



Team Kaijū

Maximilian Mayer

Lukas Goll

Tim Kaiser

Evgenija Pavlova

Contents

1. Formal Game Proposal	3
1.1. Game Description	3
1.1.1. Game Idea	3
1.1.2. Graphics and tone	3
1.1.3. Gameplay	4
1.1.4 About the background	4
1.1.5 Concept Art and sketches	5

1.2. Technical Achievement	7
1.2.1. Core technical feature	7
1.2.2. Challenges	7
1.3. “Big Idea” Bullseye	8
1.4. Development Schedule	8
1.4.1. Plan in Layers	8
1.4.2. Task Timeline	11
1.4.3. Task List	14
1.5. Assessment	16
2. Prototype	17
2.1. Overview	17
2.2. Gameplay	17
2.2.1. Attacks	18
2.2.2. Games Rules	18
2.3. Experiences and Learnings	19
2.4. Revisions to the Game Idea	20
3. Interim	21
3.1. Feedback review	21
3.2. Game Development	22
3.2.1 Level and Task System	22
3.2.2 Static Enemy AI	22
3.3. Character Development	22
3.3.1 Character : Camera & Actions	22
3.3.2 Chaining Destruction	23
3.3.3 Input System	23
3.4. Design	24
3.4.1 Color Scheme	24
3.4.2 Character	24
3.4.3 Environment and Obstacles	25
3.5. Environment & Destruction	26
3.5.1 Procedural Generation	26
3.5.2 Asset Pipeline	27
3.5.3 Destruction	27
4. Alpha Release	28
4.1 Design and Changes	28
4.1.1 Godzilla	28
4.1.2 The City	28
4.1.3 Cars as enemies	29
4.1.4 UI	31
4.2 Procedural Map Generation	32
4.3 The Game	33

4.3.1 Stages of Growth	33
4.3.2 Balancing the Game	34
4.3.3 Player's full Action Set	35
5. Playtesting	37
5.1 Playtest Setup (Physical + Explanation + Softwarewise)	37
5.2 Questionnaire	37
5.3 Results	37
5.3.1 General Information	37
5.3.2 Game Questions	37
5.3.3 Control and Interface	37
5.3.4 Final Questions	37
5.4 Changes	37
5.4.1 Gameplay/-design	37
5.4.2 Controls	38
5.4.3 UI (J)	38
5.4.4 Bugs	38
5.4.5 Appeal/Graphics/Feedback (sprich Häuser + Animation)	39

1. Formal Game Proposal

1.1. Game Description

The game will follow the theme of destruction, by enabling a player to destroy a city as Kaijū Monster.

1.1.1. Game Idea

Whereas the Japanese word Kaijū can be translated to strange beasts and was used to describe legendary creatures and the concept of paleontology, in pop culture it refers to unstoppable powerhouses of nature such as Godzilla or King Kong. For the theme of this year's Games Laboratory: destruction, our team plans to develop a game, in which the player takes the role of a Kaijū to destroy a whole city. The goal in our game is going to be to destroy a massive landmark, such as a radio tower or skyscraper in an urban area. The monster won't be able to do so right from the beginning. Rather we are planning to have multiple stages of growth for our beast. To be big enough to reach that target, the Kaijū has to destroy smaller objects around the city to grow. With increasing size, its power to destroy will grow too, until finally the target can be reached and destroyed. Of course, there will be efforts from humans to stop that monster. Therefore the player needs to act strategically, by planning a route of destruction through the city while avoiding enemies.

1.1.2. Graphics and tone

While usually Kaijū movies are themed as catastrophic and correlate to a sense of threat, we do not aim for a serious tone for our game. Rather we want to emphasize an almost childish experience of destruction, similar to destroying a sandcastle or Lego built city. Therefore, we are going for friendly graphics, cartoonish designs, and brighter colors.

(See 1.1.5)

Choosing voxels for our visual representation of our world is going to support that playfulness. It will give our world a sense of a playground for our Kaijū to have some fun smashing buildings and cars. It also helps us with the creation of 3d assets and their destroyed representations. While the city will be aligned on a grid, dynamic objects like cars, helicopters and the monster itself.

1.1.3. Gameplay

The player controls the Kaijū from a third-person perspective. To thresh through objects, he is able to move the monster freely through the voxelated city, which will consist of static (e.g buildings, bushes, park seats ...) and dynamic objects (e.g cars), with relative different sizes.

At the beginning of the game, the Kaiju is rather small. Therefore, the player needs to navigate in streets and other urban places to destroy objects that are of similar height. By destroying objects the character grows, most likely implemented in discrete steps, after certain thresholds of destruction are reached. He will need to undergo levels of growth until he can take on the target landmark, the final objective of the game.

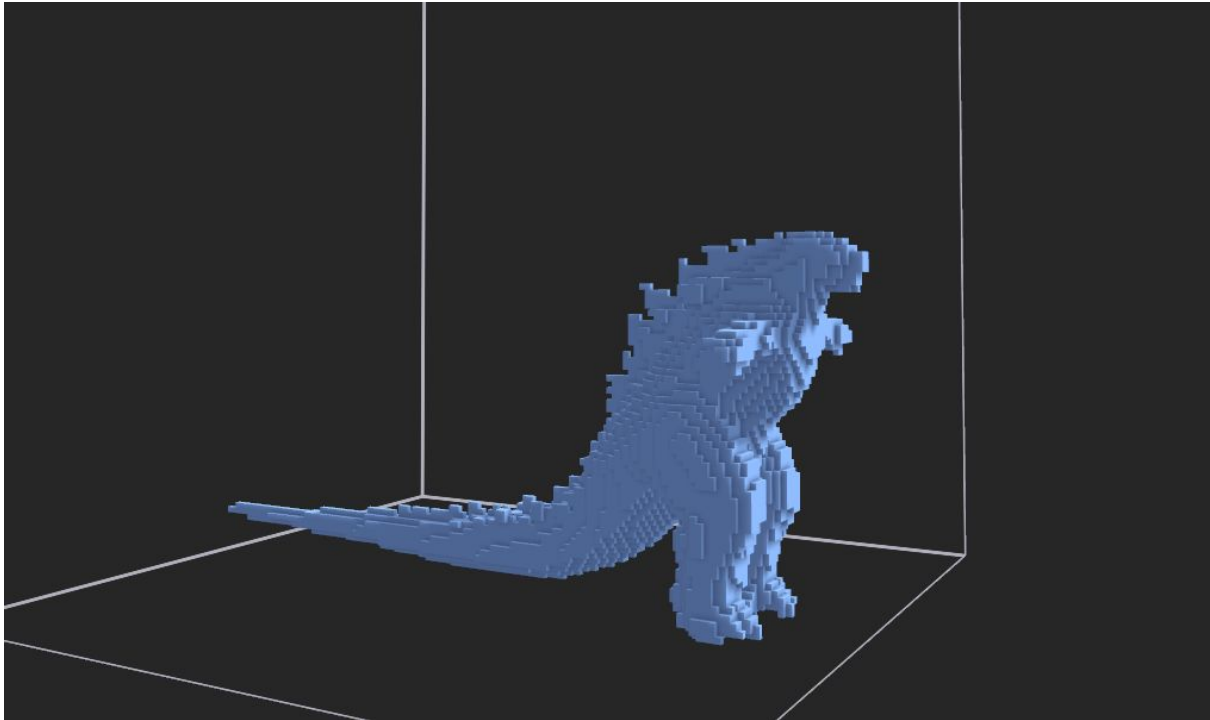
The playthrough will be constrained by time since the presence of the creature will alert more and more security over time. Security will try to disable the monster on his rampage. While in the low-level phase of the game, the player only is limited by a true time counter, the high target aims for a system, in which damage to the character causes the beast to shrink again. With more and more enemies approaching overtime, the player needs to be ahead with destruction, so that he is able to reach the height necessary for destroying the objective.

To destroy objects, the game provides a set of attacks and movement abilities that are triggered by pressing keys in the right order and timing. We want to aim for a game flow that feels dynamic. We want the player to be able to destroy objects in proximity in destructive combos. To reward this, the amount of growth while maintaining a combo is going to be greater than destroying single objects. Additional points are also scored to create an incentive to replay the game, optimize the previous performance and compare his/her self to other players. The goal is going to be a fun destructive flow here.

1.1.4 About the background

From a historical perspective, the first Godzilla movie (1954) is often interpreted as allegory of the trauma caused by the loss of WW2 and the nuclear strike against Hiroshima and Nagasaki. It is necessary to say that our game does not claim to represent anything meaningful besides a playful take on the destruction of a voxel cartoonish city. We had some considerations to weave a simple story about a little Kaijū child that has been separated from his mother, but we decided to discard that idea for now and to focus on gameplay experience.

1.1.5 Concept Art and sketches



This voxelized godzilla model could be a first take on how the playable character might look like. We are planning on creating all 3D models on our own in a 3D voxel modelling software called MagicaVoxel. It is necessary to set a specific scale for objects that can easily be translated into Unity units. For the color scheme we will define a certain color palette that all artists can work with in order to share the common tone.

After final stages of growth, the monster is going to outsize the skyscrapers. With the growth there will be a unique camera shift that the player might not notice at first. While at the beginning the camera is tilted upwards, the player will feel small amongst all of the tall buildings. When growing, the camera will tilt downwards more and more until the player has the feeling of looking down onto the miniature city. This change of perspective should emphasize the immersion of the player actually being part of the monster he is controlling.





This is how we envision our level: A bright voxel city that awaits its destruction. For simplicity of map generation, the city will be split into prefab modules that all look different from each other. The modules will be squares of housing blocks that can be put next to each other and combined into a new city with totally different layout. For the low target, cities will have a predefined layout enforced by roads. However, our desired target is to implement an algorithm that generates the complete city including the infrastructure network from scratch.



Cool voxel destruction effect visualized by Daniel Hoyos. Our current method of destroying objects will be a separation of the model into multiple smaller model pieces that can fall apart. These can easily be worked with and with additional voxel particle effects the satisfaction of destruction will come fast.

1.2. Technical Achievement

1.2.1. Core technical feature

To add a bit of a technical challenge to an otherwise rather design-focused game we will procedurally generate the level, consisting of a city full of destroyable objects. This adds additional replay value to the game since the layout of the map will change with every new round. Additionally, we can increase or decrease the size of the city a lot easier, as we do not have to redesign the map every time. It also takes away some of the design work, since it is not necessary to hand-place every object onto the map.

Procedural generation is also easily scalable depending on the available time and the progress of the project. In the beginning, we might only stitch the map together from a few predefined chunks, made of a piece of street, some buildings, and a few smaller hand-placed destroyable objects. Later we might also procedurally generate the contents of each chunk, allowing for a greater variety and overall more content for the game. Finally, we might even randomly assign certain parameters to each object, such as color, size or rotation.

It is important to note that, while we want to implement this procedural map generation, it is not the main focus of our project. Gameplay and getting the player in the flow are more important for the final game.

1.2.2. Challenges

Of course all of this comes with several technical challenges, otherwise it wouldn't make much sense to include this as a core technical feature in this project.

First of all, there is the usual problem with procedural generation: humans are very good at recognizing patterns. This means that even though we might be able to generate millions of different objects solely by randomly assigning a handful of parameters, most players will pretty quickly notice the patterns behind it. This will make the game feel more repetitive. The best solution for that is to use many different generation techniques and combine them with hand-designed content.

Another problem that it is difficult to find bugs in procedurally generated maps since many of them might only happen in a few rare instances. This can only be solved by doing more tests. Testing can, however, be made easier by using a different seeded random generator for each procedurally generated feature.

And lastly probably the greatest challenge of this project: it will be difficult to get the player in the flow, which is our main goal. It's already difficult to achieve when hand-designing the level, as all destroyable objects have to be placed conveniently for the player to reach, neither too far apart, since this would break too many combos, nor so close that it is too easy to get points. This is made a lot harder when the computer places the objects since we do not have the same amount of control. This problem can only be solved by putting a lot of time and effort into the fine-tuning of the procedural generation.

1.3. "Big Idea" Bullseye



The main goal of our game is to create a fun experience for the player. Through an intuitive character controller and a well designed combo system, the player should be able to experience flow.

To allow replayability on a high level, the destroyable cities are going to be procedurally generated, so that each playthrough is different from the last. Generating authentic and completely destroyable cities will be our main technical achievement for the game.

