Juha Kuoppala

LEVEL FLOW AND PLAYER GUIDANCE IN A 3D MULTIPLAYER LEVEL

Bachelor’s thesis
Game Design

2019

South-Eastern Finland
University of Applied Sciences

Author (authors)	Degree	Time
Juha Kuoppala	Bachelor of Culture and Arts	April 2019

Thesis title

Level flow and player guidance in a 3D multiplayer level

39 pages

Commissioned by

South-Eastern Finland University of Applied Sciences

Supervisor

Marko Siitonen

Abstract

The aim of this study was to analyse the effectiveness of both direct and indirect player guidance methods in achieving an optimal level flow in a 3D multiplayer level and how these methods should be used in practice. The goal with the thesis level design project was to design a level with minimal problems in the flow. The thesis level design project was to serve as an example of how well the methods perform in practice.

The methods of player guidance were first studied by content analysis of literature and articles of the level design field. Prototyping was then used to create the first versions of the level design project. The prototype version of the level was then iterated according to data gathered by qualitative research methods in the form of playtesting and gathering player feedback. The effectiveness of the player guidance methods was evaluated during the entire level design process.

The study showed that the different methods of player guidance analysed, evaluated and put into practice performed well and allowed for a smooth and unhindered flow in the level. The goal of the thesis project was achieved and proven in playtests in the later versions of the level. The level also gained popularity with both casual and competitive players of the game it was designed for.

Keywords

level design, level flow, 3D, multiplayer

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

The purpose of this thesis is to study and prove in practice different principles and methods of player guidance inside a 3D multiplayer level. This thesis will include both theoretical research from literature of the field as well as a practical project. The thesis will also analyse other level design works from both single- and multiplayer games to use as examples. The practical part of the thesis will be used to visualize the usage of all the methods in practice as well as prove their usefulness in achieving the proper level flow through gameplay testing.

The practical projects used as examples in this thesis are the author’s personal level design projects for Valve Software’s game Team Fortress 2. This game and the level design projects for it are used as examples. Firstly, even though the game is already quite old and the technology behind its game engine, the Source Engine, is far from the most modern, it still uses many of the same principles relevant to multiplayer shooters to this day. Secondly, the game is shipped with all the necessary tools required for designing custom user created levels free of charge. The creation of levels is even encouraged to this day by the developers through the implementation of Steam Workshop, through which the developer Valve Software can feature, license and monetize (generate revenue for the content creators) user created content for the games’ updates. Out of all the projects referenced and used as examples, one will be chosen as the main project to be featured in this thesis.

Structurally this thesis will begin by giving a brief overview of level design and what the level designer’s job is in today’s industry. In the next chapter, gameplay mechanics and level design theory of Team Fortress 2, the game for which the level design was completed for, will be briefly explained. The technical features of the game’s game engine, the Source Engine by Valve Software, as well as the tools used for level creation inside the engine will also be summarized. Next, the design process in the completed project is explained and visualized with in-game screenshots from different stages of the process. After this, the methods used to achieve optimal level flow and the different ways of player guidance are discussed in relation to the level design project. Examples will mainly be drawn from the main project, but other projects from the author as well as other level designs from professionals within the industry will be used. Lastly, the findings will be examined in the conclusion of this thesis.

2 BRIEF OVERVIEW OF LEVEL DESIGN

This chapter will discuss the general overview of what level design has meant before and what it means in the industry today. The origin of video game level design reaches almost 40 years into the past, all the way to the first video games created. When the game industry was still at its infancy the title of a level designer was not born yet as the programmers were usually the ones designing the levels during the development of a game. During the rapid development of the games industry, the amount of work creating and designing levels has increased dramatically and so the job of a level designer was born. (Shahrani 2006, 1.)

In today's game industry, the job of a level designer varies considerably. The importance of a game level depends heavily on the type of a game being developed. A simple mobile game ideally wants to rely on interesting and hooking game mechanics rather than a diverse environment to explore, one that can be found on games with a larger scale and budget. Game designers create the rules and mechanics for a game, providing a backbone for the game, whereas it is the level designer's job to implement these mechanics and gameplay ideas into a playable form. (Byrne 2004, 5.)

3 TEAM FORTRESS 2 LEVEL DESIGN

The game for which the level design project portrayed in this thesis was completed for is Team Fortress 2 by Valve Software. The game was released in 2007 as a part of Valve Software's Orange Box (Dobson 2007). Team Fortress 2 is a great way to gain experience and knowledge in the field of level design regardless of the fact it was first published so long ago. The game still holds a steady player-base to this day, is currently free to play and is still supported by the developers. The game also includes vast amounts of highly diversified gameplay mechanics providing level designers with challenges and opportunities to work with. (Valve 2014.)

Team Fortress 2 features 9 different classes with 3 sub-lasses for players to choose from. A well-designed level will need to tend to all the different playstyles these classes offer the players, and the level must be fun for each individual class to play on. The levels in Team Fortress 2 revolve around two teams, RED and BLU, battling each other in set game modes. (Valve 2014.)

Game mode is the most basic gameplay defining element in a TF2 level. There are at least 7 different game modes currently available in TF2 ranging from more classic modes, such as “Capture the Flag” (CTF in short) to more modern implementations like “Player Destruction” and “Payload”. These game modes define the player’s objective in the level and in the match played. The modes always feature a clear objective the players must achieve to best the opposing team. Players fight over and gain ground by pushing the enemy back and control points act as checkpoints of sorts, focusing the fight on them. (Valve 2014.)

Levels for TF2 can be either symmetrical (mirrored or rotational symmetry) or asymmetrical. Generally symmetrical maps are for game modes, where both teams are in both the attacking and defending role, and asymmetrical maps are for game modes where RED is in the defensive role and BLU team is in offense. The game mode selected for the level design used as an example in this thesis is five capture point push, which is a symmetrical game mode, in which both teams must capture all the control points in order to win. The rotational symmetry in the level allows for an even playing ground for both teams whereas in asymmetrical maps, two rounds are required to be played for a fair score outcome. (Valve 2014.)

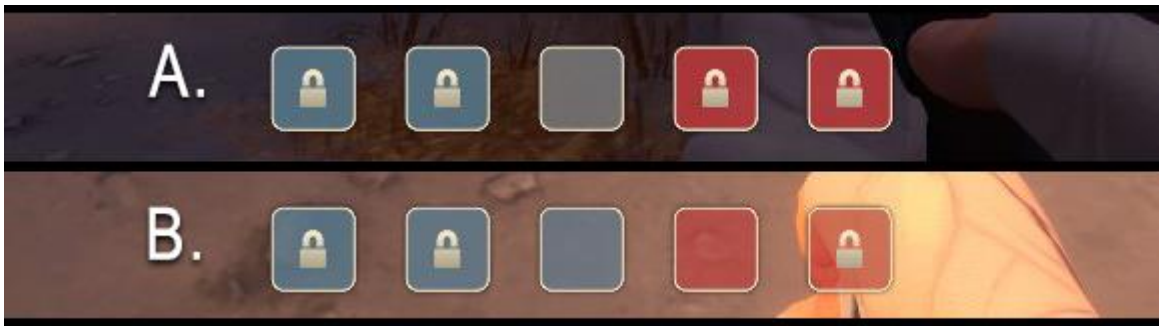


Figure 1. HUD layout for five control point push game mode (Valve 2019)

Figure 1. depicts the in-game HUD layout for a level with five control point push game mode. At the beginning of the match (case A.) both teams start with two control points in their team's side of the map and the middle control point is neutral, which means not one of the two teams owns it. At the start of a match both teams rush towards the middle control point and try to capture it. Once one team has captured the middle control point, the enemy teams next point is unlocked (case B.), and they can continue pushing towards the enemy base. The defending team is always able to push back towards the points the enemy owns, but only two control points are active at a given time. The match ends once one of the teams manages to successfully capture the enemy team's last point, awarding them a score point. Capturing a control point also awards the team a new spawning position, referred to as a forward spawn, that is a bit closer to the enemy team.

4 SOURCE ENGINE

The game engine used in Team Fortress 2 and in this thesis is called Source. It is Valve Software's own, mainly 3D game engine, used by Valve themselves as well as many third-party companies (Valve, 2018). While the engine itself is highly out-dated technology in today's games industry, even games as recent as Electronic Arts' Titanfall and Titanfall 2 use a heavily modified version of the engine, as stated by John Shiring in an interview conducted by Kevin Dunsmore (2016) for Hardcoregamer.

4.1 BSP format

The source engine uses the BSP file format, descendant from Half-Life 1’s game engine called GldSrc, which in turn is based on the Quake 2 BSP file format. The format stores most of the information the game engine needs to render and play a map in the game. This information includes elements of the level such as all the polygons of the level’s geometry, references to the names and the orientation of textures to be drawn on the polygons, the physics data used to simulate physics inside the level and the location all the entities used in the level. More information on the types of elements and entities the level is constructed from will be listed in the next chapter. The BSP file can also contain textures and assets used in the level which are not included in the game’s files. The assets can be directly loaded from the BSP file’s Pakfile lump. Lighting in Source engine levels is mostly pre-calculated and stored inside the BSP file. For the designer to view the level inside the engine with lighting, it must be compiled, and lighting must be calculated by the compiler. (Valve 2019b.)

4.2 Source SDK and Hammer world editor

The main program used for creating levels for Team Fortress 2, and any other Source engine games, is called Hammer World Editor. Hammer functions similarly to any other program for creating 3D objects, but the engine being as old as it is, it comes with a fair bit of limitations. In this chapter the thesis will briefly explain the basic building blocks of a level made for Source engine with the Hammer editor.

The most fundamental building block of any Source engine level is called a brush. Brushes are limited to convex shapes, and if concave geometry is needed, it will need to be constructed from multiple brushes. Hammer editor can generate basic primitive shapes out of brushes, but generally they are only used to shape the environment and larger architecture in levels. After a brush has been created, a texture and material can be assigned to its faces. (Valve 2019a.)

Any areas and or objects that require finer and denser detail are modelled in a separate 3D-modelling software and imported into the level as entities. Entities are the second fundamental building block of any Source engine level. There are two types of entities inside the editor; brush-based entities, which are created in the editor but have more capabilities than the inanimate world brushes and point entities, which are objects like props (3D-models), lights and logic entities. (Valve 2019a.)

