

Wave Voyager

© CJ Productions

v1.0, Chris Loch, Jan-Eric Schober, 09.11.17

Table of contents

1. Game Overview	4
1.1. Game Concept	4
1.2. Genre	4
1.3. Target audience	4
1.4. Game Flow Summary	4
1.5. Look and Feel	4
1.6. Project Scope	5
1.6.1. Environment Design	5
1.6.2. Number of NPCs	5
1.6.3. Number of vehicles	5
1.6.4. Number of upgrades	5
2. Gameplay and Mechanics	6
2.1. Gameplay	6
2.1.1. Game Progression	6
2.1.2. Challenge Structure	6
2.1.3. Objectives	6
2.1.4. Play Flow	6
2.2. Mechanics	6
2.2.1. Physics	6
2.2.2. Movement	7
2.2.3. Objects	7
2.2.4. Actions	7
2.2.5. Combat	7
2.2.6. Economy	7
2.3. Screen Flow	8
2.3.1. Screen Flow Chart	8
2.3.2. Screen Flow Description	9
2.4. Game Options	10
2.5. Replaying	10
2.6. Cheats and Eastereggs	10
3. Setting and Vehicles	11
3.1. Game World	11
3.1.1. General look and feel of the world	11
3.1.2. Procedural Environment Generation	11
3.1.3. Color Schemes	11
3.2. Vehicles	12
3.2.1. Vehicle #1	12

3.2.2. Vehicle #2	12
3.2.3. Vehicle #3	12
4. Level Generation	13
4.1. Track	13
4.2. Environment.....	13
4.3. Enemies.....	13
4.4. Obstacles	13
4.5. Pickups	13
5. Interface	14
5.1 Virtual System.....	14
5.1.1 HUD	14
5.1.2 Menus	15
5.1.3 Rendering System	19
5.1.4 Camera.....	19
5.2 Control System	19
5.3 Audio	19
5.3.1 Music	19
5.3.2 Sound Effects	19
5.4 Help System.....	19
6. Artificial Intelligence	20
6.1 Enemy AI	20
7. Technical.....	20
7.1 Target Hardware.....	20
7.2 Development Hardware and Software.....	20
7.3 Development Features and Standards.....	20
7.4 Game Engine.....	20
8. Game Art.....	21
9. Secondary Software	21
10. Management.....	21
10.1 Risk Analysis	21
10.2 Localization Plan.....	21

1. Game Overview

1.1. Game Concept

- The concept of the game is basically a space racer, in which the player has to destroy or dodge obstacles and enemies.
- The environment, all objects and enemies as well as possible pickups are procedurally generated by the characteristics of a music title of the players choice.
- The player can earn points for several events and actions.
- The player is traveling forward all the time, can move in two directions (Up/Down & Left/Right), and have several weapons and abilities per vehicle.
- Vehicles and upgrades can be bought by points earned while playing a game.

1.2. Genre

- Music
- Indie
- Racing
- Shoot 'em up

1.3. Target audience

- Casual gamers

1.4. Game Flow Summary

- At first the player selects a song.
- After that, the level data is being generated, and the player begins "riding" his song until the end, earning rewards for shooting enemies and dodging obstacles.
- After the player have finished the song, he will receive a final rating of his performance.

1.5. Look and Feel

- The game is in 3D.
- All models will consist of simple geometrical shapes.
- A colored stencil-style will be used for all models, the environment and the effects.
- The player will have the feel to ride inside a tube or on a track through a fancy and colorful universe.
- The environment is adapting to the song the player is listening to.

1.6. Project Scope

1.6.1. Environment Design

- The environment is an open universe, which is filled by various shapes and effects spawning randomly.
- 3D-Models as well as 2D-Effects will be used to fill the empty space.
- The player is riding on a sort of track.

1.6.2. Number of NPCs

- 3-5 different enemy types.

1.6.3. Number of vehicles

- One entry level vehicle.
- 4 vehicles which can be bought with points.