1.4. Development Schedule

1.4.1. Plan in Layers

- Functional Minimum:
 - Basic character movement, attack and camera control
 - Input with controller
 - Attack timing and chain timing
 - Standard voxel plane as city module foundation
 - First destructible environment objects

- Timer and Highscore as game goals
- => First basic minimal level against the time
- Low Target:
 - Advanced set of attacks
 - Inclusion of combos and chaining of attacks with the right timing
 - Walk and basic attack animations for the character
 - Multiple destructible environment objects such as houses and streets to form a residential area
 - Destruction refinement
 - Simple user interface and HUD for in-game operations and information display
 - Task based goals (e.g. destroy object X)
 - Map generation: Fill given layout with prefab city modules
 - First in-game sound effects for hits & environment
 - => First city-like level, destroy sensibly and in flow
- Desirable Target:
 - Character growth and new attack mechanics for larger character
 - Camera adjustment to support feeling of growth
 - Combo refinement and additional possibilities for large character
 - Model additions and refinement for the environment, make the city lively and good looking
 - Static enemy models like cars in the streets, enemy mechanics
 - Map generation: Generate whole maps from scratch with prefab city modules
 - Voxel particle effects for environment & character actions
 - Addition of background music and advanced sound effects
 - Fancy in-game UI and HUD, menus
 - Flow coordination for a bigger city
 - Small control tutorial
 - => Full city level with destruction objective after certain growth, playable game rounds
- High Target:
 - Advanced game tutorial for step by step introduction to game mechanics
 - Addition of Power Ups and "Time Ups"
 - Multiple levels (either different difficulty level during generation or premade levels without generation, e.g. for the tutorial)
 - Enemies with custom behavior, attacking the character
 - Dynamic object reactions
 - Small story with introductory narration and images
 - Map generation: Generate new prefab city modules
 - Increased game complexity, e.g. destruction gains health
 - Additional game modes (Time trial, objective)
- Extras:
 - Inhabitants for the city as visual enhancement
 - 2D map of all explored areas
 - Character skins
 - Enemy Kaiju

- Statistics screen
- Port to Iphone/Android

1.4.2. Task Timeline

PHASE	DEV	DETAILS	October		November	
PROJECT WEEK:			23	30	6	13
1	Game Design	Everyone - Deliverables	Project Setup	Report Presentation	Critiques	Report Presentation Mutual Critiques
		Everyone - Components	Game Concept Game Idea Game Design Technical Achievements Development Schedule	Paper Prototype Design Paper Prototype Building	General Code Structure Design Controls Design General Attack Design Kombo Attack Design Simple UI Design	Refining Paper Prototype Checking setup for presentation
		Everyone - Playtest		Playtesting Paper Prototype		
		Evgenija - General Design	Draw Concept Arts		Define Color Scheme Test building / object scale Gozilla Model Design	Obstacle Model Design Gozilla Walk Animation
2	Modelling / Animation	Evgenija - Gozilla Model			Gozilla Basic Model	Gozilla Rigging
		Evgenija/Maxi/Tim - Environment			Floor Plane	Basic Obstacles
		Tim - Special Effects				
		Evgenija - Enemies				
3	Scripting	Lukas - Gameplay				Attack Timing
		Tim - Destruction		Basic Destruction Tests	Basic Destruction	Basic Destruction
		Tim - Level Generation				
		Lukas - Camera & Character			Basic Controls Implementation	Basic Controls Implementation
		Maxi - Game Systems			General Scene Structure General Object Structure Timer & Highscore	General Scene Structure General Object Structure Timer & Highscore
Everyone - Bugfixing & Polishing						
4	UI	Everyone - Menus				
		Everyone - In-Game				Only necessary In-Game UI
		Everyone - HUD				Only necessary HUD
5	Sound	Maxi - Effects				
		Maxi - Soundtrack				
					Functional Minimum	
			Game Idea Milestone		Prototype Milestone	

Q4									
Interim Demo		December						Alpha Release	
20	27	4	11	18	25	1	8		
		Report Presentation					Report Presentation		
		Refined Attack Design Advanced UI Design Enemy AI Design			High Target: Pickup / Powerup Design Level Design Tutorial Design Dynamic Enemy Design	Pickup / Powerup Design Level Design Tutorial Design			
Playtesting FM			Playtesting LT			Intermediate Playtesting			
House & Environment Design	House & Environment Design	Enemy Visual Design	Gozilla Scale Design + tests			HT: Story Design	HT: Game experience design		
Gozilla Walk Animation Gozilla Attack Animation	Gozilla Run Animation Gozilla Attack Animation	More Animations	Gozilla Advanced Kombo Animations	Gozilla Advanced Kombo Animations			Polishing BUFFER		
First House Street Planes	First Environment	First full Puzzle Piece	More Houses & City Pieces	More Environment	More life for the city	First full city	Polishing		
					Voxel Particle Effects	Voxel Particle Effects	Polishing		
			Static enemy modelling	Static Enemy Modelling	Dynamic Enemy Modelling	Dynamic Enemy Modelling	Polishing		
Combo Mechanics	Combo Mechanics	Combo Mechanics	Gozilla Scale Mechanics Enemy Mechanics	Enemy Mechanics	Enemy Mechanics	HT: Enemy Mechanics	Polishing BUFFER		
Destruction Refinement	Destruction Refinement	House & Environment Destruction	Fine Tune Destruction		Application of Particle Effects in Game	Application of Particle Effects in Game	Destruction Recap for the whole city		
Map Generation: Fill blanks	Map Generation: Fill blanks		Map Generation: whole map	Map Generation: whole map	HT: new square generation	HT: new square generation			
Combo Controls	Combo Controls	Flow & Timing	Camera Scale Mechanics	Advanced Combo Controls	Flow & Timing		Polishing		
Advanced Point & Highscore Mechanic	Task Mechanics	Task Mechanics	Level Finish Condition: Object Destruction	Level Finish Condition: Object Destruction	Game Flow: from Menu to game and back	Main Menu	Polishing		
Adjust Gameplay FM Buo Fixing			Adjust Gameplay LT Buo Fixing			DT Bug Fixing	DT Bug Fixing		
			Game Menu UI	Game Menu UI	Game Menu UI	General Polishing Simple Tutorial Fancy In-Game UI	Simple Tutorial		
	Simple In-Game UI	Simple In-Game UI			Fancy In-Game UI	Fancy In-Game UI	Fancy In-Game UI		
	Simple HUD	Simple HUD			Fancy HUD	UI Polishing	Polishing		
	First Effects	First Effects		Multiple Effects	Multiple Effects	Ambience	Ambience		
				Soundtrack	Soundtrack	Soundtrack	Polishing		
Low Target			Desired Target						
Interim Demo				Alpha Release					

1.4.3. Task List

PROJECT DETAILS								HOURS	
STATUS	PRIORITY	START DATE	END DATE	DURATION	TASK NAME	ASSIGNEE	DESCRIPTION	ESTIMATED HOURS	ACTUAL HOURS
Game Idea Milestone								62	0
Complete	Functional Minimum	23.10.2019	29.10.2019	6	Tools Setup	Lukas	Setup and build all tools needed for the project	3	0
Complete	Functional Minimum	23.10.2019	29.10.2019	6	Project Setup	Lukas	Set everything up to start the project	3	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5	Report	Everyone	Write the Report for the milestone and upload to the wiki	10	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	3	0
Complete	Functional Minimum	23.10.2019	29.10.2019	6	Game Concept	Everyone	Crate the basic game concept	5	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5	Draw Concept Arts	Everyone	Draw concepts arts and sketches of the basic mechanics in the game	3	0
Complete	Functional Minimum	23.10.2019	29.10.2019	6	Game Idea	Everyone	Define the basic game idea	5	0
Complete	Functional Minimum	23.10.2019	29.10.2019	6	Game Design	Everyone	Define the game in more detail	5	0
Complete	Functional Minimum	22.04.2019	28.04.2019	6	Technical Achievements	Everyone	Define the technical achievements of the game	1	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5	Paper Prototype Building	Everyone	Build the Paper Prototype and test play it	5	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5	Paper Prototype Design	Everyone	Design the Paper Prototype	3	0
Complete	Functional Minimum	23.10.2019	29.10.2019	6	Basic Destruction Tests	Tim	Implement some basic destruction with objects from MagicaVoxel	5	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5	Study MagicaVoxel Development Schedule	Everyone	Download MagicaVoxel and get familiar with it	4	0
Complete	Functional Minimum	30.10.2019	05.11.2019	5		Maxi	Define the tasks and development schedule for the game	10	0
Prototype Milestone								110	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Critiques	Everyone	Written critiques of all other projects as an email to the supervisors	3	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	8	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	5	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Mutual Critiques	Everyone	Submit mutual critiques on the wiki (every team member separately)	1	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Controls Design	Everyone	Design the control scheme of the game	1	0
Not Yet Started	Functional Minimum	06.11.2019	19.11.2019	13	Controls Implementation	Lukas	Implement the control scheme of the game	10	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Refining Paper Prototype	Everyone	Refine the paper prototype and finalize for presentation	5	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	General Code Structure Design	Everyone	Define the structure that every programmer has to stick to	5	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	General Attack Design	Everyone	Define basic and advanced attacks by the character	2	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Kombo Attack Design	Everyone	Define the basic and advanced combos of the character	2	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Simple UI Design	Everyone	Mockup a simple UI and define all necessary parts, draw concept sketches	3	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Define Color Scheme	Evgenija	Set the final color scheme for character and environment, make MagicaVoxel palette	2	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Design Gozilla	Evgenija	Design the main character Gozilla	1	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Test object scale	Tim	Test the scale of small vs big objects in MagicaVoxel and define a base scale	2	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Gozilla Basic Model	Evgenija	Build the first iteration of the Gozilla character with MagicaVoxel	10	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Floor Basic Model	Tim	Build a floor of 1 voxel height to put houses and objects on top	1	0
Not Yet Started	Functional Minimum	06.11.2019	19.11.2019	13	Basic Destruction	Tim	Research more about destruction and implement the destruction interface	10	0
Not Yet Started	Functional Minimum	06.11.2019	19.11.2019	13	Basic Controls	Lukas	Implement the main controls of the character	10	0
Not Yet Started	Functional Minimum	06.11.2019	12.11.2019	6	Timer & Highscore	Maxi	Add simple win conditions/limitations	2	0
Not Yet Started	Functional Minimum	06.11.2019	19.11.2019	13	General Structure	Maxi	Define and implement a general game loop structure with necessary object structures	8	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Design basic obstacles	Evgenija	Design basic obstacles for this milestone	1	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Gozilla Walk Animation Design	Evgenija	Design the walk animation of Gozilla	1	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Gozilla Walk Animation	Evgenija	Rig Gozilla	5	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Basic Obstacles	Tim/Maxi	Build the basic obstacles designed by Evgenija	5	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	Attack Timing	Lukas	Build core mechanics for attack timing	5	0
Not Yet Started	Functional Minimum	13.11.2019	19.11.2019	6	In-Game UI	Maxi	Add simple necessary UI and HUD for the game	2	0

Interim Demo Milestone										147	0
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	8	0		
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	5	0		
Not Yet Started	Functional Minimum	20.11.2019	26.11.2019	6	Playtesting FM	Everyone	Playtest the functional minimum game from the previous milestone	2	0		
Not Yet Started	Functional Minimum	20.11.2019	26.11.2019	6	Adjust Gameplay	Tim, Lukas	Adjust the functional minimum gameplay based on the findings in the playtest	6	0		
Not Yet Started	Functional Minimum	20.11.2019	26.11.2019	6	FM Bug Fixing	Everyone	Fix any bugs found during the playtest and polish the gameplay	10	0		
Not Yet Started	Low Target	20.11.2019	03.12.2019	13	House & Environment Design	Evgenija	Design how the environment and houses should look like and prepare a paper	3	0		
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	Enemy Visual Design	Evgenija	Design the visuals of static enemies	2	0		
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	Advanced Attack Design	Everyone	Design advanced attacks that might have to be altered after FM	2	0		
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	Refine UI Design	Everyone	Design a refined UI after testing the FM design	1	0		
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	Enemy AI Design	Everyone	Design the static Enemy AI	1	0		
Not Yet Started	Low Target	20.11.2019	10.12.2019	20	Gozilla Animations	Evgenija	Gozilla Walk/Run/Attack Animations	20	0		
Not Yet Started	Low Target	20.11.2019	03.12.2019	13	House/Street Models	Everyone	Model houses, streets according to Evgenijas design	20	0		
Not Yet Started	Low Target	20.11.2019	03.12.2019	13	Environment Models	Everyone	Model the environment in Magic/Voxel	10	0		
Not Yet Started	Low Target	04.12.2019	10.12.2019	6	City tile prefab	Everyone	Build a first city tile piece with the models and make a prefab	3	0		
Not Yet Started	Low Target	20.11.2019	10.12.2019	20	Destruction Refinement	Tim	Refine destructions and adjust to bigger scale (houses)	6	0		
Not Yet Started	Low Target	20.11.2019	10.12.2019	20	Combo Mechanics	Lukas	Implement the combo mechanics for the character attacks	10	0		
Not Yet Started	Low Target	20.11.2019	10.12.2019	20	Combo Controls	Lukas	Fit the attacks to the combo mechanics and check for flow & timing	5	0		
Not Yet Started	Low Target	20.11.2019	03.12.2019	13	Map Generation 1	Tim	Build simple layout and fill blocks with prefab tiles	10	0		
Not Yet Started	Low Target	20.11.2019	26.11.2019	6	Advanced Points & Highscore	Maxi	Adjust Points and Highscore according to bigger buildings	2	0		
Not Yet Started	Low Target	27.11.2019	10.12.2019	13	Task Mechanics	Maxi	Implement a task mechanic system	10	0		
Not Yet Started	Low Target	27.11.2019	10.12.2019	13	Simple UI / HUD	Maxi	Adjust the in-game UI and HUD, improve to advanced state	6	0		
Not Yet Started	Low Target	27.11.2019	10.12.2019	13	First Soundtracks	Maxi	Create and add first effect samples for the game	5	0		
Alpha Release Milestone										222	0
Not Yet Started	Functional Minimum	08.01.2020	14.01.2020	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	8	0		
Not Yet Started	Functional Minimum	08.01.2020	14.01.2020	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	5	0		
Not Yet Started	Low Target	11.12.2019	17.12.2019	6	Playtesting LT	Everyone	Playtest the low target game from the previous milestone	5	0		
Not Yet Started	Low Target	11.12.2019	17.12.2019	6	Adjust Gameplay	Lukas, Tim	Adjust the gameplay based on the findings in the playtest	6	0		
Not Yet Started	Low Target	11.12.2019	17.12.2019	6	LT Bug Fixing	Everyone	Fix bugs encountered during the playtesting	10	0		
Not Yet Started	Desirable Target	11.12.2019	17.12.2019	6	Gozilla Scale tests	Evgenija, Tim	Test scaling Gozilla to a bigger version, incorporate destruction	3	0		
Not Yet Started	Desirable Target	11.12.2019	24.12.2019	13	Advanced Animations	Evgenija	Create animations for advanced attacks and combos	15	0		
Not Yet Started	Desirable Target	11.12.2019	31.12.2019	20	Advanced Asset Creation	Everyone	Create more houses, environment, city pieces	10	0		
Not Yet Started	Desirable Target	01.01.2020	07.01.2020	6	Full city	Everyone	Build a first city from prefab tiles	1	0		
Not Yet Started	Desirable Target	11.12.2019	24.12.2019	13	Static Enemy Models	Evgenija	Model the static enemies like cars etc.	10	0		
Not Yet Started	High Target	11.12.2019	24.12.2019	13	Dynamic Enemy Models	Evgenija	Model the dynamic enemies like soldiers etc.	X	0		
Not Yet Started	Desirable Target	11.12.2019	17.12.2019	6	Gozilla Scale Mechanics	Lukas	Implement the Gozilla scale mechanics and increased damage etc.	5	0		
Not Yet Started	Desirable Target	11.12.2019	31.12.2019	20	Enemy Mechanics	Lukas	Implement static enemy mechanics & interaction with the character	12	0		
Not Yet Started	High Target	01.01.2020	07.01.2020	6	Dynamic Enemy Mechanics	Lukas	Implement dynamic enemy mechanics	X	0		
Not Yet Started	Desirable Target	11.12.2019	17.12.2019	6	Camera scale mechanics	Lukas	Make the camera move up with the scale of Gozilla	3	0		
Not Yet Started	Desirable Target	11.12.2019	17.12.2019	6	Advanced Combo Controls	Lukas	Implement advanced combo controls + finetung flow	8	0		
Not Yet Started	Desirable Target	11.12.2019	17.12.2019	6	Finetune destruction	Tim	Fine tune the destruction for big scale Gozilla	3	0		
Not Yet Started	Desirable Target	07.01.2020	14.01.2020	7	Destruction Recap	Tim	Recap the destruction for the now first final city	3	0		
Not Yet Started	Desirable Target	11.12.2019	24.12.2019	13	Map Generation 2	Tim	Implement map generator for creating a whole city from scratch	10	0		
Not Yet Started	High Target	25.12.2019	07.01.2020	12	Map Generation 3	Tim	Implement map generator that creates new city modules	X	0		
Not Yet Started	Desirable Target	11.12.2019	14.01.2020	33	Fancy In-Game UI	Maxi, Everyone	Implement fancy in-game UI, for better usability of our game	10	0		
Not Yet Started	Desirable Target	25.12.2019	07.01.2020	12	Visual Effects	Tim	Voxel Particle Effects for the environment and the character	8	0		
Not Yet Started	Desirable Target	25.12.2019	07.01.2020	12	Effect Inclusion	Tim	Include Voxel Particle Effects into the game	2	0		
Not Yet Started	Desirable Target	25.12.2019	07.01.2020	12	DT End Condition	Maxi	Implement the DT end condition: main objective destroyed and include in game cycle	8	0		
Not Yet Started	Desirable Target	11.12.2019	24.12.2019	13	Game flow	Maxi	Implement a full game cycle from menu to game and back with fancy visuals	10	0		
Not Yet Started	Desirable Target	18.12.2019	14.01.2020	26	Soundtrack	Maxi	Add more soundtracks to the game	10	0		
Not Yet Started	Desirable Target	18.12.2019	14.01.2020	26	Sound Effects	Maxi	Add different soundeffects + ambience to the game	10	0		
Not Yet Started	Desirable Target	01.01.2020	07.01.2020	6	Playtesting DT	Everyone	Playtest the almost final version of the desirable target game	5	0		
Not Yet Started	Desirable Target	01.01.2020	14.01.2020	13	Add Simple Tutorial	Everyone	Add a simple tutorial for the upcoming playtests	10	0		
Not Yet Started	Desirable Target	07.01.2020	14.01.2020	7	Bugfixing & Adjusting	Everyone	Adjust the gameplay based on the findings in the playtest, bugfix	12	0		
Not Yet Started	High Target	25.12.2019	07.01.2020	12	Powerup Design	Everyone	Design Power ups for the character	X	0		
Not Yet Started	High Target	25.12.2019	07.01.2020	12	Dynamic Enemy Design	Everyone	Design dynamic enemies	X	0		
Not Yet Started	High Target	25.12.2019	07.01.2020	12	Advanced Tutorial Design	Everyone	Design an advanced Tutorial	X	0		
Not Yet Started	High Target	25.12.2019	07.01.2020	12	Level Design	Everyone	Design new playable levels e.g. for the tutorial	X	0		
Not Yet Started	High Target	01.01.2020	14.01.2020	13	Story Design	Everyone	Design a story for the game and increase game experience	X	0		
Not Yet Started	Desirable Target	07.01.2020	14.01.2020	7	Polishing	Everyone	Polish the game desired target	20	0		