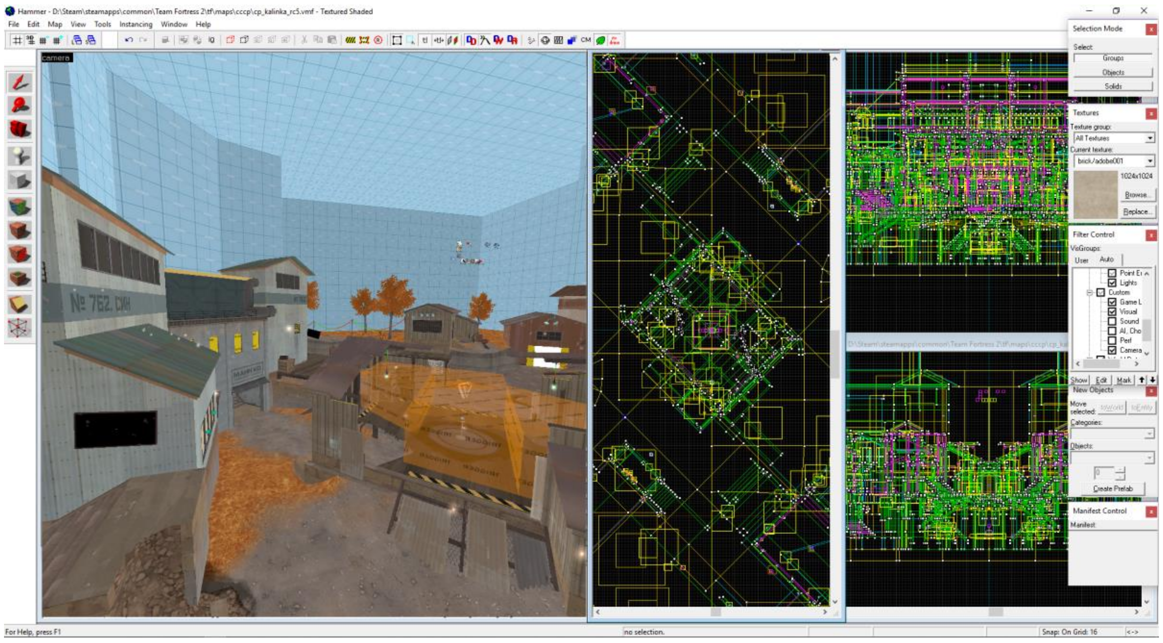


Figure 2. Layout of the Hammer World Editor (Valve 2019)

Figure 2 above visualizes the general layout of the Hammer editor. The left most column features the tools available for the designer; the selection tool, magnify, camera, entity tool, block tool, toggle texture application, apply current texture, apply decals, apply overlays, clipping tool and vertex tool. The most used tools in this list are the selection tool, entity tool, block tool, and vertex tool, all of which are vital in creating levels. The editor also features viewports in both 3D and 2D. The 3D viewport on the left side allows the designer to view a preview of the level and move the camera freely inside the space, and the 2D viewports on the right side allow fine and precise adjustments of the BSP-brushes and entity locations. On the right most column are visible additional features the level designer can utilize. On the top of the column are the different selection modes available. Under that is the texture window where the designer can search for textures to apply to the brush geometry. Next in the column is the filter control window, which allows the designer to hide sets of objects from the view. This is especially useful during later stages of development when there are lots of trigger brushes and other invisible gameplay objects obstructing the view. The new object window is active when creating a new brush or an entity and displays options for them.

5 SOURCE ENGINE LEVEL DESIGN PROCESS

The usual level design process for Team Fortress 2 follows a three-stage format. The first stage of development is called alpha, the second stage beta and the third stage release candidate. The level design process for the thesis level design project, called Kalinka, also followed the same format.

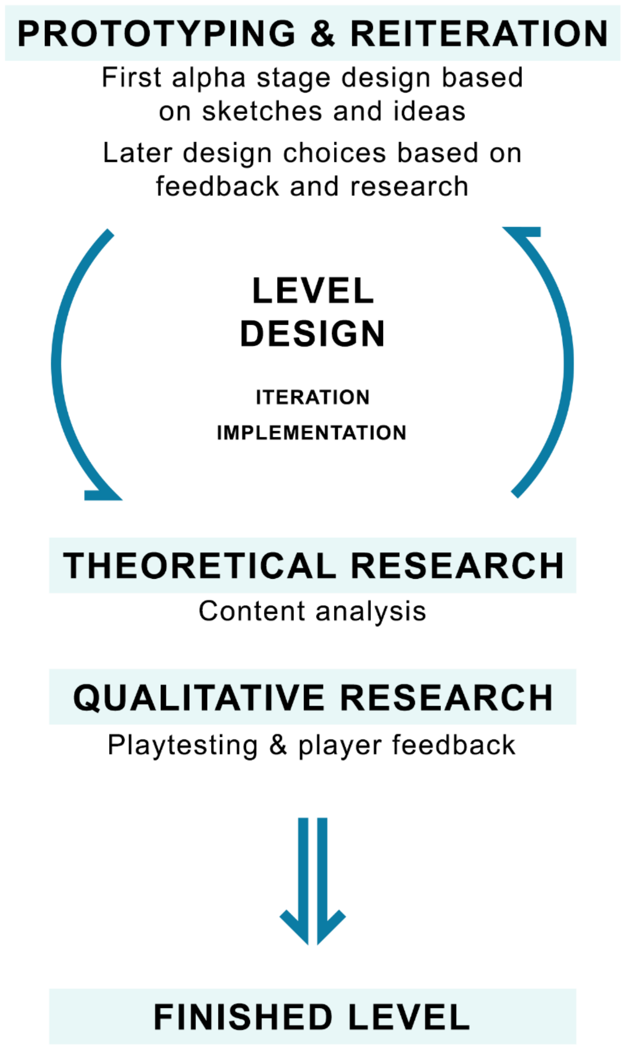

PROTOTYPING & REITERATION
First alpha stage design based on sketches and ideas
Later design choices based on feedback and research
LEVEL DESIGN
ITERATION
IMPLEMENTATION
THEORETICAL RESEARCH
Content analysis
QUALITATIVE RESEARCH
Playtesting & player feedback
FINISHED LEVEL

Figure 3. Thesis project structure

The structure of this thesis project is visualized in Figure 3 above. The first step in the process is prototyping, in which the very first version of the level design is produced according to the sketches and ideas the designer has for the level. Once the level reaches a playable form it is tested numerous times and reiterated according to player feedback. Gameplay is also analysed and problematic areas in the level are identified and redesigned. The design choices are also backed with theoretical research in the form of content analysis. The reiteration process is repeated throughout the three stages of the process until the level reaches a finished stage. The way this process is followed varies from designer to designer and level designers working for a company should of course follow the instructions from the game company.

5.1 Alpha stage

Starting a new level design project may appear overwhelming. Even if the designer has clear vision of what they want to achieve, jumping straight into the editor might not be the optimal way to start a project. Good foundation work is always important and will help the production enormously in the long run. The most important concept to figure out before starting is: “what makes this level fun for the player” and how to support the fun already included in the game mechanics with your level design. Fun is a board term and defining it exactly would be an impossible task; for this reason, a level designer must know the project or game the level is to be a part of and trust their instinct as well as listen to feedback to make the design choices. (Byrne 2004, 58.)

A good starting point for some designers is a blueprint sketch of the level. This way the designer can visualize the ideas on paper in a coherent form and identify any apparent problems in the layout before starting. It is of course hard to describe height variation on the blueprint format. Sketching out on paper and creating prototype blueprints also allows for the designer to examine and evaluate their layout ideas as whole. The original layout sketch for Kalinka is portrayed in comparison with the current, finished layout of the level in Figure 4.

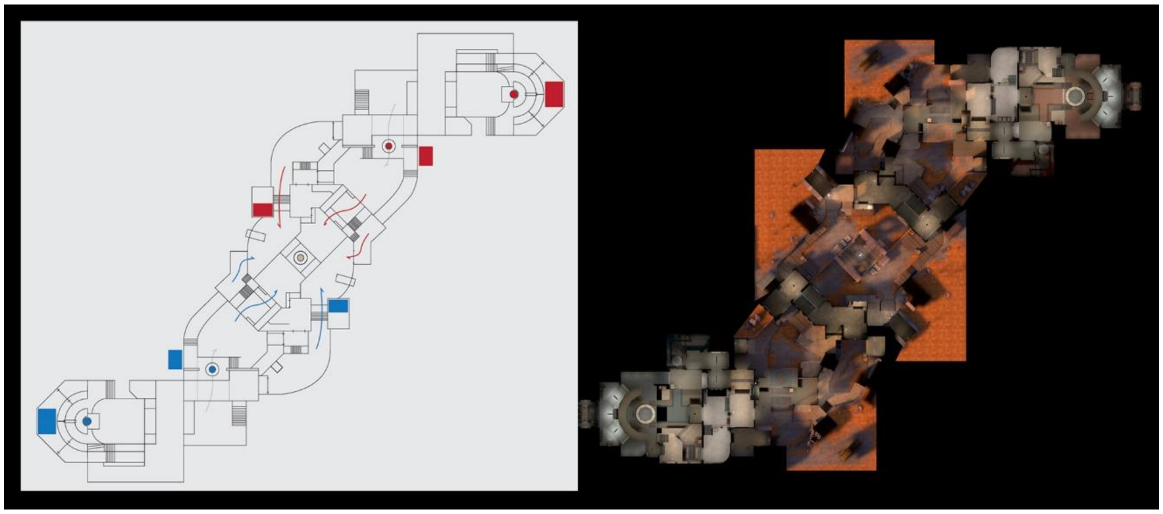


Figure 4. First level blueprint sketch of Kalinka compared to the final layout

Without proper planning a level might feel unfocused and confusing for the player. The player will be experiencing the level designer’s work closely and if the level designer lacks vision and focus the player will most likely experience the lack of them as well. For this reason, proper preproduction work is essential in the beginning of alpha stage. (Galuzing 2011, 16.)

Once the designer has completed enough planning in preproduction, the process can move on to the level editor. The first step for the designer is to block-out the form of the level in simple geometric shapes. This process is also often called grey-boxing because many of the placeholder developer textures used in this stage in the industry are simple grey coloured textures, sometimes with grid measurements on them. Figure 5 below is a screenshot from the very first alpha version of Kalinka, in which grey-boxing has been completed and the level has reached a playable state.

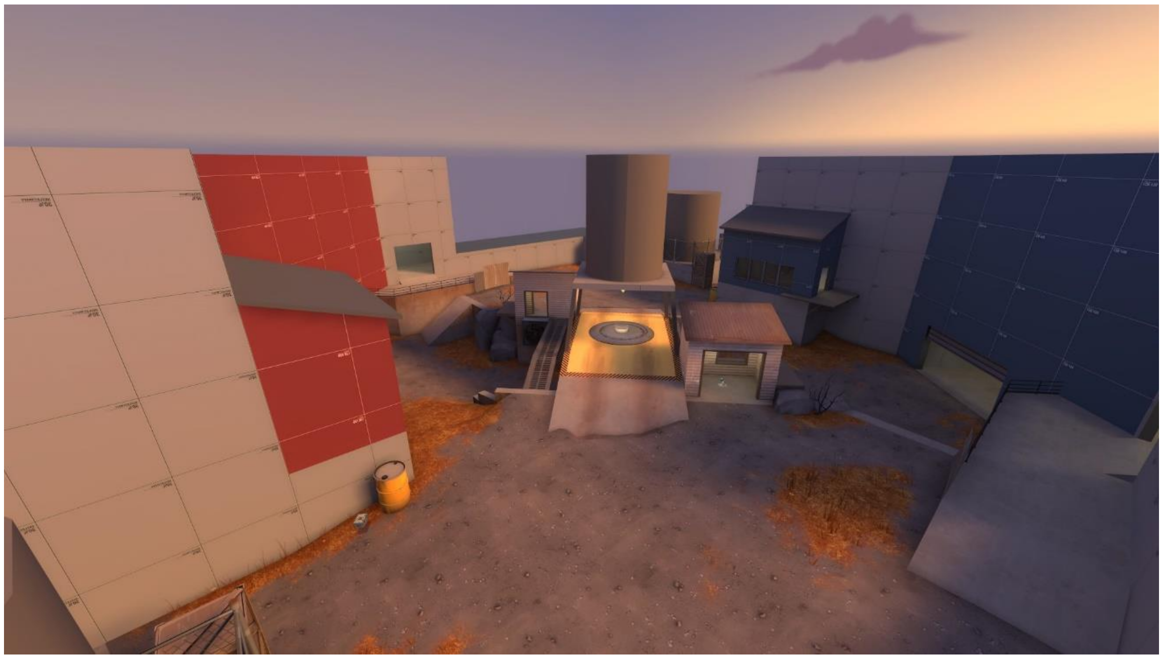


Figure 5. The first alpha version of Kalinka greyboxed

During the entire level design-process the level should always be tested as much as possible. Testing in-house is useful whenever possible, but the level should be tested with a larger player base to ensure the feedback is honest and represents how the average player feels. Testing in-house may sometimes lead to inadequate level design choices as the developer team (or hired testers) may already be familiar with the games' mechanics, and do not know how the level feels for a completely new player. (Galuzing 2011, 39.)

