

TYPEFIGHTER

Pre - Production

Serious Game Project at JMU University of Würzburg, by

Lena Lang

Jessica Topel

Lydia Bartels

Marlon Franz

Benedict Bihlmaier

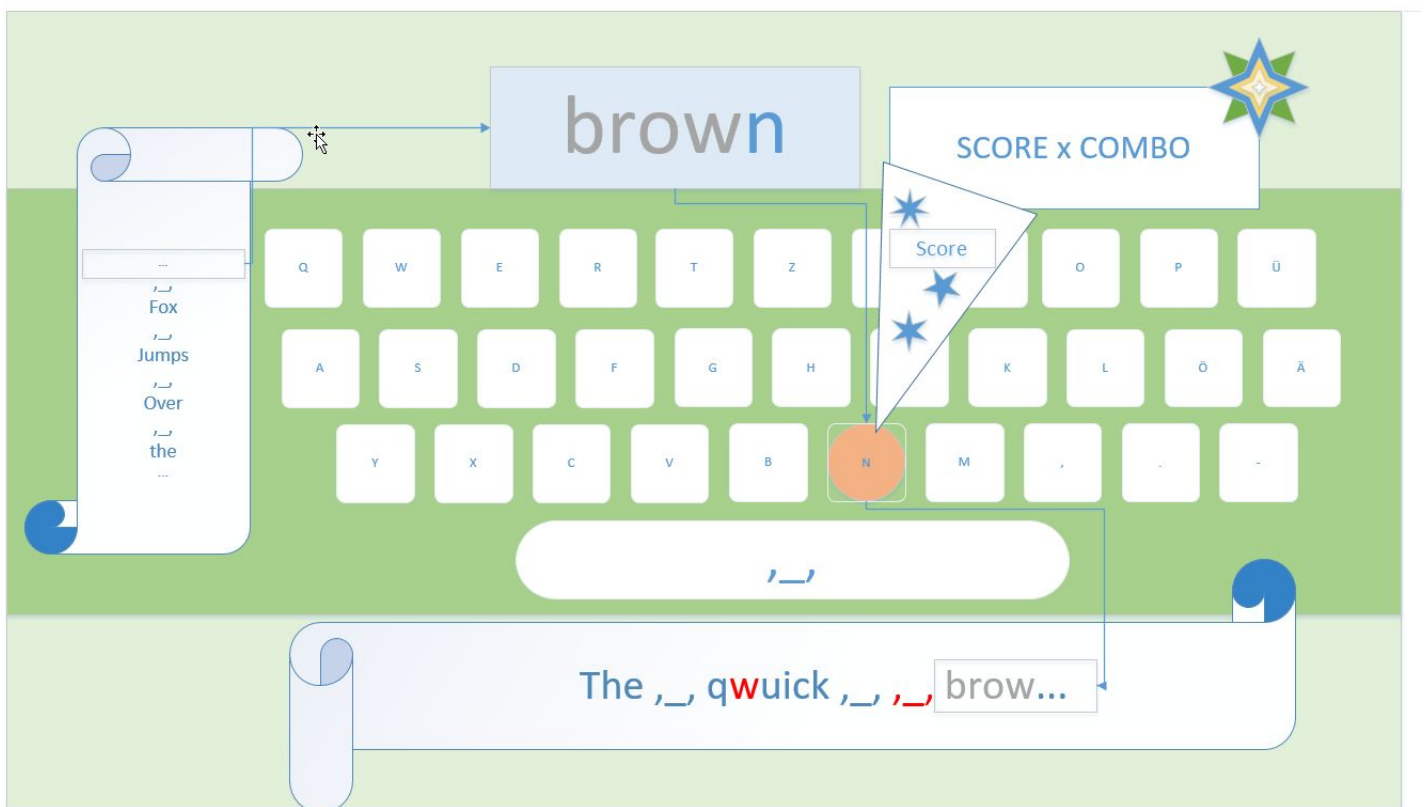
Introduction	3
Concept Stage	3
Use-Cases: Type-Trainer	4
Concretizing	5
Technical Requirements	6
Database setup	8
Final Preparations	9