Playtesting Milestone							130	0	
Not Yet Started	Functional Minimum	22.01.2020	28.01.2020	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	8	0
Not Yet Started	Functional Minimum	22.01.2020	28.01.2020	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	5	0
Not Yet Started	Desirable Target	15.01.2020	21.01.2020	6	Playtesting DT	Everyone	Playtest the desirable target game	6	0
Not Yet Started	Desirable Target	15.01.2020	21.01.2020	6	Adjust Gameplay	Tim, Lukas	Adjust the gameplay based on the playtest	10	0
Not Yet Started	Desirable Target	15.01.2020	21.01.2020	6	DT Bug Fixing	Everyone	Fix any bugs encountered during playtesting	10	0
Not Yet Started	Desirable Target	15.01.2020	21.01.2020	6	Gameplay Polishing	Tim, Lukas	Polish the gameplay for the playtest session	5	0
Not Yet Started	Functional Minimum	15.01.2020	21.01.2020	6	Create Questionary	Everyone	Create the questionary for the the playtest session	2	0
Not Yet Started	Functional Minimum	15.01.2020	21.01.2020	6	Playtest Scenario	Everyone	Define the playtest session and what the participants are supposed to do	2	0
Not Yet Started	Functional Minimum	22.01.2020	28.01.2020	6	Playtest Session	Everyone	Playtest the game with external players	20	0
Not Yet Started	Desirable Target	15.01.2020	28.01.2020	13	Visual Polishing	Everyone	Final Gozilla, asset and effect visual polishing	10	0
Not Yet Started	Desirable Target	15.01.2020	28.01.2020	13	Sound Polishing	Maxi	Polish the sound of the game	5	0
Not Yet Started	Desirable Target	15.01.2020	28.01.2020	13	UI Polishing	Everyone	Polish the UI of the game	5	0
Not Yet Started	High Target	15.01.2020	28.01.2020	13	Advanced Tutorial	Everyone	Create an advanced and more detailed tutorial	X	0
Not Yet Started	High Target	15.01.2020	28.01.2020	13	HT models	Everyone	Create high target models (environment, power ups etc.)	X	0
Not Yet Started	High Target	15.01.2020	28.01.2020	13	Gameplay Additions	Lukas, Tim	Add designed high target gameplay additions to the game	X	0
Not Yet Started	High Target	15.01.2020	28.01.2020	13	New Game Modes	Maxi	Add new game modes	X	0
Not Yet Started	High Target	15.01.2020	28.01.2020	13	New Combos	Lukas	Add high target combos to the controls	X	0
Not Yet Started	Desirable Target	22.01.2020	28.01.2020	6	Apply Feedback to Game	Everyone	Apply the feedback from the playtest to the game and fix any complaints	20	0
Not Yet Started	Desirable Target	22.01.2020	28.01.2020	6	Bug Fixing	Everyone	Fix any bugs encountered during playtesting	10	0
Not Yet Started	Desirable Target	22.01.2020	28.01.2020	6	Polishing	Everyone	Polish any unpolished aspects encountered during playtesting	10	0
Not Yet Started	Functional Minimum	22.01.2020	28.01.2020	6	Feedback Analysis	Everyone	Analyse the feedback from the playtest session	2	0
Final Release Milestone							142	0	
Not Yet Started	Functional Minimum	05.02.2020	11.02.2020	6	Report	Everyone	Write the Report for the milestone and upload to the wiki	15	0
Not Yet Started	Functional Minimum	05.02.2020	11.02.2020	6	Presentation	Everyone	Make the Presentation for the milestone and upload to the wiki	10	0
Not Yet Started	Functional Minimum	29.01.2020	11.02.2020	12	Video	Everyone	Make a video that highlights exciting aspects of the game	15	0
Not Yet Started	Functional Minimum	05.02.2020	11.02.2020	6	Compiled Game	Everyone	Compiled final version of game with sources	2	0
Not Yet Started	Desirable Target	05.02.2020	11.02.2020	6	Bug Fixing	Everyone	Fix all known bugs for final version	40	0
Not Yet Started	Desirable Target	05.02.2020	11.02.2020	6	Polishing	Everyone	Polish everything for final version	40	0
Not Yet Started	Desirable Target	05.02.2020	11.02.2020	6	Playtesting	Everyone	Playtest the whole game to find any rough edges	20	0

1.5. Assessment

A main incentive for players picking up video games are the unreal worlds that they can dive into in which things are possible that would never come true in reality. Our game lets players control a Kaiju and destroy a city which is unimaginable otherwise. We want to give players the power of the Kaiju and let them freely roam inside a cartoon city. The game should be abstract enough to ridicule itself with bright colors and environment designs. The world should depict an imaginary place of fun, a playground for the young and grownups. Different landmarks can show relations to real buildings to increase the feeling of power by giving a comparable example that the player knows.

The voxel graphics intend to keep a childish look and open up the game for all ages of players. The main audience will be casual players who enjoy a quick match with a low time limit. There does not need to be complex story or gameplay elements for casual players, as long as the core concept is fun. Thus, the focus is completely on the flow of wildly running through a miniature city and randomly destroying it. By providing objectives and different levels, a short term game flow is targeted at. We want the players to pick up our game and try another round to beat their former highscore in a level they have already completed. With the combo and timing mechanics in addition we try to include an increasable skill level that players can reach by practicing for many times. The game should quickly be rewarding for beginners but also leave some space to improve oneself.

Overall the design can be called a successful design if after the release players want to play "just one more round". The main factor is the game flow that is tightly connected to the character controls, the combo mechanics and the city design and placement of destructible objects. What is going to be destroyed and how does not really matter, but having fun destroying it is the main criteria for our game. We will try to support the flow and fun feeling

with sound effects, challenging music, nice visual and particle effects for players to enjoy destroying a city.

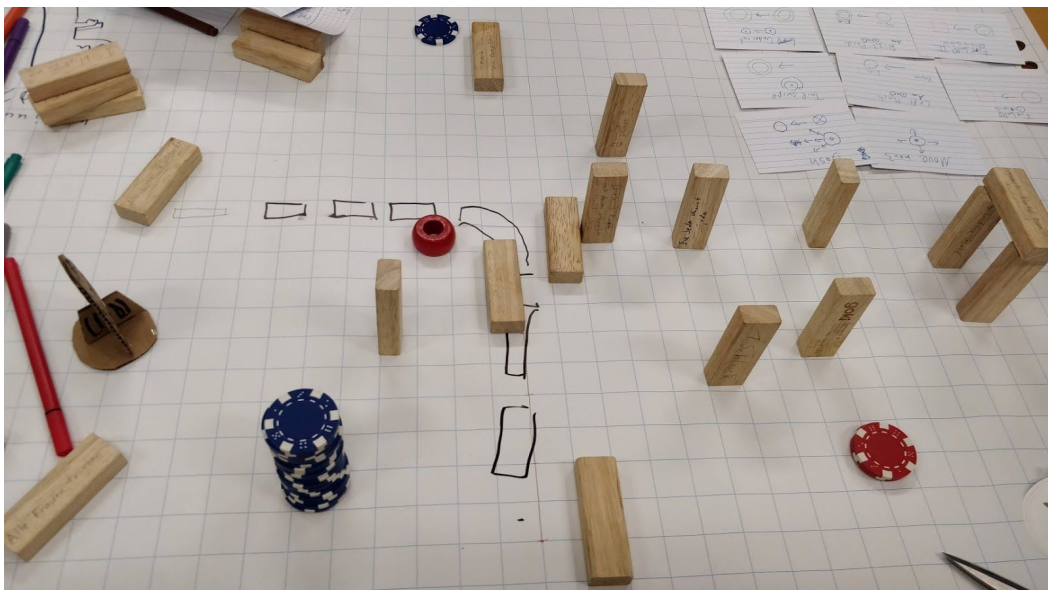
2. Prototype

2.1. Overview

The paper prototype is a method to evaluate certain core features of a game early on. Referring to the Kaiju game, it is based on two core features: the idea of a Godzilla destroying cities that will never look the same due to procedural map generation and a fun and flowing experience while chaining attacks. With our prototype we are trying to incorporate the generation feature by random placement of bricks on a map that are symbolizing houses or small obstacles of the game. The main focus, however, is the playability and the fun we can create by combo attacks. This aspect is our “big idea”, therefore, it is also the central topic of our paper prototype. Without a perfect control system and attack set that would make the player feel an immersive flow, destroying houses could not even be half the fun. Thus, we decided to create a move pool that a player can choose one per round which should resemble the possible attacks later in the game. It is very important to note that the paper prototype was designed as a round based game. Testing distances and timings on paper has to be precise and each move has to be thought about thoroughly. The timer planned to be in the game that would represent a perfect combo timing was replaced by counting the rounds that an attack could be continued. That way we could decide whether the combo system would be engaging, although it felt more like a puzzle game on paper.

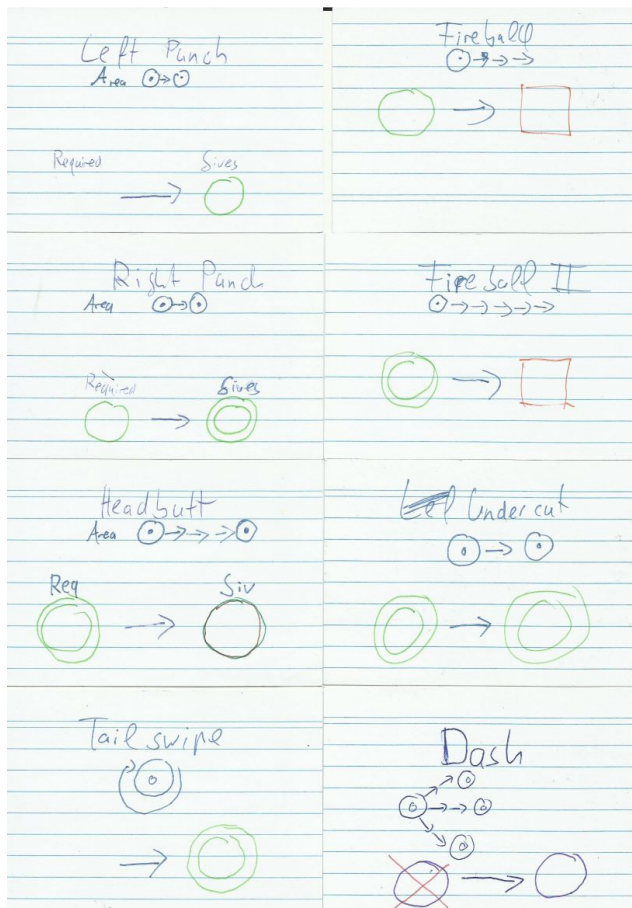
2.2. Gameplay

The Gameplay of our prototype is centered around the playable character, the Kaiju. With the mindset to explore possible player actions, we set up a playing field on a big square sheet:



Next, we set up some wooden blocks, that should represent objects which need to be destroyed, and a plastic dinosaur, representing the playable character. Later the toy was replaced with the red torus for convenience. Whereas our video game is going to have a target objective, for the prototype we decided that we want to test the players short term objective to maintain an attack combo.