Gameplay testing the level is especially important during the alpha stages of development. When a new version of the level is released it should be tested a couple of times, or enough number of times to gather enough feedback to push out a new version. However, all the feedback should not be taken as the absolute truth, but rather proper evaluation of the feedback is needed before changes are pushed to the design. The designer may ask themselves questions such as: "What seems to be the general opinion?" and "Which areas appeared to be problematic?".

Once the designer has decided on what are some of the current issues in the design, they should work on the problems and push out a next test version. This process is to be repeated until the gameplay of the level is up to the desired standard and the players do not seem to have any unified annoyances with the level. Testing the gameplay of a level is always good to do as early as possible. Usually the gameplay of the level should be as refined as possible before the level moves to the next stage in the process and is art-passed. This way the levels visuals and art will not need to be changed each time some of the level's geometry is changed around. Figures 6 and 7 below represent two different versions of the level in which the implementation of play testers' feedback is visible.

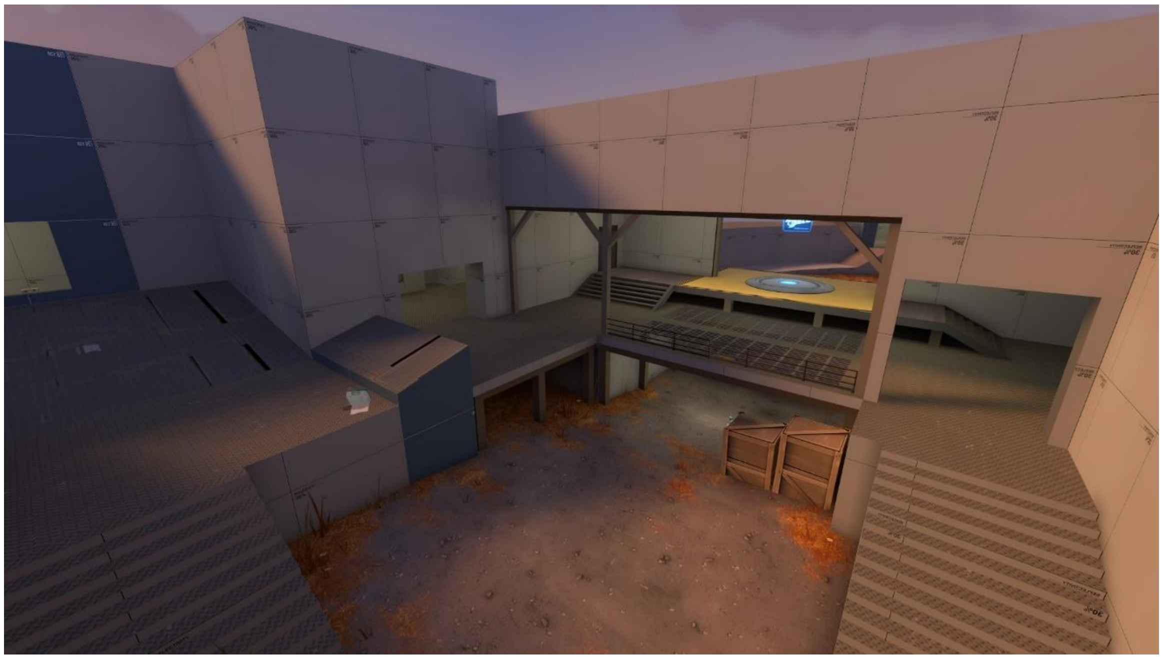


Figure 6. Earlier alpha version of Kalinka.

In the earlier version of the level the testers reported that the roof on top of the capture point was restricting the vertical gameplay of the area (Figure 5). This feedback was deemed correct by the designer and a redesign of the area was necessary.

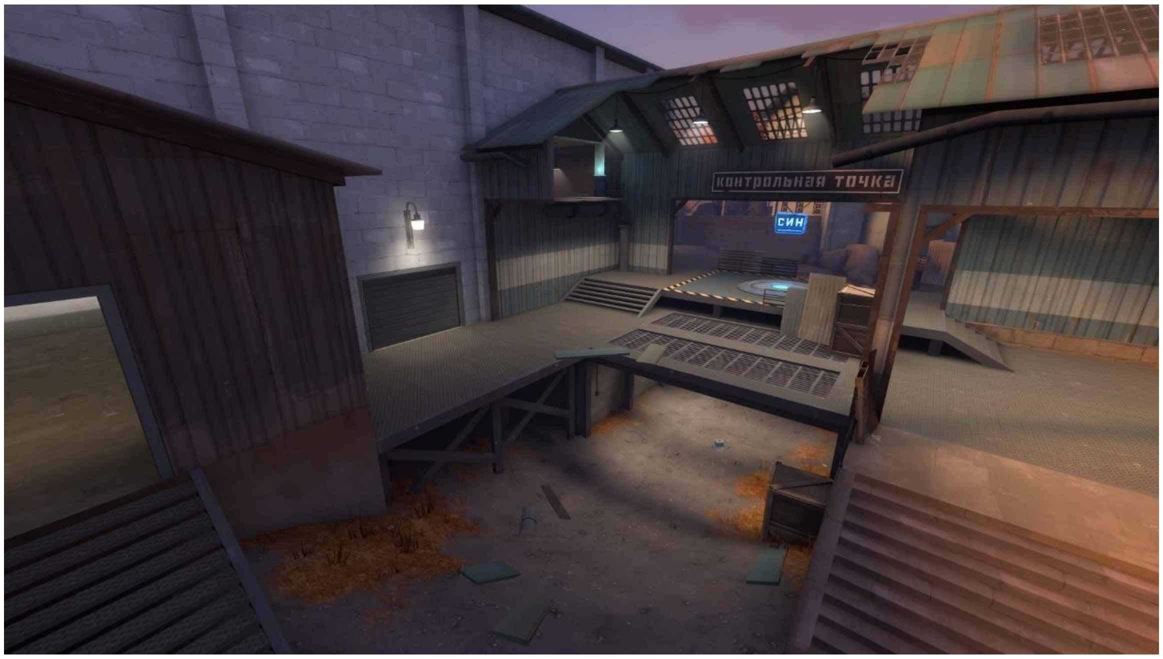


Figure 7. Area redesigned in line with players' feedback further in development.

The roof was then redesigned to be higher and broken open above the point to allow more vertical gameplay in the area in later versions. Gameplay testers felt the area had improved so the iteration was kept (Figure 6).

5.2 Beta stage

The next step in the three staged development process used in this thesis project is the beta stage. If the process has been followed correctly, the gameplay of the level should be at a nominal point, and the level can be art-passed. This means that the grey-boxed placeholder geometry is detailed to fit the games art-style and the theme that the designer has decided for the level. The level should still be tested during the beta stage and the final problems in gameplay should be refined out. Considerable layout changes to the gameplay areas require more work during the beta stage, as the designer needs to move all the detailing together with the layout. However, gameplay changes during the later stages of the development are not impossible and should be done when the level requires them.

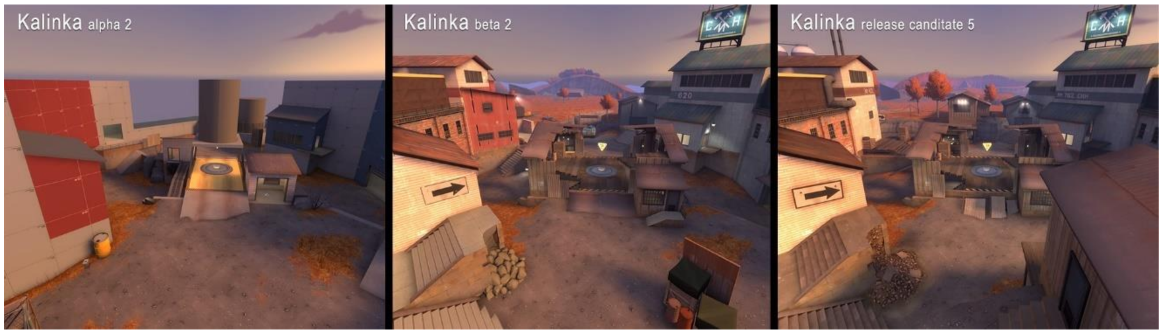


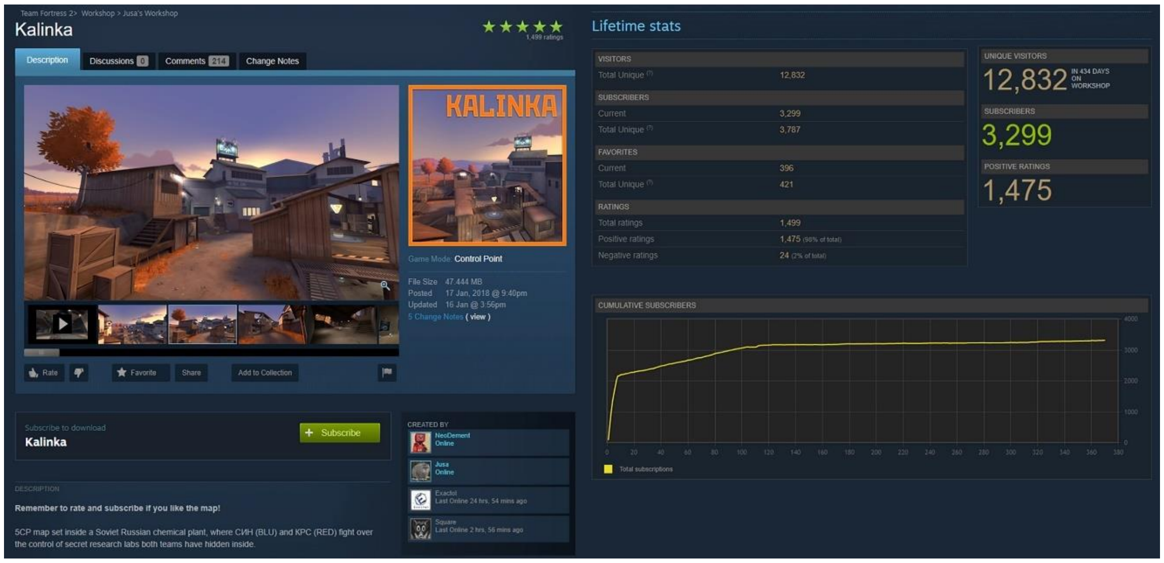
Figure 8. Comparison between a greyboxed alpha version, the first art-passed version and the final version of Kalinka.

Figure 8 above represents a comparison between alpha stage, beta stage and the final art-pass from the thesis project, Kalinka. The detailing of the level may still change drastically after the very first art-pass, hence multiple passes may be required to reach the wanted outcome and look for the level. As seen in Figure 7, the first art-pass may still be different from the final look of the level.

5.3 Release candidate

The third and final stage in the used level design process is called release candidate. The name stems from the fact that the designer deems the level to be in a state fit for bigger public release. During the release candidate stage the levels gameplay should be at a final level and major changes to the layout will require a substantial amount of unnecessary work, since the detailing has reached a higher density. Changes are still possible, but it is more optimal to have the gameplay at a final stage before moving to release candidate.

When it comes to levels created for Team Fortress 2, once the level has reached release candidate it is also a prime time to release the level to the Steam Workshop to gain some attention for the level. The level should only be posted to the Steam Workshop at a late stage in development so to not leave negative first-impression on the level for players who view it there. The ratings and views the level will gather during the very beginning of its appearance in the Steam Workshop is vital, as it will be buried among the other levels if it will not reach the featured spot, through which many times more players tend to view new Workshop releases from.