1.6.4. Number of upgrades

- There will be at least three upgrades in each of the following categories and per vehicle:
 - Engine
 - Shield/Resistance
 - Weapons
 - Abilities

2. Gameplay and Mechanics

2.1. Gameplay

2.1.1. Game Progression

After a song has been selected by the player and the game has started, the player is riding along a path throughout the whole song, and has to shoot and dodge its way through the enemies and obstacles which are being spawned. During this time the player has several possibilities to earn points, and his performance will be rated after the song has been finished.

After that, the player has the chance to spend his earned points, or start another game.

2.1.2. Challenge Structure

- The challenge is to earn as much points as possible. This can be achieved by killing enemies, destroying and/or dodging obstacles and activating several abilities.
- When the player gets damaged, he will receive penalties, such as point loss or a weapon lock.
- The better the player performed, the more points he will get with the final rating. (Such as bonuses for fulfilling several goals.)

2.1.3. Objectives

- Collect as much points as possible.

2.1.4. Play Flow

- The player is riding his vehicle through space and time for the whole duration of the song.
- While the song is playing, enemies and obstacles and pickups are being spawned. Positions, quantities and such variables are modified by characteristics filtered out of the particular song.

2.2. Mechanics

2.2.1. Physics

- You're in space. No gravitation!

2.2.2. Movement

2.2.2.1. *Player Movement*

- The player can move in 2 DoF (Up/Down & Left/Right).
 - The player can't move forward or backward manually, the forward motion is constantly present and its speed is determined by the current pace of the particular song.

2.2.2.2. *Enemy Movement*

- The enemy AI will always try to fly towards the player, but in different movement patterns (Spirals, squares, straight, etc.).

2.2.3. Objects

2.2.3.1. *Picking Up Powerups*

- Powerups can be picked up by touching them with the vehicle.
- The certain powerup will be activated upon touching.

2.2.3.2. *Destroying Objects*

- Objects can be destroyed by shooting them.

2.2.4. Actions

2.2.4.1. *Shooting*

- The weapons can be used by pressing a configured button. The projectiles will travel in a straight line, coming from the players' vehicle itself.

2.2.4.2. *Switching Weapons*

- If a vehicle has multiple weapons, it's possible to cycle through those and select any weapon available for the current vehicle.

2.2.5. Combat

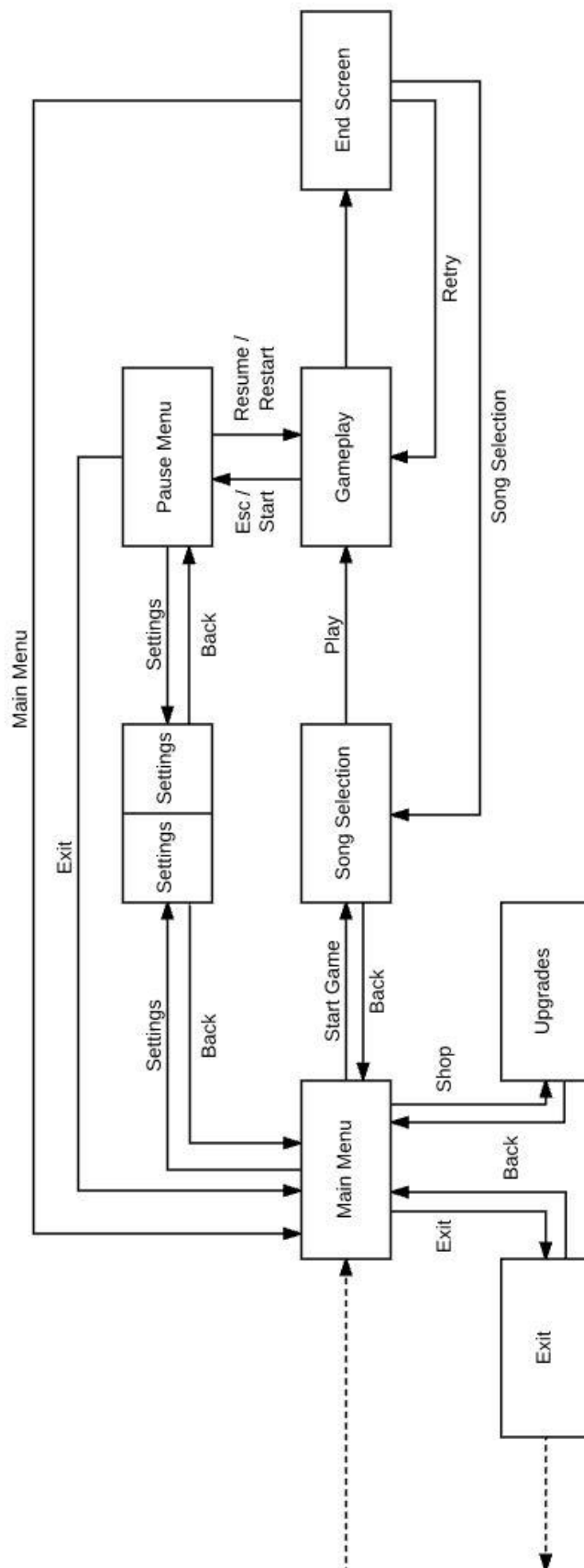
- The player will fight versus the enemy AI. Both parties will settle their arguments by shooting each other.