An attack combo is considered as a chain of attack actions that we defined beforehand. While the player can move his character freely on the playing field grid, he is required to use one of the actions that form the attack action set:



2.2.1. Attacks

On the left you can see the attack actions that the player can use. Some of them require the player to be in a state depicted by the colored sign. Under the name, a drawing describes the area covered by the attack.

For instance: The Left punch requires nothing and can be performed anytime. It covers one field forward. After it is used by the player it yields the green circle state.

From there the player can choose Fireball, Right Punch, since he is in the green circle state.

The fireball attacks throw projectiles forward, and won't move the character.

Finally we introduced the Dash ability that maintains the combo, moves the player forward 2 tiles forward or diagonal. It's only requirement is that it can't be used repeatedly.

2.2.2. Games Rules

After some experimentation, the paper prototype plays as follows:

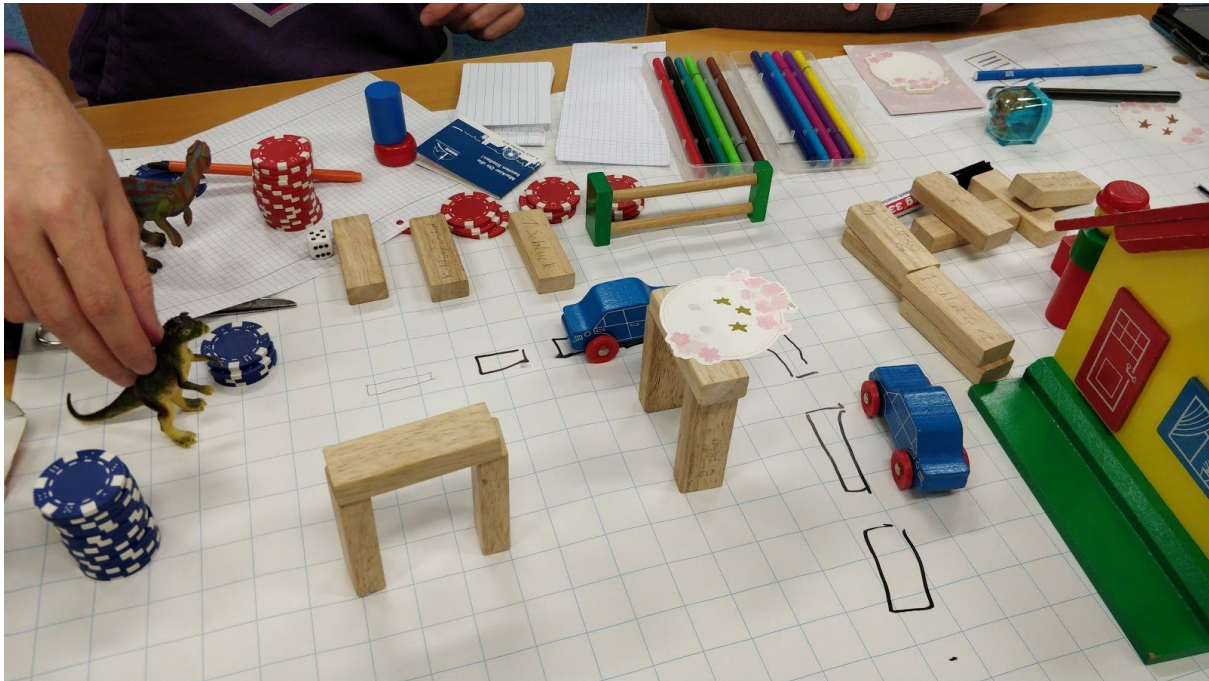
The goal of the game is to score as much points as possible in a given level/scenario. Points are rewarded if objects, represented by wooden blocks are destroyed.

Before the game a scenario is constructed by placing the player figure and the objects to destroy at positions, defined by the grid of the squared playfield.

At each turn the player can choose one action, either he moves three tiles freely, or uses one of the attacks in the attack actions. Objects are destroyed only if the area of an attack actions covers a grid position. For instance, the Left Punch attack covers the field in front of the player character. Each destroyed objects yields one point. If objects are chained in attack combos, they reward additional points: the second destruction rewards 2 points, the third 3 points and so on. In total, a destruction chain of three objects reward 6 points. Therefore, maintaining the combo is important to score the most points.

To maintain an attack combo, the player has to use one of the attacks, if he moves freely the chain breaks. The chain also breaks after 3 turns without any destruction. The scenario can have cars. Cars move 2 tiles forward after each player's action. If they collide with the player's character, they disrupt the attack combo immediately.

2.3. Experiences and Learnings



As we started to explore our newly set rules, we quickly realized how important a versatile and flowing combo system was. Due to the given combo timer and setting of “destructible” blocks it could happen that the combo chain would end before the player managed to finish to hit his last obstacle, which might be not the optimal path for the player.

Therefore, we adjusted the combo system by adding a “gap closing” “Dash”-Ability, which allowed the player to move rapidly forward without breaking the combo chain. Additionally, a “fireball” ability, which allowed the player to destroy far off targets without using any movement, was added.

In addition, we realized that random placement of objects could prevent the player from finding an optimal route through the city. If the placement of objects is too scattered, even a gap-closing ability will not help. The conclusion from this learning is that we still have to care about the alignment of destructibles and program the map generation in a way, that from a level design perspective there is no flow breaker. We will have to care about intended gaps and perfect alignment of fitting obstacles so that the player will always have the possibility to find sections on the map for well flowing combos.



After those adjustments we found the game to be engaging. It was fun to try to carefully map out the path in which the destruction is most efficient and earns more points, leading to a higher highscore. This was the first confirmation that our game idea was actually fun to play.

Beside the importance of the combo system, we also could tell that “enemies” can add to the game. As cars in the prototype are a potential risk of losing a combo streak, the player has to think around them to get to his goal. This adds an additional layer of complexity to the gameplay.

2.4. Revisions to the Game Idea

Before trying out our game idea with this prototype we had multiple open questions, which we tried to answer during these playtests, because we could not decide on them during the idea finding phase.

One of the most important of them is what the goal of the game should actually be. We had several different options to choose from. The goal could be that the player has to destroy a certain amount of objects or reach a highscore in a limited amount time, survive waves of ever more difficult enemies or reach a predefined milestone, such as destroying a building of a certain size. All of them have upsides and downsides. After testing our prototype we decided to start with a time-based approach, since it is easy to implement, and then switch to a milestone oriented playstyle later in the development, as this gives the player a feeling of progress. This approach also means that the character growth mechanic is an important aspect of the game.

The other option we heavily discussed beforehand, and could only decide on after extensive tests, was whether or not to add enemies to the game. We tried adding enemies that follow the player and are able to attack over distance, and realised that we would need a complex system for the player to defend against them. So we decided to focus on destruction of objects, and instead of adding enemies, only to implement dynamic obstacles such as cars. These are able to hurt the player on touch, but won't try to follow the player.

3. Interim

Having concluded this milestone, we are supposed to have arrived at our low target for the game. At first we are going to give an overview of which targets we met and which still need more focus. Here are all low target bullet points:

- Advanced set of attacks MISSING
- Inclusion of combos and chaining of attacks DONE
- Walk and basic attack animations for the character DONE
- Multiple destructible environment objects DONE
- Destruction refinement DONE
- Simple user interface and HUD DONE
- Task based goals DONE
- Map generation: Fill given layout with prefab city modules PARTLY
- First in-game sound effects for hits & environment MISSING
- => First city-like level, destroy sensibly and in flow PARTLY

Addressing the missing or partly declared bullet points, one has to say, that the basic attacks are now implemented and a general structure has been set up. Therefore, future inclusions of further attacks will be available faster and will be added soon.

For map generation we have been trying out different ways that proved to not be efficient enough, thus, we are slightly behind the time plan.

First in-game sound effects were pushed to the next milestone, instead, we already started working on static enemy AI that was planned for the next milestone. As enemies were widely requested by the reviewers, we are intending on putting more focus on this topic.

All in all we can say that good progress has been made with some alterations of the plan and foundations have been laid for the next milestone.

3.1. Feedback review

After the last milestone, we received a lot of positive feedback by the other teams and their members, especially about the idea of growth throughout the game cycle. The difference in between feeling weak as a small character at the beginning while growing and becoming stronger towards the end was received positively and mentioned a couple of times. The perspective change we imagined from below houses to above them clicked with the other developers and all of them could right away imagine a cool scenario. We conclude that this idea is interesting enough to be a main feature of our game and we will leave it unchanged. In addition, one student supported the combo system as a mechanic that adds depth to the gameplay, which is another hint that the character design is fulfilling.

On the contrary side, 80% of the developers criticized the lack of a challenge that only a time constraint is trying to enact. The highly suggested addition of enemies will now be strongly considered by us and put as priority for our high target goals. We are striving to deliver a smooth flowing character in the first place, but making the game more exciting will come right after. Thus, we won't proceed any thoughts about multiplayer as it would blow up the

scope of the project too far.

Concluding it can be said that the challenge of the game in the end will be a combination of time restriction, highscore chasing and enemy gameplay. This should hopefully satisfy all critics.

3.2. Game Development

On a general level, a full game loop has been implemented that makes it possible to start and end a game including leaving the application and reloading the level. A very basic UI informs the player of highscore, timer and tasks that are yet to be fulfilled. The background structure for scene changing and data consistency has also been set.

3.2.1 Level and Task System

The level and task system reads and evaluates .json files which can easily be expanded with the predefined structure. Each level includes multiple tasks that have to be executed in order. As the final task of a level the objective (e.g. high skyscraper) will be set. Each task receives information about destroyed obstacles and checks, whether these are contained in its description. Once all necessary objects are destroyed, the follow-up task will be loaded. This task system is expandable and can also support events like reaching a certain highscore or combo. Moreover, it will be used in the future to include a task based tutorial that introduces the player step by step to the controls.

It's clear advantages are the scalability once the foundations have been implemented. New levels and tasks can easily be included by adding to the .json file and different types of tasks are quickly implemented, too.

3.2.2 Static Enemy AI

Because of the reviews, we decided to start design and implementation of the first basic static enemies, in our case cars, earlier than planned. The car AI has to conform with the map generation code as it is basically traversing the streets that are procedurally generated and it needs to work for whatever layout outcome. After first tests, cars are now able to randomly traverse a street graph with crossings as nodes and edges as streets. Cars will take a turn at crossings and turn around once they hit a dead end. However, there is no car collision yet and cars do not interact with each other or the character. These will be the tasks to address next for the desired target milestone.

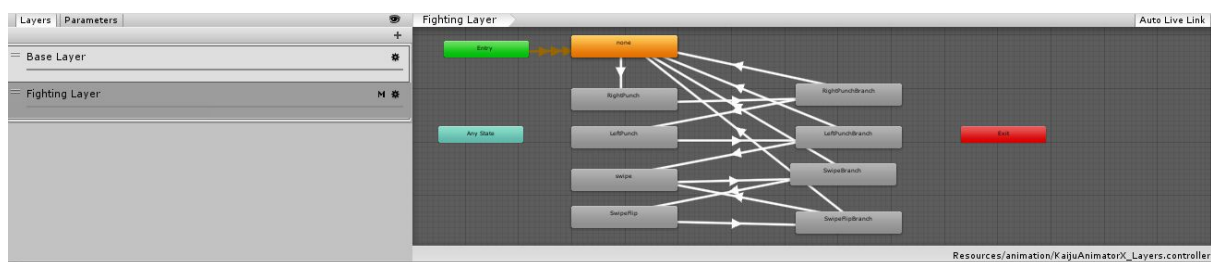
3.3. Character Development

3.3.1 Character : Camera & Actions

The third person controlled character has been implemented by us. The main camera follows the character and can be rotated around its center target freely. Vertical rotations affect the distance to its target, so that the camera is able to move away from the player as it approaches a top down perspective. We're planning to utilize this feature with our narrative of a growing Kaiju, so when he reaches bigger sizes, the camera looks with a downward angle on him. The character can run freely and features two attacks so far. A light attack that

resembles a punch, and a heavy arm swipe attack. The light attack currently deals 25 damage and the heavy one 50. These attacks are mapped to two buttons. The light attack can be triggered anytime, while the heavy attack requires the light attack to be triggered twice before. Triggering an attack will cause the character to play an animation. During this animation, the attack collision with a destructible object causes this object to take damage depending on the type of attack. In summary, the basis controls, a basic attack scheme and damage dealing have been implemented. Even if the fundamentals for our Kaiju have been established, there is a lot of more work to be done. However, to make the game a fun experience, more attacks need to be designed and developed.

To animate the character smoothly and transition between animations turns out to be most challenging, and it takes quite some time to time the interval and effect area of the collision during an attack animation. To make things easier, the character includes a human rig, which has the advantage, that any human rig animation can be mapped onto its rig bones. Therefore, it is possible to use free animations from the internet. Of course, these require fine tuning to work with our Kaiju. For the interim demo, our attacks are animated and the full animation graph is layered to the upper body for attacks and lower body to transition between idle, walking and running.



The character animator: A two layered state machine that plays animations for the upper body and the base.