Kalinka

1,499 ratings

Game Mode: Control Point

File Size 47.444 MB
Posted 17 Jan, 2018 @ 9:40pm
Updated 16 Jan @ 3:56pm
5 Change Notes (view)

Subscribe to download
Kalinka

CREATED BY
NeoDement Online
Jusa Online
Exactol Last Online 24 hrs, 54 mins ago
Square Last Online 2 hrs, 56 mins ago

DESCRIPTION

Remember to rate and subscribe if you like the map!

5CP map set inside a Soviet Russian chemical plant, where СИН (BLU) and КРС (RED) fight over the control of secret research labs both teams have hidden inside.

VISITORS	
Total Unique (?)	12,832
SUBSCRIBERS	
Current	3,299
Total Unique (?)	3,787
FAVORITES	
Current	396
Total Unique (?)	421
RATINGS	
Total ratings	1,499
Positive ratings	1,475 (98% of total)
Negative ratings	24 (2% of total)

UNIQUE VISITORS
12,832 IN 434 DAYS ON WORKSHOP

SUBSCRIBERS
3,299

POSITIVE RATINGS
1,475

Figure 9. Kalinka’s Steam Workshop page and subscriber graph. (Valve 2019.)

Figure 9 displays the layout of the Steam Workshop submission page as well as a graph of the subscribes the level gathered during its time in the steam workshop. As it is visible from the graph in Figure 9, majority of attention a level in the Steam Workshop gets is gained from reaching the featured spot in the first days after its release. After the level is no longer featured the views reduce drastically, which is rather problematic in the current state of the Workshop.

6 LEVEL FLOW AND PLAYER GUIDANCE

The next part of this thesis will focus level flow and player guidance, the main subject of this thesis studied in connection with the thesis project, Kalinka. The term level flow is used to describe how well players find their way around the level and how natural it feels for them to traverse from objective to objective. The designer should always be mindful of how the designer intends to guide the player through the level. The designer will eventually know the level they are designing like the back of their hand, so it is important not to look at the design of the level from the perspective of a player playing it for the first time. The most important and best guidance of the player is the kind the player themselves do not even realize is happening. (Byrne 2004, 62.)

When it comes to guiding a player inside a multiplayer level the designer needs to rely greatly on visual hints and clues given to the player. In a multiplayer setting, where the players main goal is to better the other player, the designer must use clues the player can understand easily and swiftly. All the basic principles of visual design also apply to level design. (Ross 2014.)

For this thesis, achieving optimal level flow is categorized into two main categories; Direct guidance and indirect guidance. Both categories list multiple sub categories, each with examples found to be beneficial in practice during the thesis project and the playtests arranged for the project map on its different stages. Some examples are also backed up with references to other levels.

6.1 Direct guidance

With direct guidance the thesis is referring to the sort of guidance the player will notice and realize during gameplay. With this type of guidance, the level designer can guide the player with direct signals at a conscious decision-making level. Using and relying too much on direct guidance may cause the player to feel like none of the choices they made in the level are theirs but rather every step they take is being directed by the level designer. For this reason, a good balance between direct and the indirect guidance, mentioned in the next main chapter, is needed.

6.1.1 Signs

The designer may also use a more direct way to guide the player with assets used in the level. Objects like arrows, road signs and graffiti signs pointing the player to the right direction work well and reinforce the player’s sense of going the right way (Figure 10). Signs like these should always be tied together with the levels story especially if the game uses a realistic visual style. The use of signs should not be too extensive so the player will not feel like they are just following the sign props around. (Galuzing 2011, 89.)

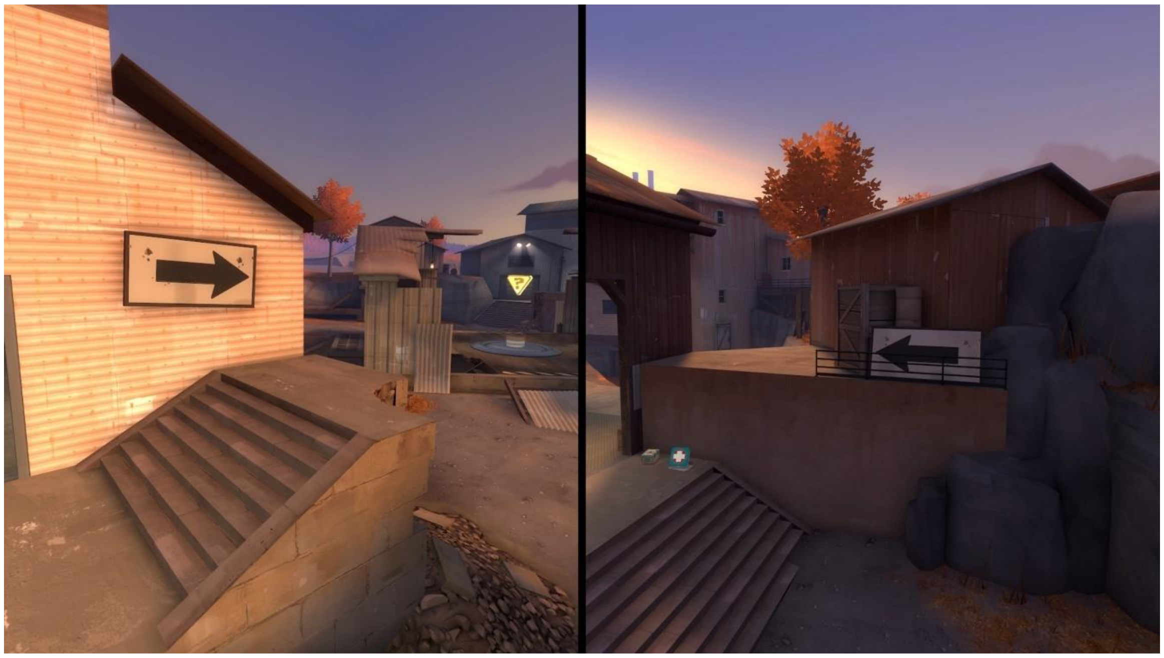


Figure 10. Direct guidance of the player with the use of an arrow sign. (Kalinka)

Too many signs, like the arrow sings visible in Figure 10, could lead to confusion for the player. Arrows should be used to highlight the choices the player has, not to make choices for the player. The level should never take the choice out from the player’s hands, but rather guide the player towards their options, leaving the decision making for them.

6.1.2 Visual gameplay elements

Some of the gameplay elements required for a game mode are also objects inside the level. These act as a form of direct guidance for the player. The objects are recognizable and shared between game modes, so even new players get quickly familiar with the gameplay objects. The objects are often tied together with the game’s HUD.

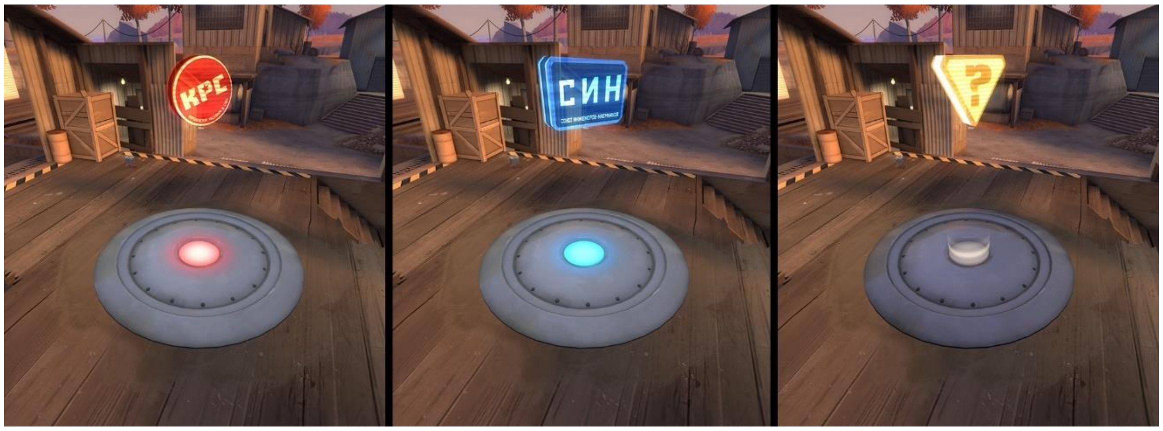


Figure 11. Capture point props in Kalinka.

In this thesis project, a good example is the game’s usage of a recognizable capture point model together with an emissive, bright hologram. The large metal base of the capture point is easily recognizable and informs the players where the capture zone for said point is. This is also usually paired with a clear hazard texture line on the edge of the capture zone. The prop also features a large, emissive hologram, which changes based on the ownership of the control point. This bright hologram draws players’ eye towards. The control point model is visualized in Figure 11 in all three stages of point ownership (RED, BLU and neutral). In Kalinka the holograms were re-textured to feature the Russian localizations for the teams to fit the theme of the level.

Team Fortress 2 also supports an outline feature that directly guides the players. These outlines can be used to highlight important gameplay objects and make them visible through any geometry in the level. This way the player can locate the important objects, even if the objects are dynamic and moving through the level.

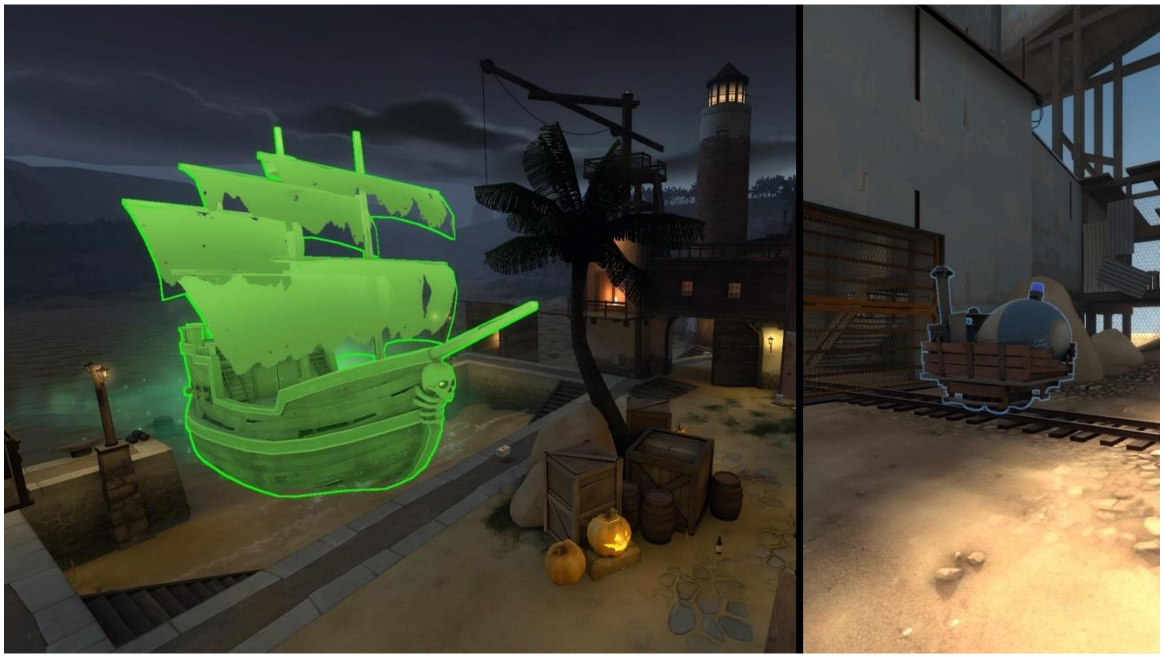


Figure 12. Examples of the usage of outlines (Valve 2019)

Figure 12 features examples of the usage of outlines in Team Fortress 2. On the left side is an example from another level, Cursed Cove, which the thesis author worked on. The large ghost pirate ship moves through the middle of the level and players need to reach it in order to score points. The outline gives the players a direct notion on where the ghost ship is currently at, making it easier for them to reach it in time. On the right side of Figure 12, there is an example from Valve Software’s Team Fortress 2 payload game mode map, Upward. In the payload game mode, the main objective of the level is the payload cart, which blue team needs to push. This object is outlined for a short period of time after a player spawns in to the level. This gives the players an idea of how far their team has pushed in the gameplay space, and an instant notion of where to head.

