemed alright. Some counter-strike levels have more than 2 possible pathways. That's how many the game seemed to have most of the time.		(I havent played enough CS to have a valuable opinion here... and I suck at it)	These maps were simpler than in CS, and also felt smaller. The fact that these maps were flat, and didn't have any difference in height, made it a little bit repetitive to play, in contrast to CS.	Much less polished. Counter-strike maps are usually made with several entryways into a bomb plant spot, in order to allow the "terrorist" a fair chance in reaching these. The generated maps here usually had choke points, that would stop the terrorists from reaching the bomb spot safely.		I haven''t played counterstrike.				Better graphic. Actual buildings that can be used. Flashy weopons and objects that can be collected or upgraded.	Couldn't compare.
If you have played Call of Duty, how did the generated levels compare?	Havent played it enough to tell.	Levels are more compact but I've only played the earlier iterations of CoD in multiplayer		(same)	These maps were simpler than in CS, and also felt smaller. The fact that these maps were flat, and didn't have any difference in height, made it a little bit repetitive to play, in contrast to CS.	The generated maps compared much more to call of duty maps, since they also have a lot of choke points. The main difference would be that there is a lot more verticality in the call of duty maps, which makes for more interesting gameplay.		I haven't played COD				I've never played Call of Duty	Have not played it.

If you have played other similar multiplayer FPS games, how did the generated levels compare?	I could see many different games fit into these sort of levels. And of course as long as I am getting to vote for the level, that level aint gonna suck ;)	Maps are compact, similar to Unreal Tournament or Quake 3 maybe but less fast. Some levels were a bit more high paced and that was more similar to Unreal. Some also required more movement and agility, also similar to Unreal.		(same)				They were alright but still very basic. It would be fun with a "layered" map (e.g. a house with several floors), so you could perhaps get in a sniping position or easily escape the other team by jumping down to the level below.					Really can't compare them.
How would you rate the best level that you played? (1-6)	6	6	5	5	5	4	4	4	6	5	6	4	5
Did you win most of the rounds played? (1-5)	4	3	1	3	5	5	1	2	4	3	3	2	2
Did you enjoy the game? (1-6)	6	6	3	4	6	5	2	4	6	5	5	2	5
Tell us why you did or didn't enjoy it	Lots of yelling and good old fashioned fun with a gun. Whats not to like? Except no shotgun :((	It was enjoyable both when I was winning in the 15 player test and in the 6 man test. It felt like a challenge in the 6 player version, while in the 15 man test I was mostly steamrolling, though that felt like gloating enjoyable with the occasional challenge from another player.	I suck at FPS games, and didn't clearly know the rules. I was getting shot all the time. But if I was a better player, I would have enjoyed it more.	Not so much into twitch FPS. My reaction time sucks.	It was easy to get into, it was fun to play with people around you, and the fact that none of us had ever tried it before helped it feel "new".	I can see the game not being fun if played alone and online. The "lan" atmosphere you get from playing it sitting in a room with people, helped it a lot. The ability to influence the map layout was fun and made it interesting to see the next step.	Cause I suck terribly at fps games	Straining my hand did effect the experience a bit. But overall it was fun. Especially because I was sitting next to the people I was playing with, so you could kid around.	The fps mechanics were fun, plus the levels became better and better. I really loved the last level.	Bombs, sniping flying when shot :D	It as great fun because we always needed to find your way around instead of being able to learn the maps and walk around by know the places by heart.	I'm really bad at FPS. Sorry X)	Cause you guys made it!
Do you have any final thoughts?	Great job guys, good luck!	Make the space between bombs bigger, like opposite sides of the map or something or at least divided by a decent amount of walls. And sniping is very deadly.	Good work :)	- Better generated levels than last playtest! - New voting system rocks (but is a bit confusing)!- - A little buggy... - Sniper is cool but seems overpowered.		A fun game and interesting concept. The shooter part was a bit easy to exploit. Way too easy to use the sniper unscoped and just kill people from the other side of the map.	It was an ok experiment, it would have been better if the enviroment textures and apearance changed every time. maybe adding leevels where only snipers or other weapon restrictions existed.	Not anything I haven't already mentioned.	Considering that it's not a polished game, it was much more fun than expected. The level generation was a really nice touch. I had the feeling that I could become a mini-game designer myself. By creating the maps I also felt a bigger urge than usual to vary my approach to winning to explore what was fun about the level and what wasn't.	Having to many thumbs up was confusing :D		I think you guys are doing a great job with the overall leveling. I'm just a terrible tester! Sorry.	Nope :)