3.3.2 Chaining Destruction

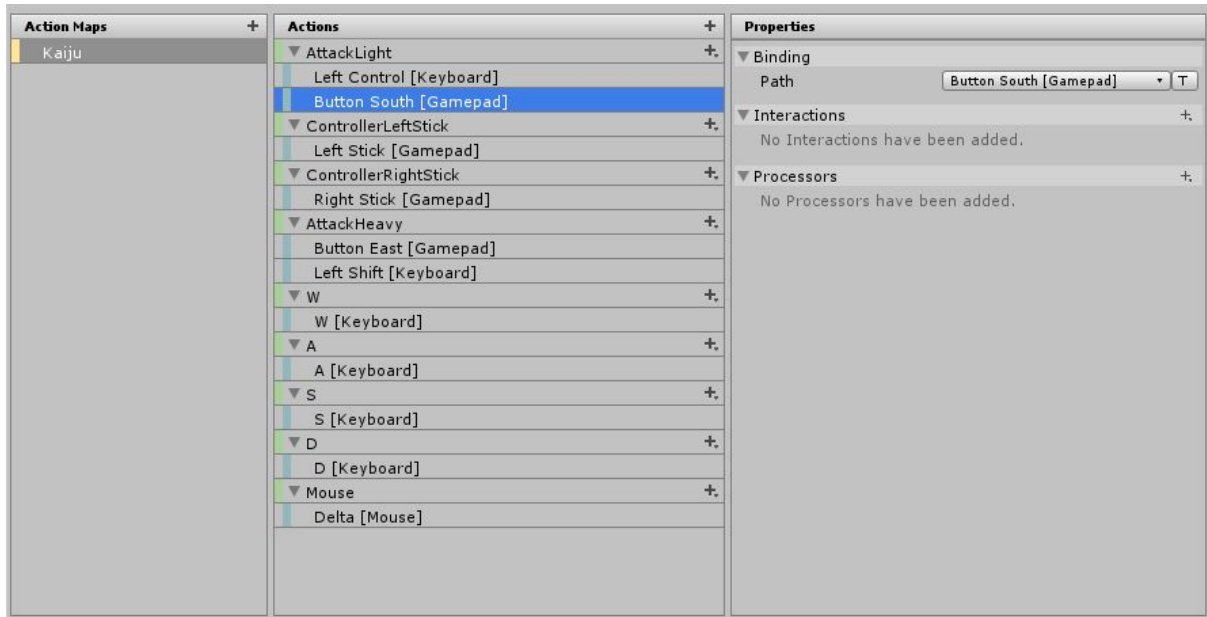
For our game we want to establish flow by creating an incentive for the player to stay in a combo as long as possible. This is essentially maintaining destruction without breaking the action. For now, a basic system is implemented, so that every successful attack refreshes the current combo. We are planning to reward good timing with a increasing multiplier on the total points of a finished combo. The combo itself broadcasts to the game core if it runs out, so that the points can be given properly.

The player will be rewarded with more points, if he's able to destroy more objects in sequence. There are already first considerations for the player on how he destroys the object in scene and which attacks to use.

3.3.3 Input System

For the process of mapping device keys to actions, we are using Unity's new input System that has been introduced this year as an alternative to the default input system. It provides a set of actions that can map to multiple key or axis bindings. For example, moving the character can not only be mapped to an analog stick of a controller, but also the WASD keys of the keyboard. By understanding and implementing the new input system for our project

we gain the advantage, that different controller types can be supported easily and reconfiguration of key bindings should yield low effort.



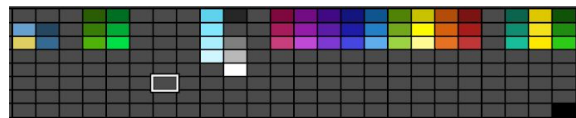
Unity's 2019 Input system allows excellent action -> key or axis binding, universal for any gamepad device or keyboard + mouse configuration

3.4. Design

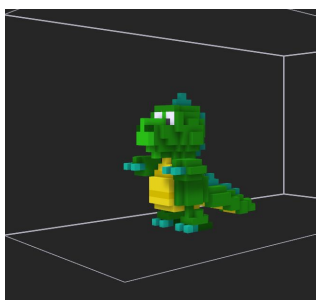
As established in the other chapter, our goal is to create a colourful and fun experience in a 3D voxel environment. The vision was to distance our game look and feel from the expected "Godzilla-game-experience", which typically is more grim and dark. Adjectives that should describe the look and feel of our game should be: Colorful, Sweet, Fun, Different.

3.4.1 Color Scheme

Even though our game was planned to be "colourful", it's still had to have a colour scheme that would fit everything together. Because we already knew that many events and impressions (e.g. rising high-score numbers, particle effects, etc.) will occur during the game, it was important not to overwhelm the player with an overly powerful colour scheme. Therefore, more saturated colours shall be used. Additionally, Godzilla itself has his own primary colours, emphasizing the player character.



3.4.2 Character

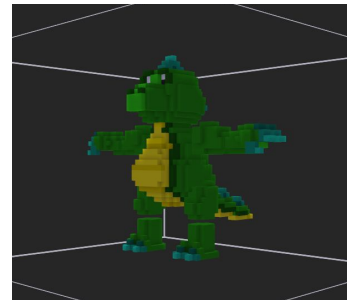


It was challenging to create a mesh that resembled godzilla, but still made him look cute enough to fit into our concept. Choosing brighter colors and making the head and body shape rounder helped in this endeavor.

After finishing the first version of Godzilla and revisiting the design, the design was changed. The initial design was too round and was missing the needed body parts to efficiently destroy a house.

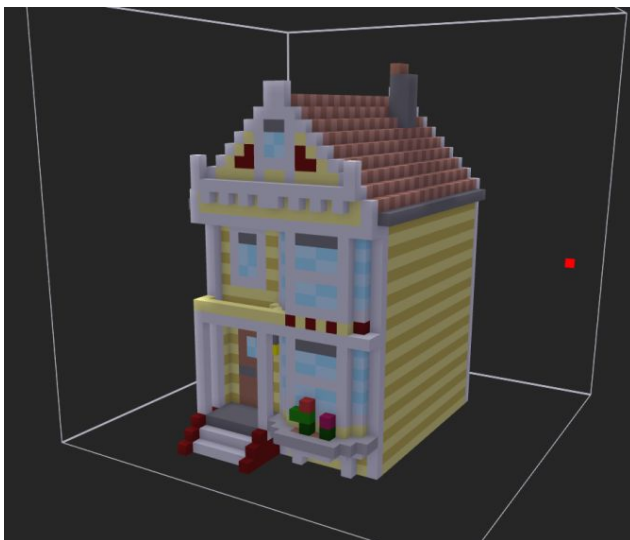
The second version is at the moment the most promising one. Adding limbs and making the figure bigger, helped in persuading a “strong” character. Also, by having longer arms, our Godzilla gained a little bit of reach, so his animation have a greater impact.

As we explored the medium of voxel graphics, we quickly realized that animation is not easy task. As traditional animation methods (using a rig adjusted with bone weight) seemed to fail, due to voxel “stretching” as the body itself moves. Therefore, the godzilla mesh was “sliced” to allow the limbs to glitch into another without the risk of the voxels losing its shape. As we looked into other voxel games (e.g. “Cube World”), we realize that the brief intersection of limbs had little impact on the overall game experience.

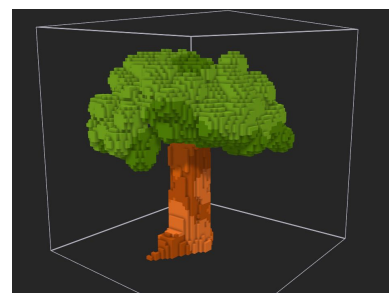
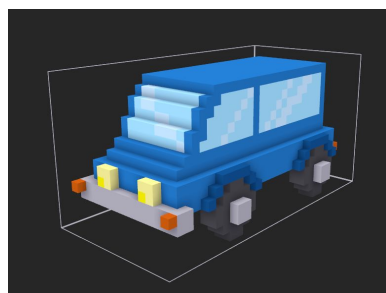
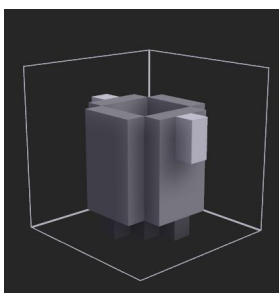


3.4.3 Environment and Obstacles

For the design of the city, we decided to take San Francisco as reference. The overall aesthetic and colorscheme of the city blends well with our initial design idea in mind. (See our version of the “Painted Lady”).



For the first obstacles we decided to go with smaller to medium objects, therefore ranging from a simple trash can to one-family houses. The goal remains to Godzilla being able to destroy skyscrapers and the highest landmark in the city, but we decided to establish an “easier” obstacle course first, before adding bigger assets into the game.



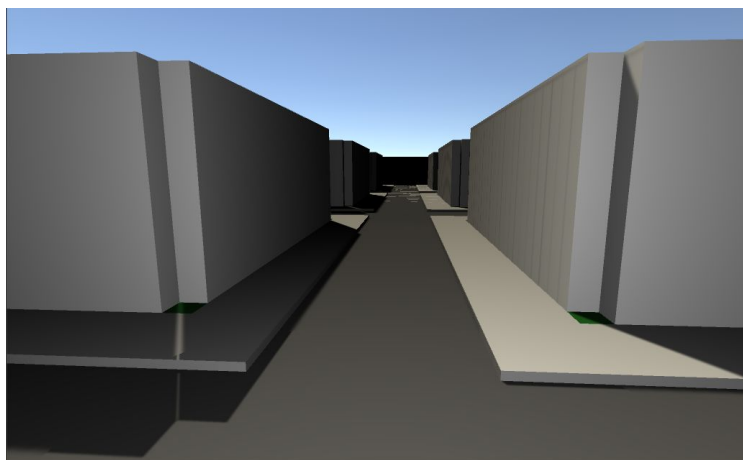
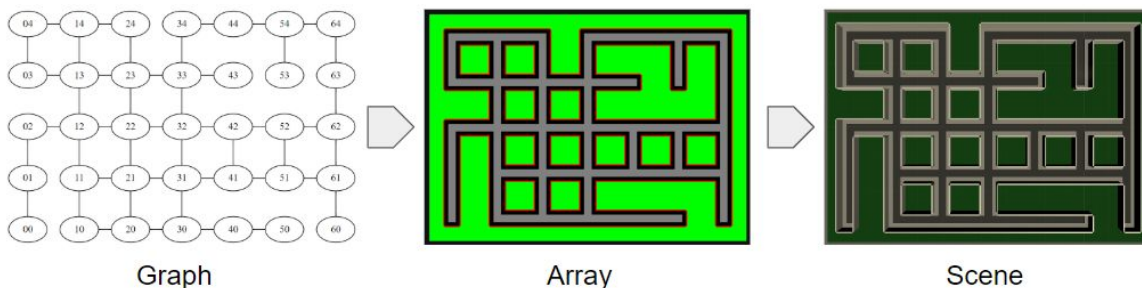
3.5. Environment & Destruction

3.5.1 Procedural Generation

Development on the procedural generation has already started, even though it is not part of the first playable prototype. A street map can already be generated and some facades are being placed. But due to performance concerns it is not ready to be included in the prototype.

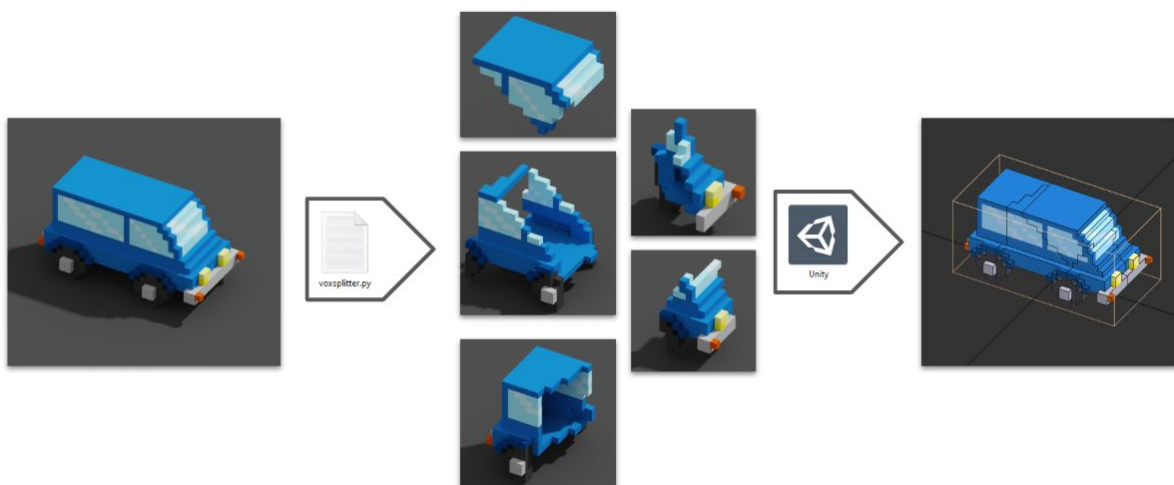
The generation process so far goes through three different steps. The first is to generate a graph in which every node represents an intersection and every edge represents a connection between two intersections. At the beginning the street network represented in the graph is just a grid, but step by step edges are removed without disconnecting the graph to generate a more organic street map. The next step is to translate the graph into a two-dimensional array, where every cell contains roughly one square meter of the final map. this is to later place smaller objects. In this array places for buildings next to streets are marked. The last step is to initialize the map with premade ground and building tiles so the unity engine can display it.

Due to the high amount of tiles initialized the performance dropped drastically. We hope to circumvent this by using significantly bigger tiles for the streets.



3.5.2 Asset Pipeline

We quickly realized that our original plan on how to create the destructible environmental assets was way to time intensive. It required us to not just model the asset, but also to then split them up by hand into smaller pieces, which turned out to be an extremely time consuming process. The splitting of the models into smaller pieces is necessary in order to create the debris parts which are left after the destruction of an object. To solve this problem we developed a python script, that reads files in the .vox format used by MagicaVoxel, our modeling software, and automatically generate a broken up version of the model. This sadly means that we lose some control on how the models are broken up, but gain a lot of time we can utilize on other more important parts of the project.