6.1.3 Texturing and colour coding areas

Another important aspect to guide the player is the use of textures. Practical experience has shown that in a game with two distinct teams identified by the team colour (in my case, red and blue) it is important to texture the teams’ own side of the level with the corresponding colour. This way a player will always know which way is towards the enemy base and which way they must backtrack to find back to their team. If the entire level is textured in a similar manner the players will easily lose their sense of orientation. The use of team corresponding textures is clearly visible in the middle area of Kalinka. Players can quickly identify which side of the map belongs to the blue team and which side is towards the red team. (Figure 13)

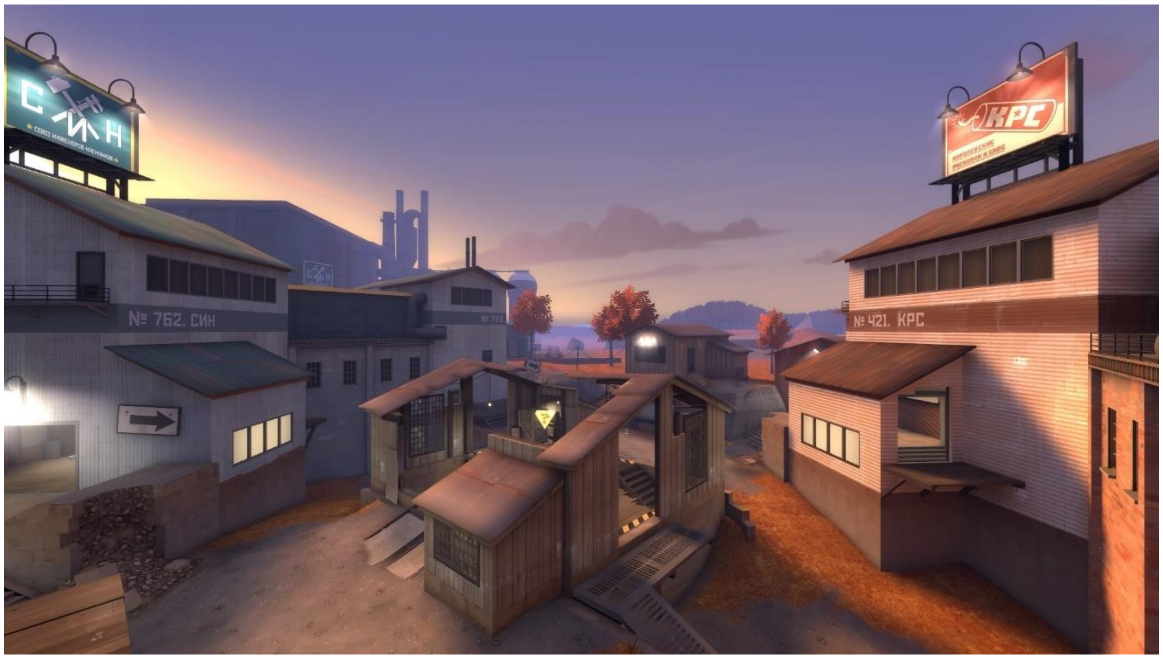


Figure 13. Each team’s side clearly visible in the middle with the use of textures (Kalinka)

The level may also feature signage and overlays depicting each team’s name or other recognizable insignia. Once the player is familiar with the insignia, identifying which teams’ side they are on is easy from just a quick glimpse at signs placed around.

6.1.4 Pathways

Pathways are another great way to direct the player towards objectives. For example, paths can be visualized on the ground with textures, having the sides of the paths be covered in grass and the main form of the path be of gravel. Paths are on the line between direct and indirect guidance and could be listed under both in different cases.

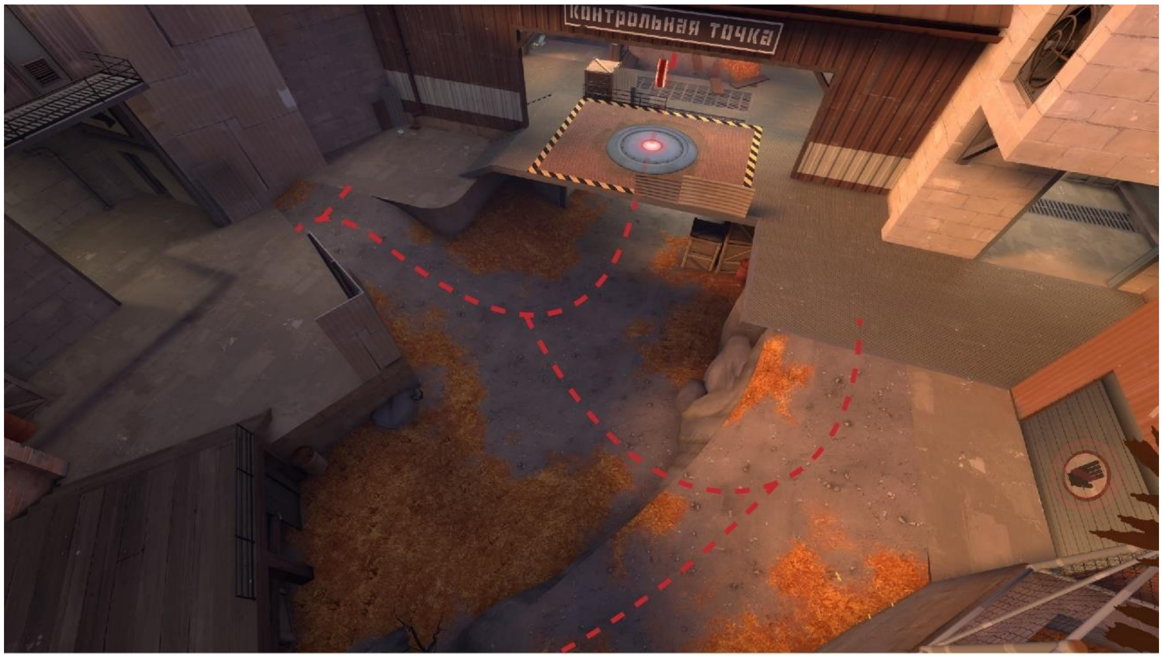


Figure 14. Example of paths in Kalinka

The usage of paths to display different options for manoeuvring the area is visible in Figure 14. The ground is separated into grass and gravel with the use of blending two textures. The gravel gives players an idea of a beaten path, which indicates that the area has been used to travel between areas. The grass is used on the sides of the paths and in the dead end near the bottom left of Figure 14 to indicate that the area does not lead the player anywhere.

6.2 Indirect guidance

In addition to the methods of direct guidance listed in the previous chapter the designer is also able to guide a player indirectly. Indirect guidance differs from direct in that the player does not necessary realize they are being guided with actual design choices. Indirect guidance relies more heavily on the subconsciousness of the player and is more subtle in nature.

6.2.1 Overall layout of the level

The most general way of improving the player’s sense of direction and orientation inside a level is the overall layout of the level. Sharp turns and overly complex layouts will disorient the player easily. Instead, there should always be a clear direction of attack or a repetitive and familiar direction of attack. If the direction of attack is not towards the same point of compass, should the direction still be repetitive and familiar to the player.

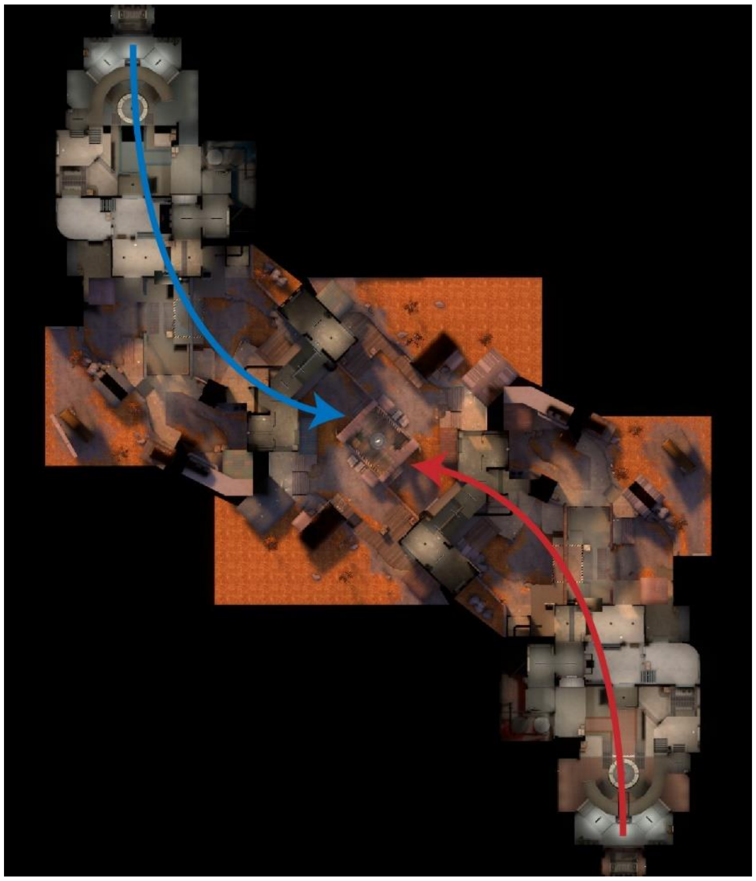


Figure 15. The overall flow of Kalinka

Figure 15 above illustrates the final layout of the thesis level design project Kalinka. In Kalinka, the layout of the level supports the guidance of the player by always keeping the general direction of attack the same for both teams. Even with slight curves in the pathing the players will not get disoriented, as the angles are not too steep and many.

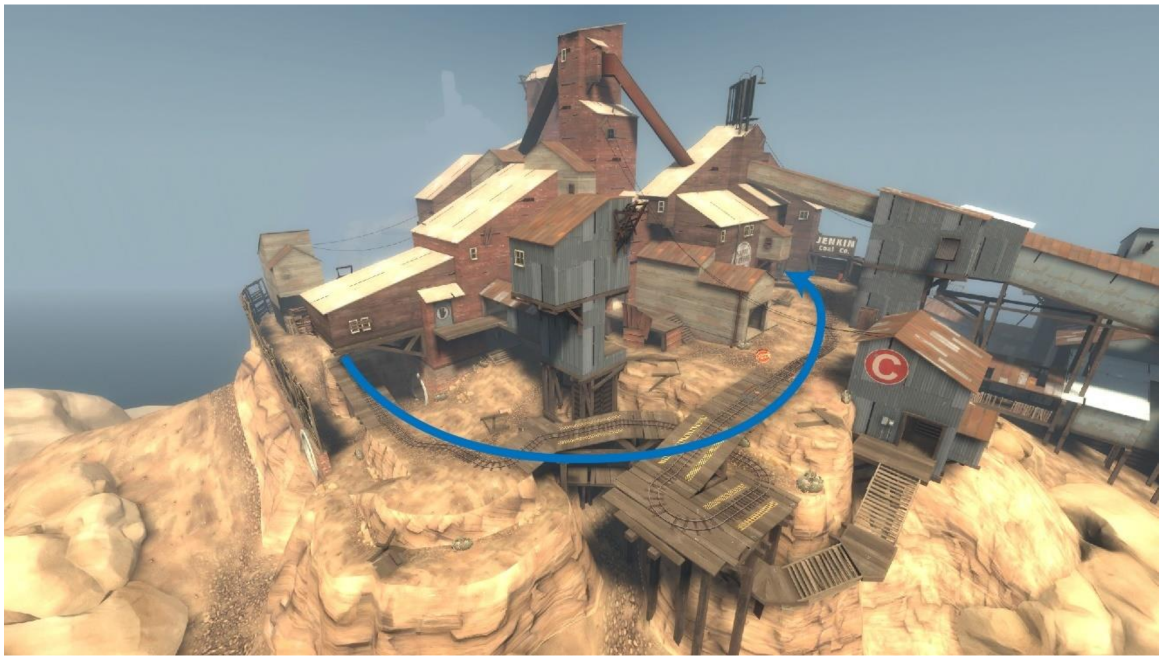


Figure 16. Flow example from payload Upward (Valve 2019)

Another level from Team Fortress 2 by Valve Software that uses this method of guidance extremely well is the payload map Upward. The overall layout of Upward takes the shape of a spiral, as seen in the left side of Figure 16, so the general direction of attack is not the same. Instead Upward makes the direction of attack repetitive and recognizable by always keeping it at an angle upwards and slightly towards the left the players feel more accustomed to it. The direction of attack is also always around the enormous building on the hill on the left for attacking players and attacking players will always have steep deadly cliff on the right. This way the level feels familiar even for the players playing on it for the first time

6.2.2 Landmarks

In addition to lighting, the use of the level geometry is ideal. The use of recognizable, unique objects around a level help to orient the player around. If the game that the level is being designed for is rather fast paced and the players are required to do quick manoeuvres, they could easily lose the track of their orientation inside a level with no landmarks. (Galuzing 2011, 123.)