2.2.6. Economy

- Points earned during a game can be used to buy new vehicles and upgrades in the shop.

2.3. Screen Flow

2.3.1. Screen Flow Chart



2.3.2. Screen Flow Description

2.3.2.1. *Main Menu Screen*

Options and screens accessible from the “Main Menu”-Screen:

- Start Game Screen
- Settings Screen
- Shop Screen
- Scoreboard Screen
- Exit (Kill the program)

2.3.2.2. *Start Game/Song Selection Screen*

Options and screens accessible from the “Start Game”-Screen:

- Song Selection
- Play (Switch to Gameplay Screen)
- Back (To the Main Menu Screen)

2.3.2.3. *Gameplay Screen*

Options and screens accessible from the “Gameplay”-Screen:

- Pause Screen (triggered by pressing Escape)
- End Screen (after finishing a song)

2.3.2.4. *End Screen*

Options and screens accessible from the “Gameplay”-Screen:

- Retry (Will restart the current song)
- Song Selection (Back to the Start Game Screen)
- Main Menu (Screen)

2.3.2.5. *Settings Screen*

Options and screens accessible from the “Settings”-Screen:

- Audio-Settings
- Video-Settings
- Keymapping
- Back (To the Main Menu Screen)

2.3.2.6. *Shop Screen*

Options and screens accessible from the “Shop”-Screen:

- Option for buying Vehicles
- Option for buying upgrades
- Back (To the Main Menu Screen)

2.3.2.7. *Pause Screen*

Options and screens accessible from the “Pause”-Screen:

- Resume & Restart (Back to the Gameplay Screen)
- Settings Screen
- Exit (Back to the Main Menu Screen)

2.3.2.8. *End Screen*

Options and screens accessible from the “Scoreboard”-Screen:

- Highest scores per song and difficulty.
- Back (To the Main Menu Screen)

2.4. *Game Options*

- It should be possible to select from at least three different difficulty-levels.
 - Easy
 - Intermediate
 - Hard
- With each level, the enemies will grow stronger, and there will be more obstacles.

2.5. *Replaying*

- Songs can be replayed as often as desired.
- The best Highscore will always be saved for the scoreboard.