3.5.3 Destruction

The destruction script that served us as proof of feasibility during the concept phase has since been extended by a health counter that determines the amount of damage, and therefore the number of hits, an object can take before being destroyed. The health of an object also slowly regenerates, creating a threshold of damage that has to be done in a certain amount of time before an object can be destroyed. Destroyed objects are replaced with broken versions of the object, generated through our asset pipeline. Other smaller tweaks and bug fixes were also done.

4. Alpha Release

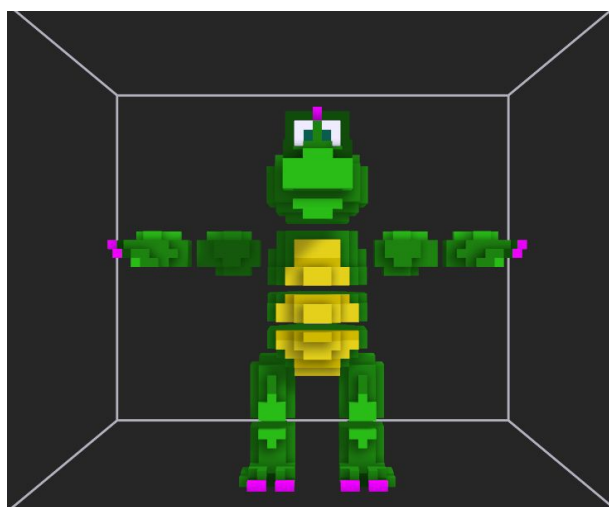
4.1 Design and Changes

During the development process, we could recognize which design ideas were actually useful and which needed to be revisited. Changes and new features are explained in the following paragraphs.

4.1.1 Godzilla

As seen in the interim presentation we already have changed the look of Godzilla once, but through the development of the last few weeks, the design was changed again two times. The design presented at the last milestone still felt very clunky and therefore had to be changed.

Taking the inspiration from puppeteering puppets, the model was segmented into more parts, allowing the rigid voxel model to move more freely. This process was iterated a second time resulting in the actual model.



The animations we used are not self-made but are taken from different sources (like the Unity Asset Store and Maximo Animations). Even though this is the case, the model is hand-rigged using different IK-constraint that would allow a smooth creation of different animations. Those are planned to be created in the last phase, due to the fact that not all animations are yet fitting for the game.

4.1.2 The City

The city consists of three big asset groups: suburb, the outer city, and the inner city.

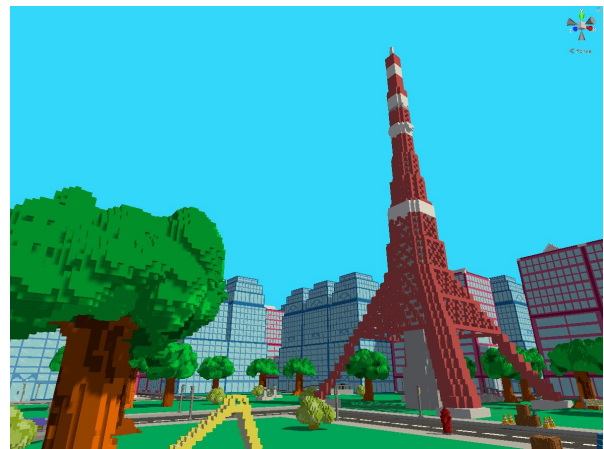
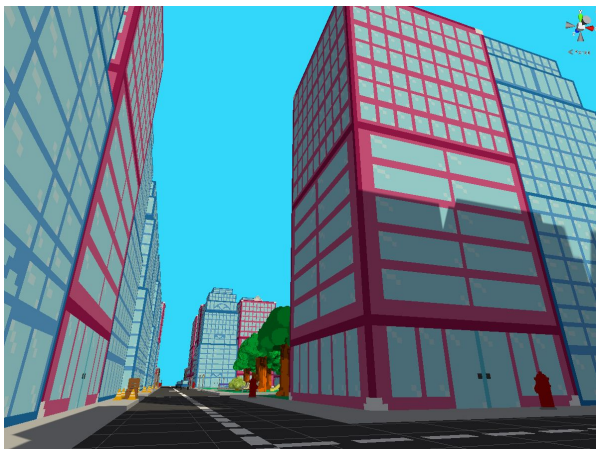
The suburb is the “beginning” area for Godzilla. There the player will be able to find small objects that are easily destroyable and therefore allow a fast growth. Benches, traffic cones, but also small houses are considered as “small” obstacles.





As Godzilla grows he will be able to then destroy obstacles in the inner city. The landscape there consists of “medium” height houses and taller trees. Cars that posed a threat for small Godzilla now become destroyable, but still not completely harmless.

In the last category, we find skyscrapers “protecting” the main goal: the TV-Tower, which resembles the one in Tokyo. Those buildings are not only especially tall, but are also placed pretty too close to each other so that big Godzilla literally has to punch his way through.



The TV-Tower is placed in the middle of the city with only small obstacles around it. Giving the player enough space to fulfill his combos to destroy the goal.

4.1.3 Cars as enemies

A highly requested gameplay feature that is both breathing life into the city and enhancing the challenge of the game is cars. Cars are supposed to fill the streets with self-steering and moving objects that pose obstacles for Godzilla. When hit or run over by a car, Godzilla will lose energy and precious time, because it is immovable for a short amount of time. Cars also serve as small to medium size obstacles that will reward a suitable amount of highscore points for Godzilla to grow after reaching level 2. When skilled, a player might already be able to take cars out while still at smallest size. Besides normal cars there will be patrolling police cars that watch the happenings in the city. In case Godzilla destroys objects while being watched, a manhunt will be started, and police cars will increasingly be aggressive towards Godzilla. The following paragraphs will give an overview of features and difficulties of the car implementation.





When a city is procedurally generated, a street network will be spawned that consists of many intersections, straight roads and Tunnels. Tunnels are spawned wherever there is a dead end of a street. These are then used for cars to spawn throughout the level to keep the number of cars on the road stable and to feed the streets with more police cars, if necessary. The preset number of cars will be calculated to be the size of road map width times height. One after the other cars will be spawned in the tunnels. On the contrary, when a car enters a tunnel it will also be removed from the map. Moreover, each intersection is internally handled as a Node whose neighbors the cars use as orientation points for where to go next.

The cars themselves are powered by Unity Wheel Colliders that allow fine tuned physics to apply in realtime. Each wheel is properly simulated and touches the street while driving, which allows cars to overcome small obstacles, too. Each wheel collider's rotation is mapped to the car wheels and makes for a nice animation. Raycasts straight out the front of the car check, whether another car is driving ahead. This mechanism will slow down the car, if it gets too close to a slower car in front. In case this does not prevent a collision with another car, cars take damage and might even be destroyed. Most of the time navigate from node to node by slowing down before entering a crossing and accelerate again when leaving the crossing. At the moment, cars randomly choose which turn they will make. However, for the future a more sophisticated logic will replace it. I am planning on a function that calculates the traffic density on the map which will then return a decision for the car to head to a less crowded area. This should erase congestion issues that still appear after a while at more frequented crossings.

The most challenging part was coming up with a logic for the crossings. It is necessary for cars to check the other traffic members and their route. This could not be solved by additional raycasts. Therefore, I implemented a manager that the car will register to, once it is shortly before the crossing. The car notifies the manager about its desired turn. Then the manager will run through every queuing car and add up the desired turns by calculating the crossing sections that will be blocked by a car. By using a flag enum the manager can sum up the sections starting from the first queuing car until the last car that still fits on the

```

10
11     /// <summary>
12     /// Enum saving the occupied lanes in 4 bits: 0000; each bit can be set to 1 to indicate that the street section is occupied.
13     /// -----
14     /// |0001|1000|
15     /// |----|----|
16     /// |0010|0100|
17     /// -----
18     /// </summary>
19     [Flags]
20     public enum OccupiedLanes
21     {
22         None = 0,
23         LeftTop = 1 << 0,
24         LeftBottom = 1 << 1,
25         RightBottom = 1 << 2,
26         RightTop = 1 << 3
27     }

```

crossing. To all these cars a drive instruction will be sent, others will be denied. Whenever a car leaves the crossing or enters it, a recalculation of the sections takes place. This system is working well at the moment, but too many cars might break the logic and accidents can still sometimes happen. I will need to look into refinements for the logic, faster and quicker updates and more time saving instructions.

The AI that will be a threat to Godzilla are the police cars. The average number on a map is currently set to 10% of all cars. However, this number can increase once the hunt is on for Godzilla. Police cars will all be set on duty and use a path finding algorithm to calculate the shortest way to the Godzilla position. When close enough, diverting from the road is allowed in order to hit Godzilla and stun it. When on duty, traffic rules might not apply anymore for the police cars and in collisions they are usually stronger than others. Speed also increases to catch the evil as fast as possible.



4.1.4 UI

Due to the fact that everything is rather colourful and overwhelming, the UI should be fairly simple.

Regarding the last week of bug-fixing and putting everything into the game, we just finished the core elements of the UI. Still, we will explain the finalized version in the next lines.

The energy bar on the left side indicates that the player has enough resources to use special attacks.

Above, you can see the remaining

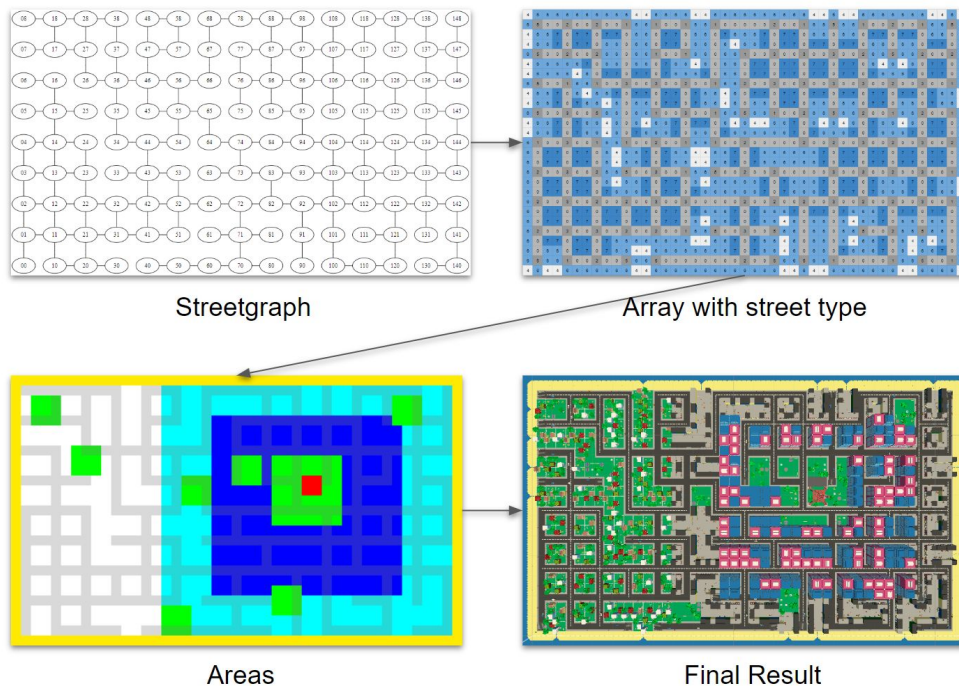
time and the “voxel” count, the number of voxels you successfully destroyed. This UI is planned to be implemented during the next weeks before the demo day.



4.2 Procedural Map Generation

The map generation has been greatly expanded and changed since the last milestone. The basic ideas behind the three-step process described in chapter 3.5.1 are still in use but some things had to be changed to increase the performance of the game. The process now looks like the following:

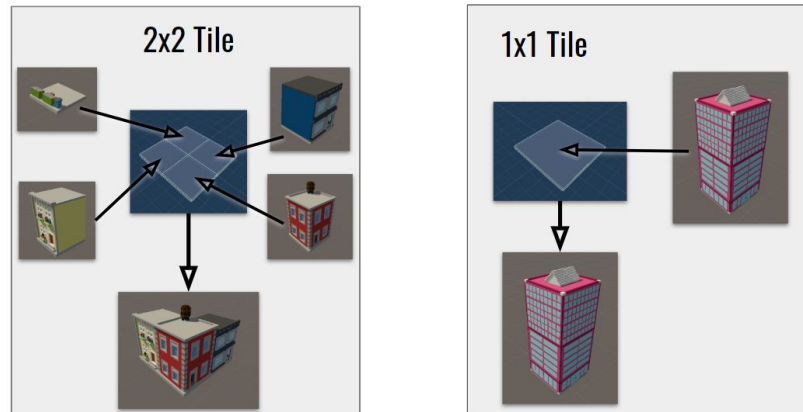
- 1) Generate a graph containing a grid (each node represents a street intersection, edges represent streets)
- 2) Delete a few edges of the graph but keep it connected
- 3) Translate graph into an array (each cell represents a 12x12 meter tile)
- 4) Define different city areas (Suburbs, City, Skyscrapers, Park, Beach, Goal)
- 5) For each cell in the grid generate a tile corresponding to the area and the type of the cell (Street [Curve, Straight, Intersection, T-Intersection, Tunnel], Building, Corner Building)
- 6) Place the tile in-game



The first steps are still the same as during the last milestone. We still generate a grid-shaped graph as basis for our street network. This makes it resemble the grid style of Manhattan. Next, we delete edges, which represent the street, to make the city feel more organic. Step three is where this generator differs from the previous version. We still translate the graph into an array, but instead of each cell representing a 1x1 meter tile, each tile is now 12x12 meter. This greatly increases performance, since we do not need as many objects, and makes designing the level easier, but comes at the cost of less control over the final result.

Each cell gets then assigned one of the following areas: Suburb, City, Skyscrapers, Park, Beach, Goal. This makes the city feel more realistic and is a great way of guiding the player through a kind of journey where they start out as small Godzilla destroying smaller objects in the suburb and end up as big Godzilla destroying giant buildings.

Lastly, the array has to be initialized. For each cell a tile is randomly generated corresponding to the type of cell and area it is placed in the game. A tile can either be made of 4 smaller sections, each of which gets randomly chosen from a selection of premade sections, or consist of only one section in case bigger objects, like skyscrapers have to be placed.



With this, the whole city is initialized and ready for the game to start.