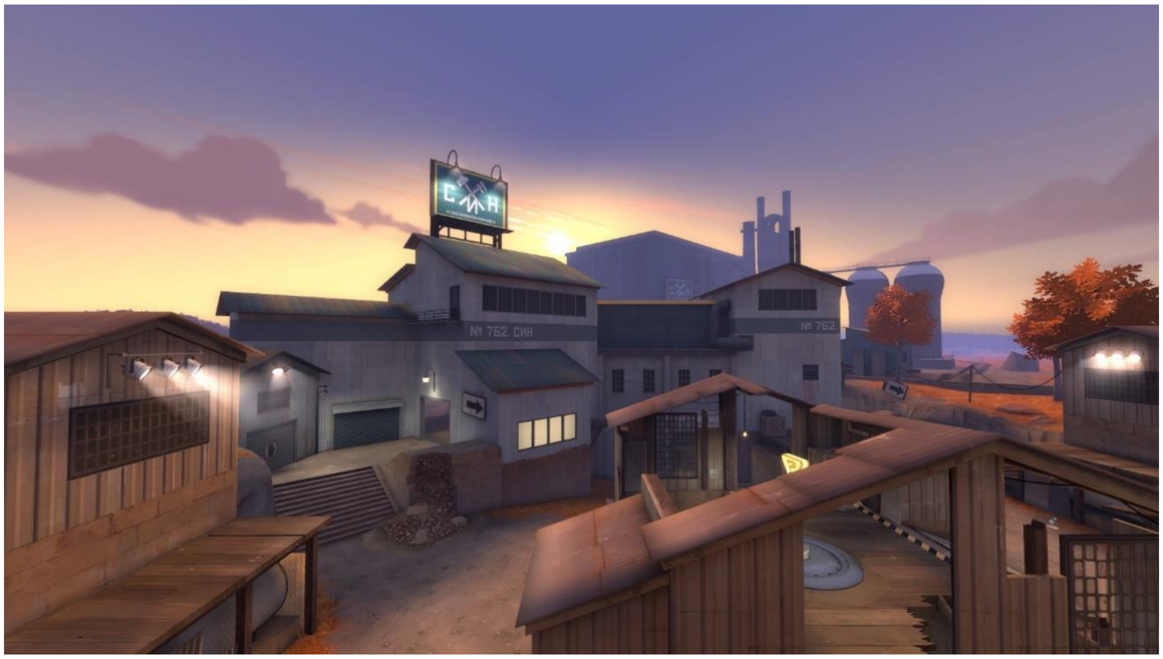


Figure 17: Usage of landmarks in Kalinka

The usage of landmarks is portrayed in Figures 17 and 18. Players viewing the level from the middle area (Figure 17) can see the blue teams’ billboard high up on the rooftops as well as the huge factory in the distance.

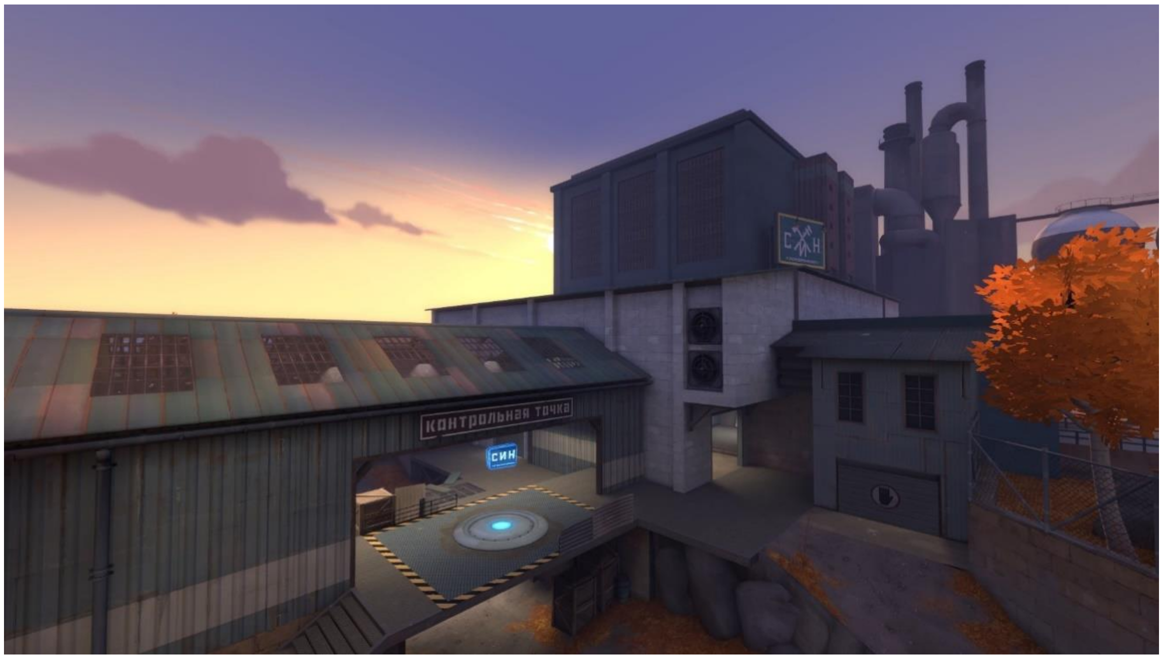


Figure 18: Landmarks visible in the next area in Kalinka

Once the players traverse to the next area (Figure 18) the factory which was visible all the way from the middle of the level is now their next goal to attack. The billboard overseeing the middle capture point which was visible in the middle area portrayed in Figure 17 is also still visible on the rooftops behind the viewing direction of Figure 18, giving players additional landmarks to orientate themselves with if they need fall back towards the middle.

6.2.3 Shapes and geometry

In addition to complete and recognizable objects the level designer can also use patterns to give the player a sense of orientation. Having multiple, too similar spaces in the level will cause confusion for the player and geometry and patterns in detailing can easily distinguish two spaces from each other. Even having one floor at a slightly different level and of different texture than the rest of the space and having recognizable patterns in detailing around the space will give the player discrete elements of larger space, making it easier for the player to understand the larger space easier. (Cox 2015, 00:14:43)

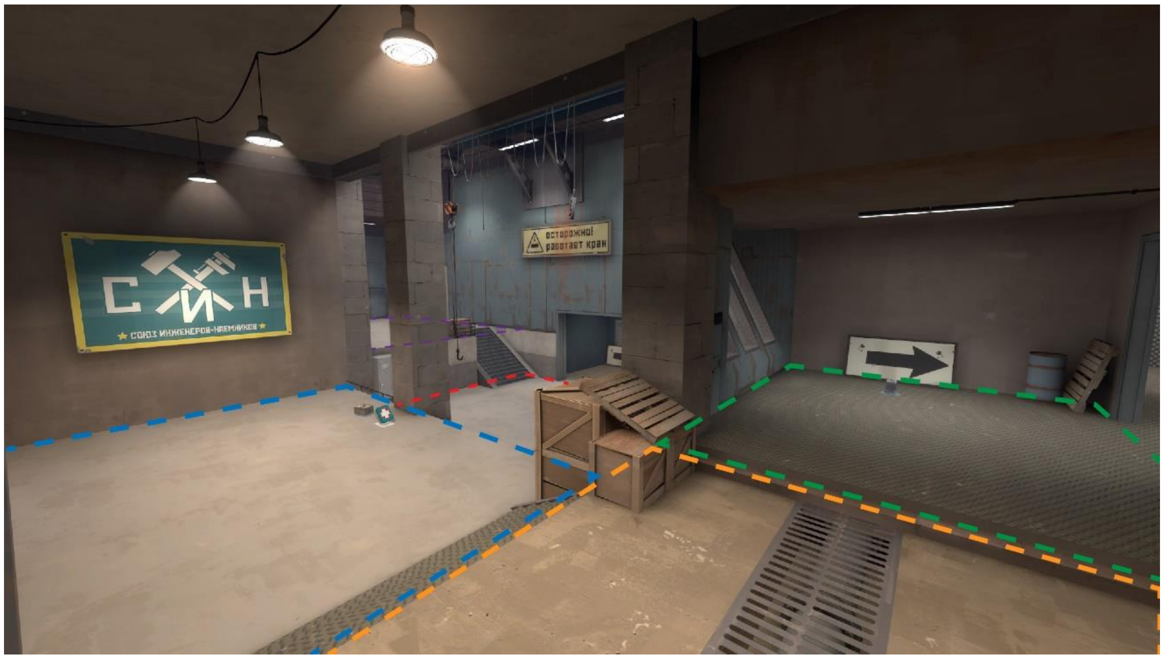


Figure 19. Kalinka interior spaces separated and detailed for easier recognition

Figure 19 above is an example of how indoors areas are separated into multiple smaller spaces to make the area easier for players to understand. Indoors areas also feature unique patterns in the form of detailing. Areas are also detailed in a way that makes sense in the context of the levels theme as to not alienate the player. The dashed lines represent the separation of the room into sub-spaces.

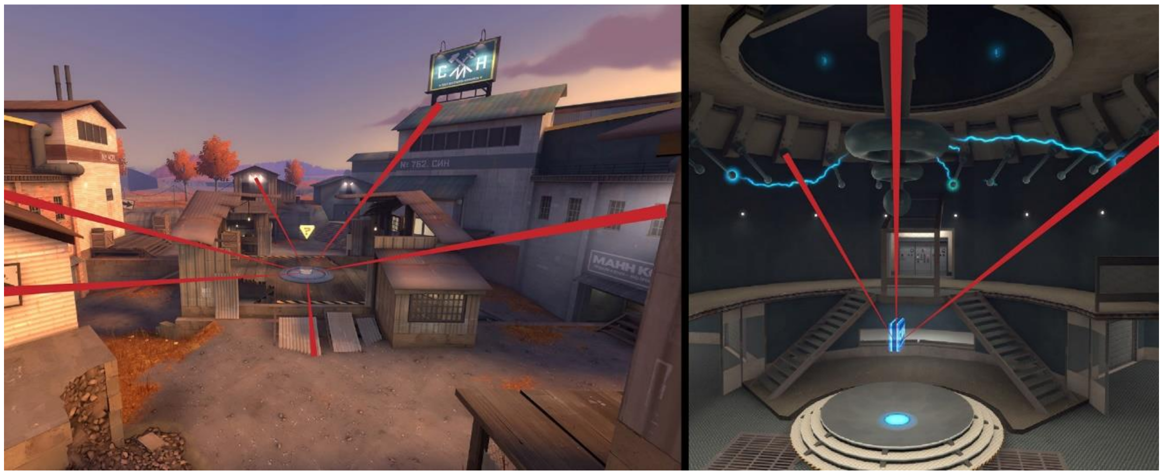


Figure 20. Geometry and lines leading the players eye in Kalinka

As with any form of art and design, composition is important to keep in mind when level designing. One of the most important parts of composition is leading the viewer’s eye, which can also be used to indirectly guide the player. The geometry of the level can be used as a tool of composition by creating compositional lines. It is in the human nature to instinctively focus on the most interesting parts of the view first, which in the case of lines is the intersections they create. Having lines and angles in the geometry slope towards the objectives and important areas creates intersections on the areas, leading the player’s eye towards them. The usage of lines created with geometry and other objects in Kalinka is visualized in Figure 20. The red lines in Figure 20 emphasize how the geometry creates lines in the composition. (Spadin 2014.)