2.6. *Cheats and Eastereggs*

- There will be multiple cheat codes, which will all result in the magical deletion of all data in the players’ database. Following events can happen:
 - Ryu will destroy the database with a Hadoken.
 - A wild Chuck Norris appears, and will perform a roundhouse-kick which destroys your database and all your pride completely.
 - Scorpion will perform a fatality on your database.
 - You’re getting Rick Roll’d and your database will quit its service.
 - All your database are belong to us.

3. Setting and Vehicles

3.1. Game World

3.1.1. General look and feel of the world

- The scenery will be set in a vast open space.
- The empty space will be filled with simple 3D shapes and 2D effects, colored in different patterns.
- The player will ride along a track consisting of lines.
- The whole environment will react to the currently playing song and different in-game events (Such as destroying enemies and obstacles).
 - Shapes will pulse and rotate.
 - Different Effects will spawn on the horizon.
 - Colors will change.

3.1.2. Procedural Environment Generation

- The whole environment will be generated based on several characteristics of the selected song. The different variables taken into account may be:
 - Different extremes in the amplitude.
 - Extremes in the frequency distribution.
 - Changes in the pace of the song (BPM).
- Following things will be influenced, based on the variables listed above:
 - The track, which will form a three-dimensional spline in space.
 - The shapes, which will be spawned in the empty space along the whole track.
 - The colors of every object and effect, which will change in different patterns.

3.1.3. Color Schemes

- The general color scheme will consist of a monochrome background (light or dark), whilst everything else will be colored.
- The different objects and effects will have independent color palettes, in which they can be colored.

3.2. Vehicles

COMING SOON(-ER OR LATER)

3.2.1. Vehicle #1

3.2.1.1. *Look*

3.2.1.1.1. *Physical Characteristics*

3.2.1.1.2. *Special Abilities*

3.2.1.1.3. *Statistics*

3.2.2. Vehicle #2

3.2.2.1. *Look*

3.2.2.1.1. *Physical Characteristics*

3.2.2.1.2. *Special Abilities*

3.2.2.1.3. *Statistics*

3.2.3. Vehicle #3

3.2.3.1. *Look*

3.2.3.1.1. *Physical Characteristics*

3.2.3.1.2. *Special Abilities*

3.2.3.1.3. *Statistics*

4. Level Generation

4.1. Track

- The track will be generated based on several characteristics of the song. It will form a spline in three-dimensional space, on which the player is riding along throughout the song.
- The inclinations and declinations of the curves will represent the pace of the current song. The speed is adapting to the distances you have to travel, since the different curves will result in different track lengths for those sections.

4.2. Environment

- Based on the characteristics mentioned above, the environment will be filled with different geometrical shapes and effects.
- The shapes are placed alongside the track, and will have several individual features:
 - Size
 - Color
 - Rotation speed/momentum
- Different effects (like explosions and such) will also be spawned, for example whenever a specific high or low note is exceeded.

4.3. Enemies

- Enemies will be spawned, based on different beats occurring in a song.
- Amount and time of enemies spawned is also determined by characteristics of the song, such as the specific amplitude of a beat or section-

4.4. Obstacles

- Obstacles will be spawned, based on different characteristics of a song.
- Amount and position of obstacles spawned is also determined by characteristics of the song, such as the specific amplitude of a beat or section.

4.5. Pickups

- Pickups will be spawned, based on different characteristics of a song.
- Amount and position of pickups spawned is also determined by characteristics of the song, such as the specific amplitude of a beat or section.

5. Interface

5.1 Virtual System

5.1.1 HUD

Simple HUD with:

- points the player reaches
- active Power Ups
- time remaining within the song
- number of collectibles
- equipped weapon

Mockup:



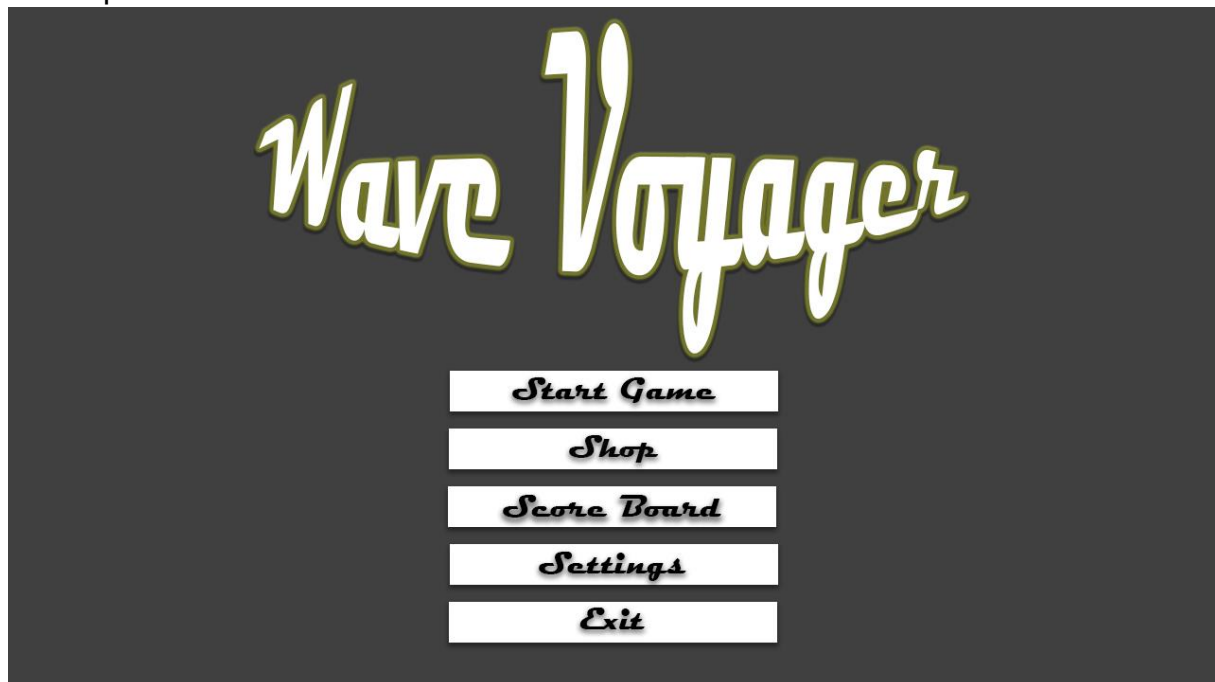
5.1.2 Menus

5.1.2.1 Main Menu

The first Menu that is seen by the player after starting the game. It contains following buttons:

- “Start Game” to choose a song and start playing.
- “Shop” to browse available upgrades.
- “Score Board” to watch the players local scores.
- “Settings” to open the game settings.
- “Exit” to quit the game.

Mockup:



5.1.2.2 Pause Menu

A menu the player can open by pressing the Esc button to pause the game. It contains following buttons:

- “Resume” to continue the session
- “Restart” to restart the current session
- “Settings” to open the game settings
- “Exit” to exit the current session

Mockup:

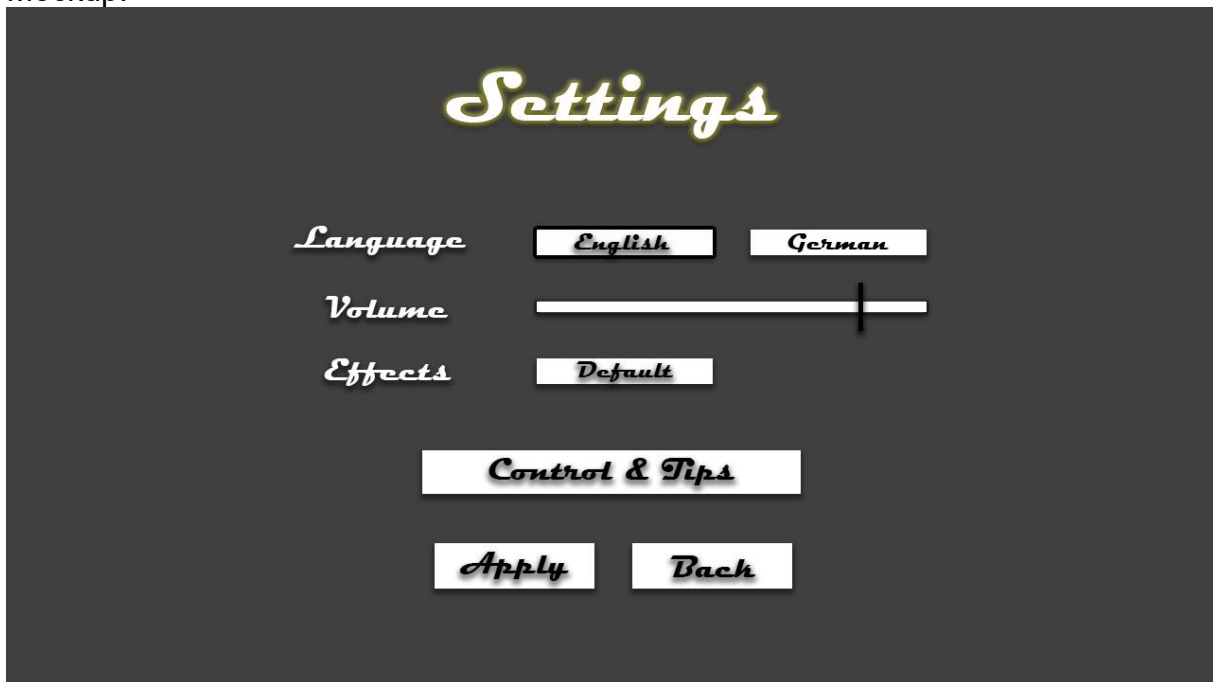


5.1.2.3 Settings Menu

A menu where the player is able to take a look into the controls and some tips aswell as to adjust some game settings. It contains following options:

- “Language” to change the gama language
- “Volume” to adjust the game volume
- “Control & Tips” to see the controls and some tips
- “Apply” to apply his/her changes
- “Back” to go back to the Main Menu

Mockup:



5.1.2.4 Endscreen

The screen the player will see if he finishes a level. It will show the name of the song, points scored, number of gathered in collectibles and it contains following buttons:

- “Retry” to replay the previous level
- “Song Selection” to choose another song
- “Main Menu” to go back to the Main Menu

Mockup:



5.1.3 Rendering System

The standard rendering system of Unity will be used.

5.1.4 Camera

The camera will be in a fix angle and it will follow the player.

5.2 Control System

The Game can be played by using mouse and keyboard or a game-controller.

5.3 Audio

5.3.1 Music

The music track is chosen by the player before a session starts.

5.3.2 Sound Effects

The game will contain following sound effects:

- explosion sounds for the player and enemies
- shooting sound for weapons
- pickup sounds
- sounds for destroying obstacles

5.4 Help System

The help system can be accessed through the “Help” button in the start menu.

6. Artificial Intelligence

6.1 Enemy AI

Coming soon.

7. Technical

7.1 Target Hardware

The game will be developed for Windows computers and is planned to be played even on older hardware. There is no specific hardware information yet.

7.2 Development Hardware and Software

Hardware:

- Computers which are running Windows.

Software:

- Unity
- Microsoft Visual Studio
- Microsoft Visual Studio Code
- Dock-Dock-Docker!
- Photoshop
- Visual Paradigm
- Aris Express

7.3 Development Features and Standards

Coming Soon.

7.4 Game Engine

The engine used for developing the game is the free version of Unity 3D.

8. Game Art

Coming soon.

9. Secondary Software

Coming soon.

10. Management

10.1 Risk Analysis

Coming soon.

10.2 Localization Plan

The game will contain following languages:

- English
- German (If there is enough time left)