4.3 The Game

With the full playing world generated, the game can finally begin. The following sections describe how our game developed from the fundamental implementation during the interim to its Alpha Release state.

4.3.1 Stages of Growth

The game in its current stage features three stages. At the beginning of the game, the controllable Godzilla is the size of a car. If you observe the image of the generated city above, the player spawns on the left side of the map, meaning he begins in the suburb area. In this area, there are small houses, gardens and parks that feature small properties that can be destroyed. The player can explore the map now, find objects to destroy, such as benches, cones, and traffic lights. If he exceeds enough points, he undergoes a surge of growth. He expands to the size of smaller houses and the preferred targets of destruction

change to buildings of equal size. At this point, it is a good idea to traverse in the world towards the right, where the city is located. Again, if enough points are collected in phase two, the character will grow to its final stage in which he is able to take down skyscrapers and the objective: the red and white TV-Tower in the middle of the level.

4.3.2 Balancing the Game

Given a procedural level with three stages, a random configuration of objects to destroy and a timed objective, it is necessary to bring structure into the game, by designing and balancing the values of each entity. The following sheet shows the stats of the objects in terms of hit-points and reward-points. Also, there are notes regarding the purpose of each object.

In the graphic below, level refers to the stage of growth, that is ideal to be in while attempting to destroy an object. Especially in the early stage, the player needs to explore for good objects since the variety of targets is the biggest currently. We are planning, that players in the alpha release will spend the biggest time of their sessions in this stage. Between level 1 and level 2, there is an area that is overlapping. This indicates that there are objects that can be destroyed early using powerful attacks. They reward more points, therefore, they are favourable. More on hit points and the damage each attack deals in the next section. Cars and police cars, the enemies in the Alpha Release are destructible too and a good target while being in the second stage. Finally, the last object of the list is the TV_Tower, which is the target objective to complete a level.

Time ? : 0:05 h					
Level	Destructables	Hitpoints	Reward	Point Multiplier	Idea
1	Barrel	40	16	1.0	two hit/kick
1	ExplosiveBarrel	40	16	1.0	two hit/kick
1	Crate	25	10	1.0	one punch
1	Cone	20	8	1.0	one kick
1	Trashcan	40	24	1.5	two hit/kick
1	Bench	50	20	1.0	two punches
1	hydrant	50	25	1.25	two punches
1	vase	20	12	1.5	one kick
1	TrafficLight	40	16	1.0	two hit/kick
1	Trashbin	100	40	1.0	combo or charge attack
1	Bush	100	40	1.0	combo or charge attack
1	fountain	200	160	2.0	very sturdy and much points
1	MerryGoAround	200	160	2.0	very sturdy and much points
1	Slide	100	80	2.0	two hit/kick and much points
2	Tree_small	250	100	1.0	inbetween item
2	suburban_1	400	160	1.0	two hit in level 2
2	suburban_2	400	160	1.0	two hit in level 2
2	house 1	500	200	1.0	combo or charge attack in level 2
2	house 2	500	200	1.0	combo or charge attack in level 2
2	house 3	500	200	1.0	combo or charge attack in level 2
2	"Yellow house"	500	250	1.25	combo + additional attacks in level 2
2	house_blue	500	300	1.5	house that rewards more points
2	house_c1	700	280	1.0	more sturdy house
2	car	500	200	1.5	Enemy : collision interrupts!
2	police_car	500	200	1.5	Chasing Enemy
3	sky_scraper_1	2500	1000	combo or charge attack in level 3	
3	sky_scraper_2	3000	1200	combo or charge attack + additional attacks in level 3	
3	TV_Tower	15000	4000	once	Destroy to complete the level . Firestorm at level 3

4.3.3 Player's full Action Set

Attack	Buttons [XBOX]	Damage	Growth Stage	Growth Scale
Punch	A	25	1	1
ChainSwipe	A -> A -> B	25 + 25 + 50	2	5
Kick	X	20	3	10
StrikeKick	X -> X -> B	20 + 20 + 60		
Headbutt	A -> A -> Y	25 + 25 + 100		
ChargeAttacks				
Charge	Hold R			
JumpAttack	R + Y	100		
ChargedStrikeKick	R + X	100		
Special Attacks				
Special attacks cost energy				
Fireball	Hold L + B	200 - 600	20 - 60	
Firestorm	Hold L + Y	100 * seconds	60 + s	

Now since every object has their hitpoints. Let's have a look at the tools the player has to destroy objects in this world. As he plays our Kaiju monster, he can move around freely, and use attacks to inflict damage on his environment. The full list of attacks can be seen on the left.

The **regular** attacks are moves that can be performed at anytime. Punch is the most regular attack, kicks are especially useful against small objects. Attacks can be chained together and ended with strong finishers. Such as the **StrikeKick**.

There are also **Charge** attacks that require the character to be charging. While charging, the character becomes faster and less easy to control. Charge attack stop the assault with some good damage.

Finally there are **special** attacks that require spending your energy. Energy is a concept we introduce in the Alpha Release. Every successful attack on a object in similar size creates energy. Good timing while pressing the buttons will reward even more energy. It can stack up to a value of 100. Special attacks are very powerful and most likely necessary to win the game.

Finally each attack damage is multiplied by the **growth Scale** that is based on the size of the character.

We are looking forward to seeing how the playtesters will play our character and how challenging it is to destroy the tv tower in a given time of 5 minutes.

5. Playtesting

5.1 Playtest Setup

Our Playtest was organized as follows. One of the team members was present at all times during a session. He/She not only observed the player and wrote down remarks and immediate feedback, but also assists, if intervention is really necessary. That is the case for game breaking bugs or if the player is stuck caused by technical difficulties, such as display and input problems. Information about the objective in the game and explanations regarding player control or what to do, has not been provided. The player was only informed about the game via an information screen that was displayed at the beginning. The infoscreen contained information about: The objective to destroy the TV Tower, how to control the character, and that he'll eventually grow by destroying objects of similar size. In some cases we carried out the session by letting the player stream from their computer and talk to them via voice chat.

A play session with one person contained at least one playthrough or more, if the player wanted to attempt the game again. Each game generates a new map and has a duration of 5 minutes maximum to complete the objective. Players were provided a gamepad, Xbox 360 preferred, but Dualshock 4 (PS4) and XBOX One has been used too. The PC device used vary from high-end gaming devices to less stronger laptops, at least with a designated graphics card.

More about the questionnaire in the next chapter. Finally when the player announces that he/she is done playing, we asked them to fill out the questionnaire, whose contents are discussed in the next chapter. Afterwards, questions about the game were answered.



5.2 Questionnaire

The questionnaire was provided online via google forms and could be filled out on a laptop, tablet or smartphone which we provided for the testers.

The questions were split into four categories, starting with general questions about the testers and their gaming habits, followed by questions about the game, questions about the controls/interface and finally questions about performance and visuals with space for additional notes.

In the general section we asked the following questions:

- Please select your age group: [18-25, 26-30, 31-40, 41-50, 51-65, 65+]
- Please select your gender: [Female, Male, Diverse]
- I love to play games in my free time. [Strongly Disagree (1) ... Strongly Agree (5)]
- Please select all game genres you enjoy playing in your free time: [Action, Action Adventure, Adventure, Role Playing, Simulation, Strategy, Sports, MMO, Hack & Slay]
- How familiar are you with Fighting/ "Punch'n'Destroy" games? [Not familiar (1)... Very familiar (7)]

The "Game" section contained these questions:

- I found the objective clear at all times. [Strongly Disagree (1) ... Strongly Agree (5)]
- Additional space for feedback for the games objective: [Free form]
- What was your strategy for winning? [Free form]
- How many rounds did you play? [1, 2, 3, 4+]
- Did you complete the game? [Yes, No, I don't know]
- How do you feel about the difficulty of the game? [Too easy(1)... Too hard(5)]
- What do you think about the cops? [They improved the overall game feeling., They made the city lively., I felt challenged., I was bothered., I did not mind., Sonstiges: (Free form)]

The "Controls and Interface" section tested for:

- Did you get stuck during any parts of the playtest? [Yes, No, Maybe]
- I found the controls of the gameplay very intuitive. [Strongly Disagree (1) ... Strongly Agree (5)]
- The control schemes were logical and easy to remember. [Strongly Disagree (1) ... Strongly Agree (5)]
- Which things would you change/add about the interface and/or controls of the game. [Free form]

And finally in the last section we have:

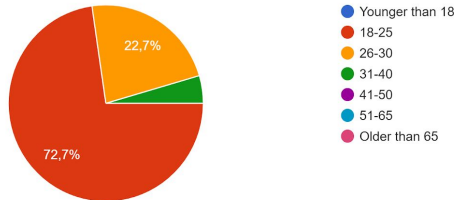
- The performance of the game was very smooth during my play-test and I had no frame stuttering or other issues. [Strongly Disagree (1) ... Strongly Agree (5)]
- I found the overall appearance of the game very appealing [Strongly Disagree (1) ... Strongly Agree (5)]
- If you could change just one thing, what would it be? [Free form]

5.3 Results

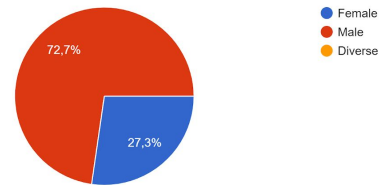
5.3.1 General Information

The test demographics are mostly as expected: an almost exclusively young, majority male group of people that love to play videogames in their free time. No real surprises here.

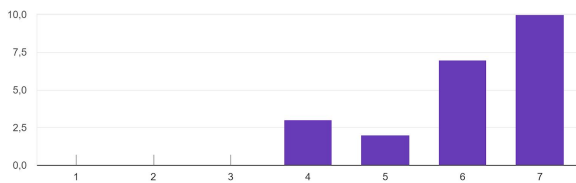
Please select your age group:
22 Antworten



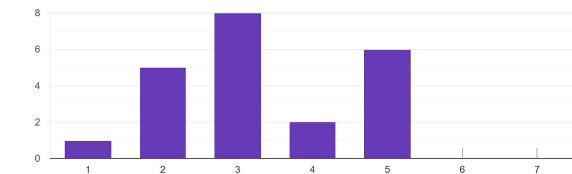
Please select your gender:
22 Antworten



I love to play games in my free time.
22 Antworten

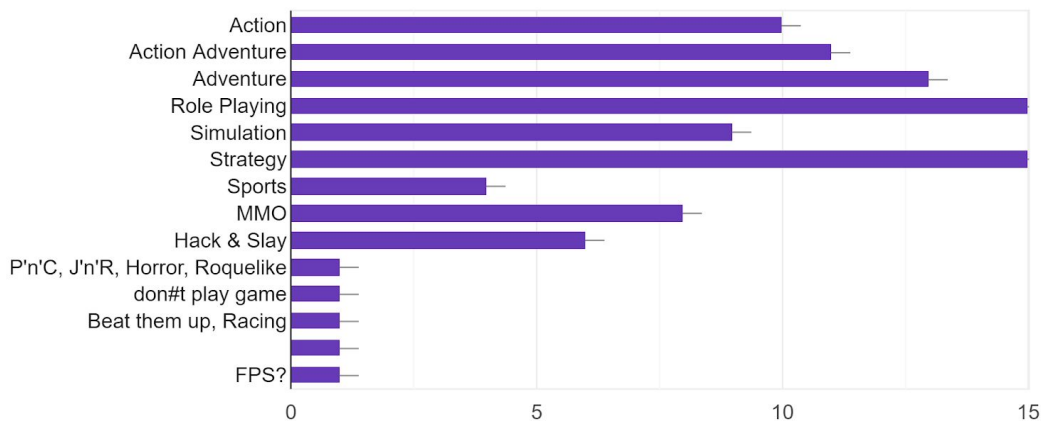


How familiar are you with Fighting/ "Punch'n'Destroy" games?
22 Antworten



The games our testers play in their free time are hard to summarize in a few sentences. For us it is mostly interesting that a lot of them like to play action (45,5%) or action adventure (50%) games, and are, therefore, at least somewhat used to the controls and mechanics of our game. The questionnaire also showed that our testers where only medium to slightly familiar with fighting games.

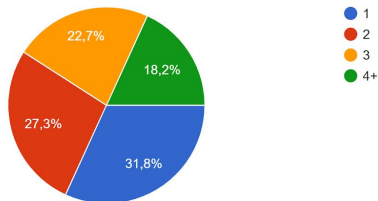
Please select all game genres you enjoy playing in your free time:
22 Antworten



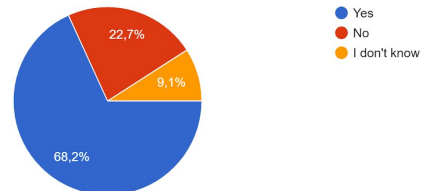
5.3.2 Game Questions

In this section we asked testers how many rounds they played and whether or not they completed the game. We asked these questions so we can understand how much time the players invested in the game and how successful they were, and use that to put their answers into perspective.

How many rounds did you play?
22 Antworten

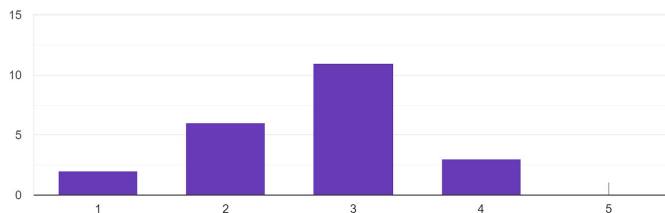


Did you complete the game
22 Antworten



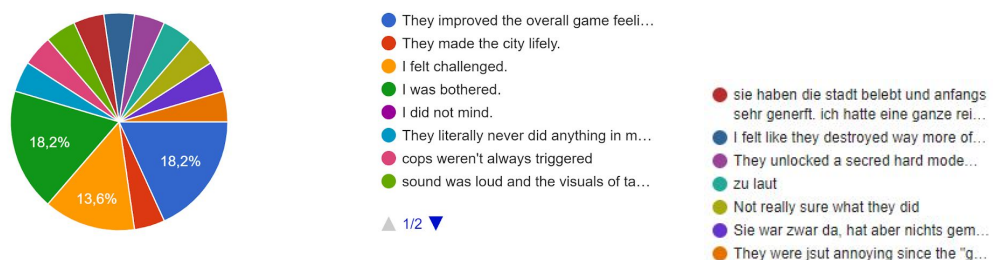
The questionnaire also showed us that most players liked the difficulty of the game, finding it neither too easy nor too hard.