6.2.4 Detail density

Another way to draw the player’s attention towards the important parts of the level is with the density of detailing. When the designer is working on the detailing of the level it is vital to always keep in mind how important each area is for the gameplay and add detail accordingly. Points of visual interest should be created towards points of gameplay interest and areas that server little gameplay purpose should be detailed more sparsely. This method has been widely used in previous Team Fortress 2 levels from Valve software. (Fabry 2010.)

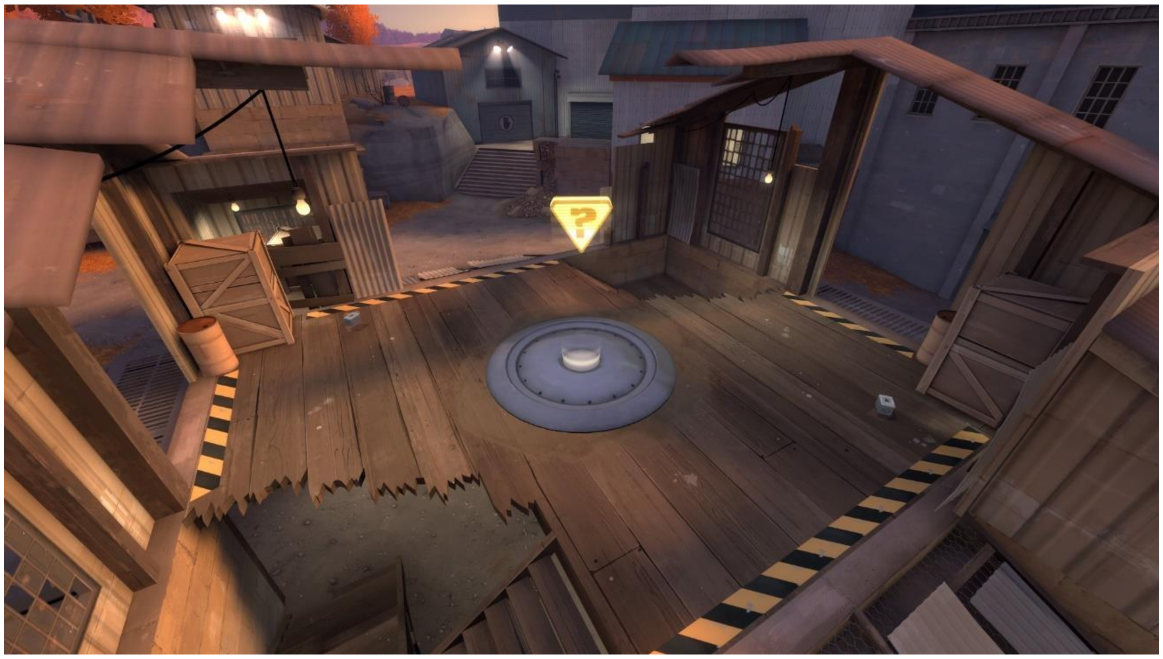


Figure 21. Usage of detail density in Kalinka

Detail density was also used to guide players in the thesis project Kalinka. In the middle point, visible in Figure 21, detail was concentrated at a much higher resolution and density on the important areas, such as the around the capture zone building. Wooden beams, highly detailed broken wood boards on the flooring and various props and light fixtures were placed around the area to draw the player’s eye. The same method was used around all the important gameplay areas of the level.

6.2.5 Lighting

One of the best ways to guide the player towards something is to use the lighting of the environment. Highlighting objects and areas that are important and lead the player towards the fight is a great practice and most importantly, the players do not feel like they are pushed to go into a certain direction. Justifying a contrasting light is always a matter of the theme of the level and is up for the designer to figure. As an example, there could be a warm light inside a building the player should head into, while other buildings would be dark and more ambiently lit. (Galuzing 2011, 84; Jensen 2012.)

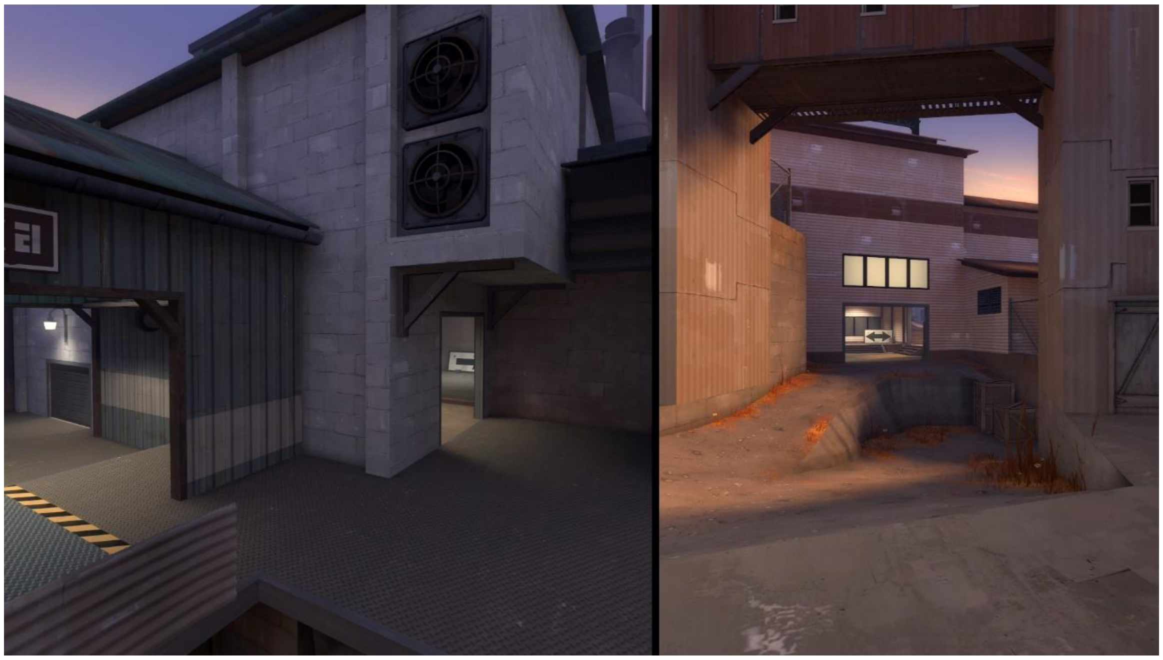


Figure 22. Players' attention drawn to the important areas with lights in Kalinka

Bright lights on the important areas catch the player's eye and the player subconsciously is drawn towards them. The usage of lighting as means of drawing players' attention is portrayed in the Figure 22 above. On the left side of the Figure 22 important doorways leading to the next area are highlighted to stand out from the rest of the area and on the right side of the figure the indoors area the players should push towards is lighted brighter than the area leading to it to draw the players' attention towards it.

6.2.6 Spawns

Guiding the player in the level can also be about quite small, generally un noticeable design choices. For example, when the player first enters the level, they will stand on the specified spawn point. With spawn points, the level designer can face the players towards the general direction the player should head to. This allows new players who are still orienting themselves around the level to walk straight out and head towards the next focal point of the level. Players who spawn facing a wall, or to the wrong direction, away from the gameplay, will experience unnecessary confusion as they need to look around and figure out which way the other players are playing at. (Galuzing 2011, 94.)

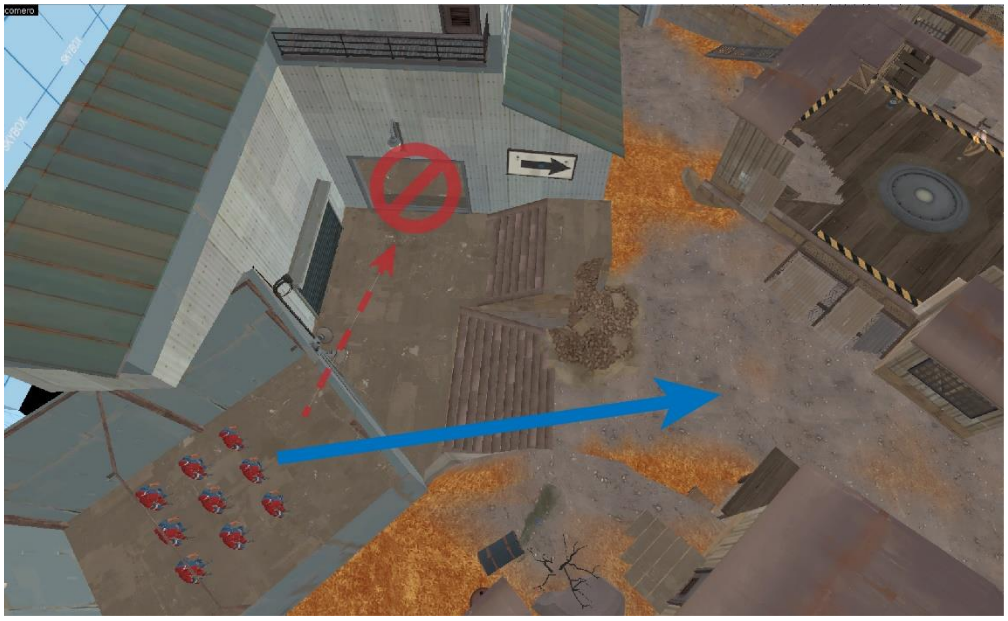


Figure 23. Spawn point entities facing the next objective in Kalinka

Figure 23 is an example of how the orientation of spawn point entities can be used to guide the player as soon as they spawn into the level. The picture is the second forward spawn in Kalinka taken from inside the Source engines level editor Hammer. In a previous version, players spawning in this spawn often followed the dashed red line towards the wrong direction. The problem was fixed by rotating the spawn points’ orientation towards the right ever so slightly, causing players to instinctively follow the correct path marked in Figure 23 with a blue arrow. This showcases how even a tiny change can have drastic effects on the flow of a level.

6.2.7 Restricting player movement

Every level needs to have its edge so as to keep the players together and the gameplay interesting. The level boundaries can consist of cliff walls, hills, landscape, fences and similar objects. These boundaries should be tied together with the theme of the level and the boundaries should feel believable and readable. The player should be able to tell which areas are off-limits and which can be accessed. The designer should also make the gameplay areas more interesting and eye-catching than the non-playable spaces so the players will instinctively know where to go. (Galuzing 2011, 65.)