How do you feel about the difficulty of the game
22 Antworten



An additional question about the police cars in the game revealed that these were a nice addition to the game, adding a bit of challenge, even though quite a few players were bothered by them.

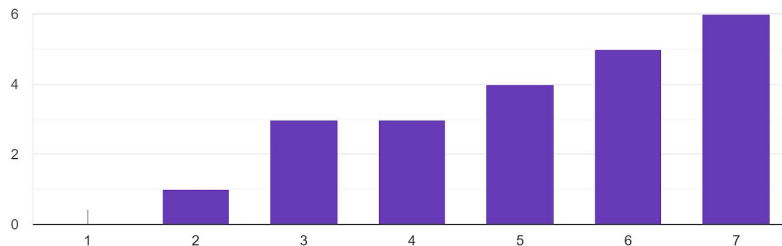
What do you think about the cops?
22 Antworten



The optional question about the testers strategy for winning the game could be answered freely. This resulted in some great insight into the players behaviour and their understanding of the game. It revealed that most players, once they have found a working combo, would use this single combo repeatedly to destroy as much objects as fast as possible as they could. It also showed that, while some were a bit lost, others understood the game and devised a working strategy.

Lastly, we asked two questions about the game objective: whether or not the goal was clear at all times on a scale from 1 to 7, and one that could be answered freely. While the scale showed that the overall objective was clear, the second question revealed that there is still room for improvement, especially when indicating the growth progress of the player character. Also the tutorial screen needs to be enhanced.

I found the objective clear at all times.
22 Antworten

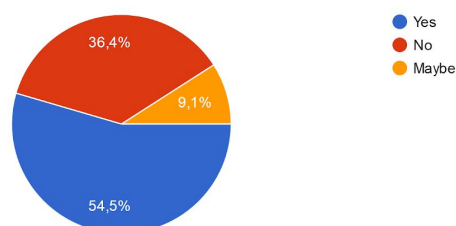


5.3.3 Control and Interface

In this section, we wanted to know, what our testers thought about the controls and interface. These questions were added because we wanted to make sure that our controls and the feedback they gave were efficient enough to feel intuitive. We also wanted to understand if the player was stuck during the game, so we could avoid unnecessary frustration regarding the controls. The result would help us to understand whether or not those needed to be revisited.

We asked our player whether or not they got stuck during any part of the playtest. Unfortunately, more than half of our testers got stuck in some kind of way during the game. Different colliders from the debris of destroyed objects hindered the player to continue their playthrough. Additionally, the controls turned out to be not as responsive as planned. More time has to be invested in those topics.

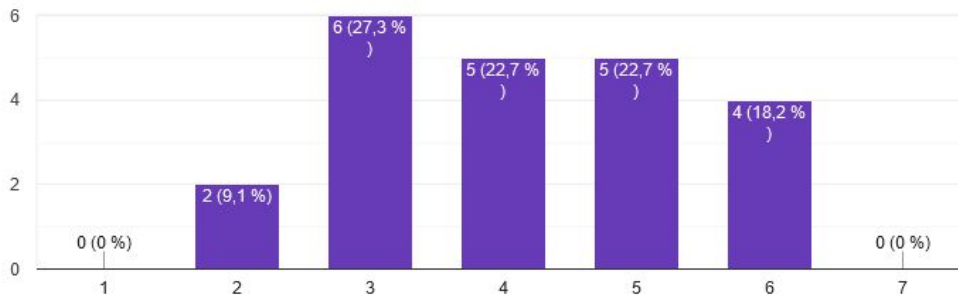
Did you get stuck during any parts of the playtest?
22 Antworten



Asking the players to rate the intuitiveness of the controls by the numbers 1 to 7, we tried to understand if the players were satisfied with our current control-settings. The player found the controls to be not very intuitive as they encountered situations in which controls seemingly had no effect.

I found the controls of the gameplay very intuitive.

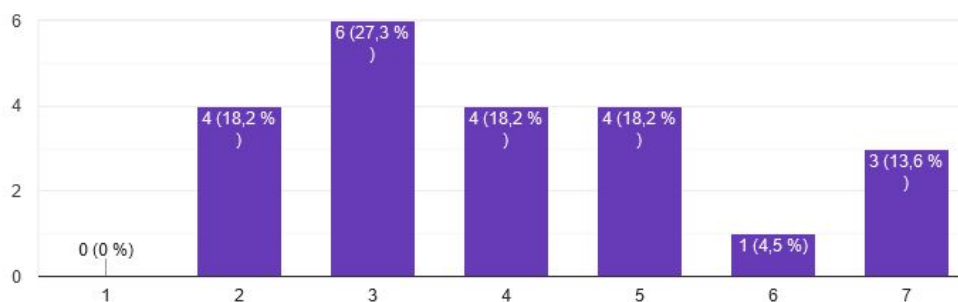
22 Antworten



The intuitiveness of the controls can be also explained by the fact that players couldn't remember control schemes that easily, as can be recognized by the next question. Rating the simplicity of the control schemes we could tell that current schemes did not work well for our players. They were too complicated and required too many different button presses for the players to be truly easy.

The control schemes were logical and easy to remember.

22 Antworten



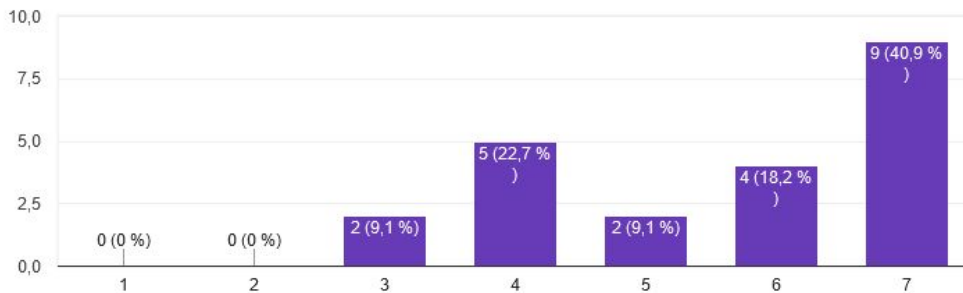
Being asked what we could improve regarding the controls and the interface, we got several different answers. The player asked for more guidance during the game, as the combos weren't easy to learn and also gave little feedback when the combo was done correctly. An overlay screen with the controls or a detailed tutorial was suggested. Also, the lack of feedback when hitting a house or an object was mentioned. Players, moreover, criticized the lack of responsiveness for the attack as in the camera control, as the camera could get stuck between different object, not showing what is going on. The testers also asked for more feedback regarding the police and the growth. It was hard for them to understand how long a police car will be chasing them and when it will be destroyed. In addition, they were confused about what parameters had to be achieved to make Godzilla grow. More UI features explaining this were asked for.

5.3.4 Final Questions

The final questions were needed to fill the gaps on the overall player experience. We wanted our players to rate their gameplay regarding the technical performance. The performance was surprisingly good for almost all our testers so that a smooth course of the game could be provided.

The performance of the game was very smooth during my play-test and I had no frame stuttering or other issues.

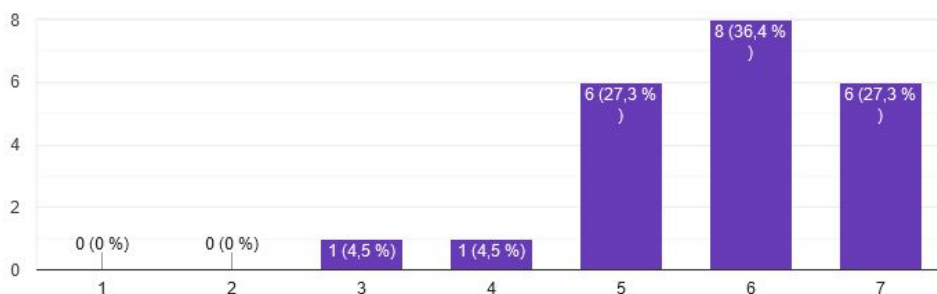
22 Antworten



Wanting to know more about the overall appeal of the game we asked to rate the appeal, meaning UI, graphics and general “look and feel”. Regarding the design and the general appeal of the game, the player seemed satisfied. The players liked the look and feel of the game, providing only little critic. Details like missing designs on the back of some structures were mentioned regarding this topic.

I found the overall appearance of the game very appealing

22 Antworten



Lastly, we asked our tester what they would change if they got to change one thing. Most of our player wished for better control and visual feedback. More indicators for the growth of Godzilla were also wished for. Additionally, they would allow Godzilla to walk through the debris as they would often get stuck for a few seconds during the gameplay. Some players also wished for less police as they were highly bothered by those.

5.4 Changes

After analyzing the results of the questionnaire, individual feedback and notes that we have taken during the playtests, there are many minor and major changes that need to be addressed.

5.4.1 Gameplay/-design

Gameplay-wise we are very pleased with the playtesting results. Without problems, each tester was able to understand the main objective of destroying the cities landmark. Therefore, only smaller quality-of-life improvements will be implemented.

First of all, once the character grows, smaller objects will be automatically destroyed by Godzilla when walking over them. We recognized that fences or trees could block Godzilla from a smooth journey through the city which does not make sense in the first place and hinders the game flow.

Similarly, the debris of houses will no longer collide with either character or cars as it was an additional obstacle blocking the player from navigating properly. This will also slightly increase the overall performance of the game.

A suggested change to the growth system, namely changing the procedure into a continuous growth, has been considered. Despite being evaluated as game improving, we will not be able to implement it in the time left. As a result, we will focus on improving the gaming experience with the current growth system to make it as satisfying as possible.

5.4.2 Controls

For better learning results we already have implemented an additional demo tutorial that explains controls step by step. However, unfortunately this has not been finished for the playtests. The tutorial is taking place on a very small scale map that introduces the player to the basic game concepts. It is making use of the task system that is used during a normal game run as well. This tutorial will certainly be helpful for teaching controls at the demo day.

Besides, we are going to rework the control scheme completely. Simplifying the combos is the main goal which we will achieve by changing the two main combos from an A -> A -> B or X -> X -> B to a way easier A -> A -> A and X -> X -> X respectively. This should be more intuitive and support the "rewarded" feeling of players as timing a combo is not very obvious in the first place. In addition, visual and audio feedback is being considered for correct inputs. This has already been part of our desired target milestone but did not make it into the build due to complex logics.

Moreover, the special attacks, such as the fireball and firebreath, have not been used at all by players in their playthroughs. Thus, we will move the controls to the Y button that will now not be in use anymore due to the reworked main combos. This further simplifies the control scheme and should make all actions executable just by using the main four buttons A, B, X, Y and the shoulder button for sprinting.

Another greater issue is the handling of camera movements. We will address the camera being stuck by removing collisions with smaller objects such as trees, which quite often caused the player to struggle when walking through an in-game park for example. We are also considering working on the camera in general and adding other ways that could make camera collision checking obsolete or we consider to sway the camera softly behind the character.

5.4.3 UI

Regarding the UI a few changes will be made. The UI size will be adjusted as it seems to be too small at this moment. Due to the fact, that the players found the tutorial screen to be overloaded and complicated, it will be reworked. Firstly, button descriptions will be rewritten to be more precise. Secondly, the combo-sheet will be made easier to read by replacing “+” with “->” as it will be easier to understand that buttons are needed to be pressed successively. The overall design will be changed to be easier readable.

As we provide several different combos, which can be easily forgotten in the first play runs, a controls/combos menu will be added to the game. It will be possible to press a button during the game to get a quick overview of attack combinations and general controls.

The already implemented task box will add an additional layer of explanation, as a new task will be added to aid the player in his playthrough. Through simple milestones, we will guide the player so that he can achieve his goal more easily. Therefore the player should understand then he will reach his new height and when it is the best moment to destroy the goal.

5.4.4 Bugs

The most severe bugs found during playtesting will be addressed in the following paragraphs.

For starters, the character’s growth animation is stopped when being run over by a car in the meantime. This lead to players not knowing that they actually gained strength and were now able to destroy bigger obstacles. This is not a game-breaking bug, but highly confusing and will be dealt with as soon as possible.

When shooting a fireball while standing too close to a house, the fireball would skip the first house and not deal any damage to it. We already know a solution to this problem and will have it fixed by the next build.

Regarding the cars there are a couple of issues that need to be dealt with. For one, after cars hit a player, the character is stopped at exactly that position. This leads to the car waiting for the player to get up which possibly causes another stun right after regaining control. Pushing the player away in the collision direction will grant more visual feedback and prevent the car from waiting right in front of the character. Secondly, while the game is paused, cars and police continue driving. This in the engine unfortunately coupled with the input system which makes it difficult switching off the rigidbodies. Quite possibly we need to come up with a work around that saves previous values and reinitiates them when resuming.

5.4.5 Appeal/Graphics/Feedback

Continuing the topic about cars, the stun animation, played when the character is being hit by a car, will be reworked. This should go in conjunction with the possible changes about Godzilla being pushed away from the car in driving direction. As an optional addition, controller vibration might be added for collisions with cars.

Another main issue of the playtest build was the lack of visual and auditory feedback for the players actions. These were due to time constraints in our last milestone, but will have priority until the final release. Most players were confused about what they could do and what would be destroyable. We will add additional audio feedback for punches, hits, destructions, special attacks, footstep sounds, background music and ambient noise. Also, reworking the police siren sound and remastering the mix of all sounds and effects is necessary.

On a visual perspective, the voxel particle effects appearing when punching an object will be enlarged and increased for more user feedback. This also has to scale with Godzilla's growth to stay visible all of the time.

A main reason why players did not dare to destroy houses was that it was not noticeable whether it was taking damage or not. As a solution, the destruction texture shader will be adjusted so that from the first punch damage is visible. The same applies to cars that were lacking a similar shader up until now.

With all these changes in mind, we hope to be able to implement as many as possible until the final release and improve the overall game experience significantly.