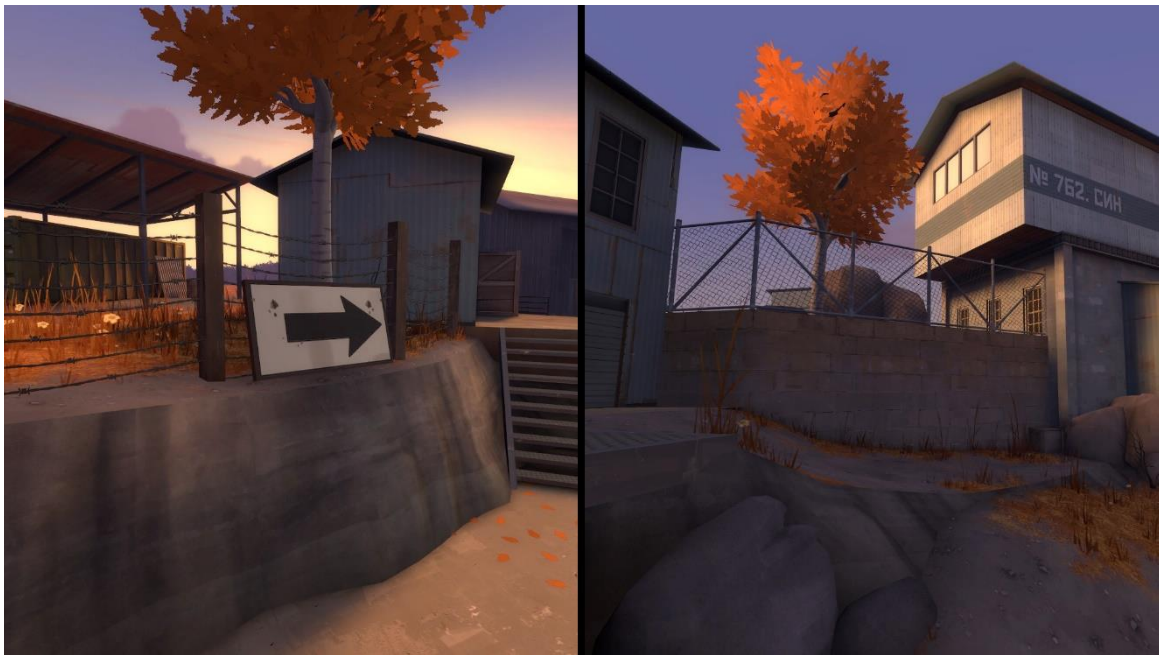


Figure 24. Restricting the player movement in Kalinka

Figure 24 visualizes examples of restricting player movement. On the right side of the figure the usage of a barbed wire fence, cliff and an arrow pointing towards the next objective quickly advice the player that the area is off limits and rather they should head right, where the next capture point quickly draws their attention. On the right side a high bland wall is used in conjunction with a chain link fence, which are often used to mark the boundaries of the map in Team Fortress 2. With the wall and fences being so high up, the player does not feel like they should be able to jump over them.

6.2.8 Sound

The usage of sound inside a level is also a viable, but maybe the most objective way to guide the player. Some levels in Team Fortress 2 use sounds such as warning sirens to draw attention to dynamic elements, such as large doors. When the level is being player by large numbers of players the sound of the gameplay action will most likely overshadow any sounds in the level. Still, the designer should keep the audio in mind and how it can be used to improve the level.

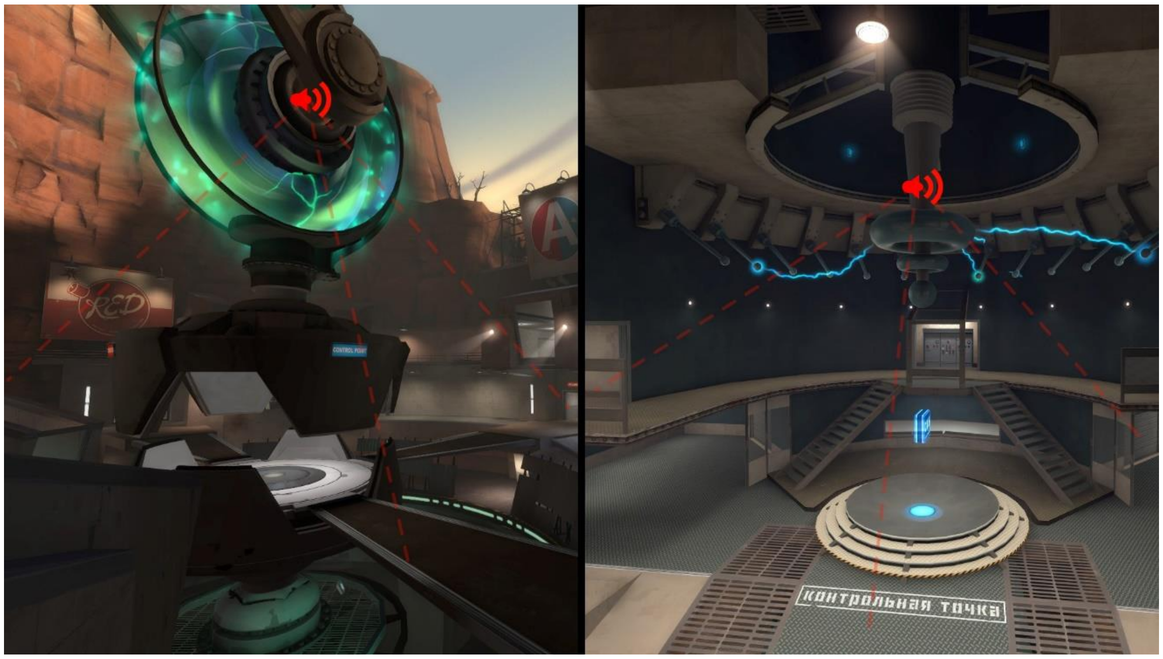


Figure 25. The usage of sound visualized in the levels Kalinka and Nucleus (Valve 2019)

Sound was used to guide the player in a very minimal way in Kalinka, as the level does not feature any large dynamic elements that would require warning sounds. The soundscapes of the level were however designed with player guidance in mind. The right side of Figure 25 depicts the last control point in Kalinka, where a loud electrical sound was used to draw player's attention towards the tesla-coils above the point. This way players can always hear the point even when slightly outside the main space of the last control point. Similarly, players are guided towards the central control point in Valve Software's Team Fortress 2 king of the hill map Nucleus, visible on the left side of Figure 25. The enormous reactor above the control point emits a continuous sound audible all around the level, giving players a general idea of its location from just the audial feedback.

7 CONCLUSION

During the development process of Kalinka and the various gameplay tests held along the development, it was proved in practice that implementing these methods of guidance into the level allowed for a more enjoyable and uninterrupted gameplay experience for any of the players attending the playtests. In playtests of the final versions of the level design, no players were getting lost or reported that they had trouble finding their way around the level. Even for an aspect of level design as hard to numerically evaluate as optimal level flow, this should serve to prove that the ways of player guidance portrayed in this thesis have achieved proper flow in the design. In conclusion, this thesis work should work as an example of how the different methods of both direct and indirect guidance are to be used in a balance to achieve optimal level flow and unhindered gameplay.

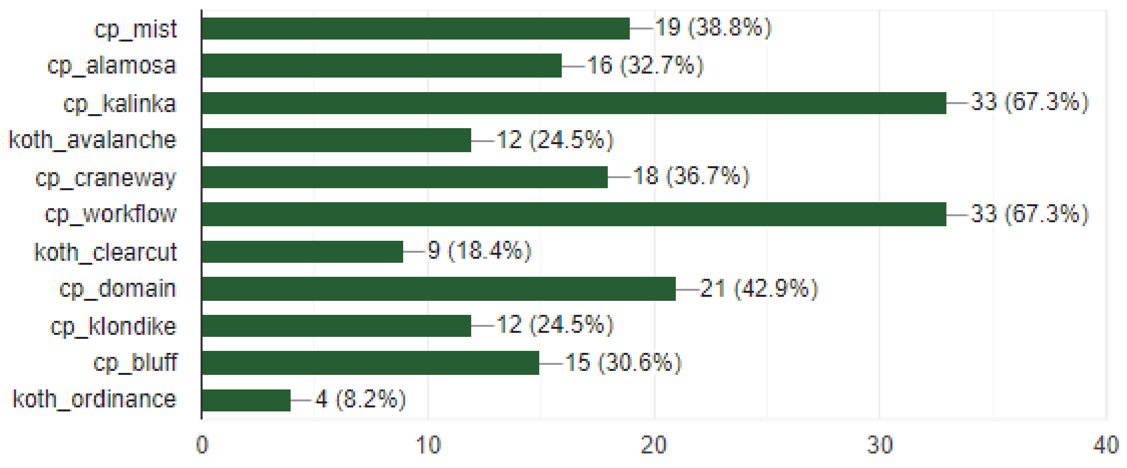


Figure 26. Poll of maps tested by high-level competitive players (Pearson 2019)

The thesis project has also proven to work well as a level design, gaining notion both in the Steam Workshop from more casual players of Team Fortress 2, as well as players from the competitive community formed around the game. Kalinka ranked among the most liked maps in a recent poll by the organizer of the tests. The participants in the poll were mainly high-level competitive players, who helped playtesting the maps and were asked each to select four, in their opinion, most league ready and competitively viable maps from the list. Overall the poll had 49 participants. The poll results are visible in Figure 26.

REFERENCES

Byrne, E. 2004. Game Level Design. Rockland, MA: Charles River Media.

Cox, D. 2015. Interior Design and Environment Art: Mastering Space, Mastering Place. GDC talk. Available at: https://www.youtube.com/watch?v=WWXsmnlmADc [Accessed 23 March 2019].

Dobson, J. 2017. Drool: Orange Box goes gold! WWW document. Available at: https://www.engadget.com/2007/09/27/drool-orange-box-goes-gold/ [Accessed 13 March 2019].

Dunsmore, K. 2016. E3 2016: Respawn Talks Content Variety, Reworekd Engine in Titanfall 2. WWW document. Available at: https://www.hardcoregamer.com/2016/06/14/e3-2016-respawn-talks-content-variety-reworked-engine-in-titanfall-2/212196/ [Accessed 13 March 2019].

Fabry, W. 2010. TF2: Density of Detailing. WWW document. Available at: http://www.nodraw.net/2010/08/tf2-density-of-detailing/ [Accessed 26 March 2019].

Galuzing, A. 2011. Ultimate level design guide. E-book. Available at: https://www.worldofleveldesign.com/categories/books_dvds/ultimate-level-design-guide-11day-level-design-guide-ebooks.php [Accessed 26 February 2019].

Jensen, M. 2012. Functional Lighting. WWW document. Available at: http://magnarj.net/article_funclight.html [Accessed 25 March 2019].

Ross, B. 2014. The visual guide for multiplayer level design. WWW document. Available: http://bobbyross.com/blog/2014/6/29/chapter-5-orientation-navigation [Accessed 13 March 2019].

Shahrani, S. 2006. Educational Feature: A history and analysis of level design in 3D computer games. WWW document. Available at: http://www.gamasutra.com/view/feature/131083/educational_feature_a_history_and_.php [Accessed 13 March 2019].

Spadin, I. 2014. Composition and you. WWW document. Available at: https://tf2maps.net/threads/guide-composition-and-you.23566/ [Accessed 25 March 2019].

Valve Software. 2014. TF2 Design Theory. WWW document. Available at: https://developer.valvesoftware.com/wiki/TF2_Design_Theory [Accessed 13 March 2019].

Valve Software. 2018. Source. WWW document. Available at: https://developer.valvesoftware.com/wiki/Source [Accessed 13 March 2019].

Valve Software. 2019a. Level Design Overview. WWW document. Available at: https://developer.valvesoftware.com/wiki/Level_Design_Overview [Accessed 19 March 2019]

Valve Software. 2019b. Source BSP File Format. WWW document. Available at: https://developer.valvesoftware.com/wiki/Source_BSP_File_Format [Accessed 13 March 2019].

LIST OF FIGURES