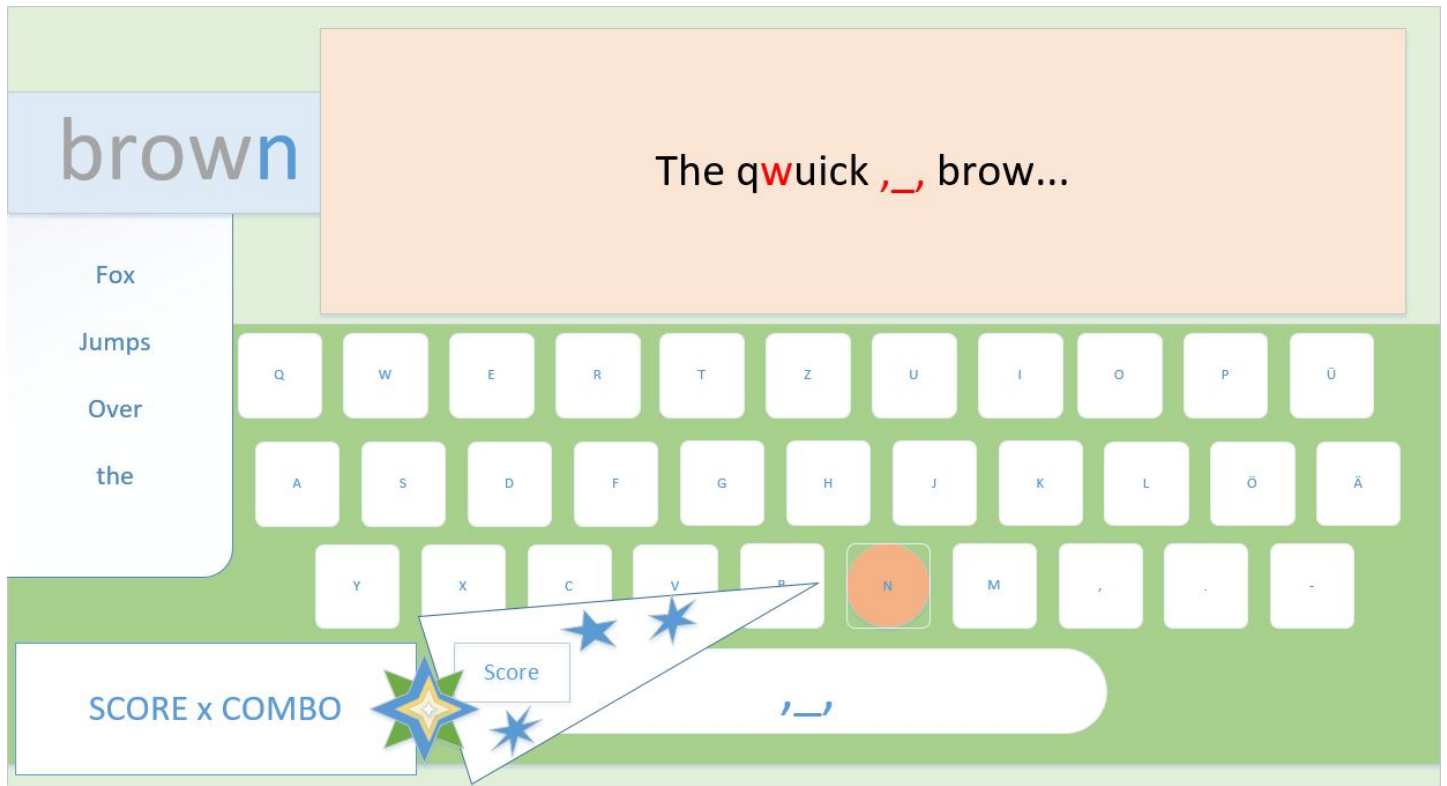
Introduction

This document summarizes the design process of the TypeFighter architecture and interface.

Concept Stage

We started by discussing a rough concept of a game, that would allow the user to learn typing using a keyboard. To concretize our idea, we created two sketches of what the game's interface should look like:





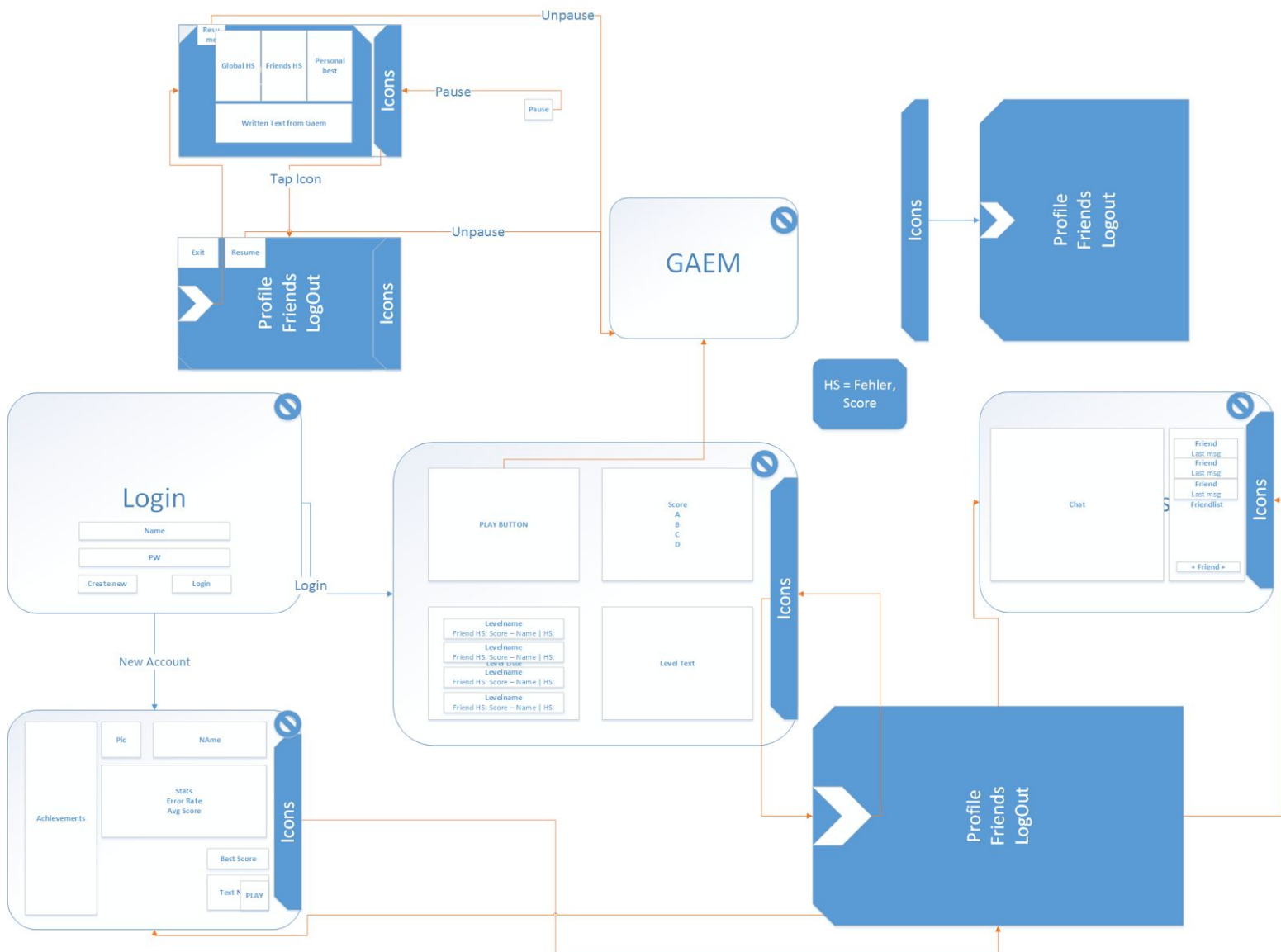
To get an overview of the different features that had to be implemented, we defined the following use-cases:

Use-Cases: Type-Trainer

1. Users can create an account using the web interface.
2. Users can log in and out using the web interface.
3. Users can start a game.
 - a. Choice of difficulty.
 - b. Abort game.
4. User can view Highscores.
5. User can view his collected Achievements.
6. User can add/delete friends.
7. User can send messages to friends.
8. Authoring-tool for Users to edit levels and user data.
 - a. Adding/deleting levels
 - b. Adding deleting users

Concretizing

Using these use-cases as a basis, we drew up a diagram of how the different screens of the game would transition. This also served as a point of orientation for the overall interface design.



Technical Requirements

We further defined what the game would be like by establishing some technical requirements.

Gameplay

Break down input String into characters

Recognize, char by char, whether correct key is pressed

-> InputStroke object:

bool is_inputCorrect

Timestamp timePressed

Add each pressed key (char object) to ScoringText object.

Render ScoringText:

Print into scoringBox, char by char, in order of timePressed - timeStart

Determine display color via InputStroke.is_inputCorrect()

GameLogic

Timestamp timeStart

Keyboard

Key

Feedback

Visual

Camera movement / shakes

Audio

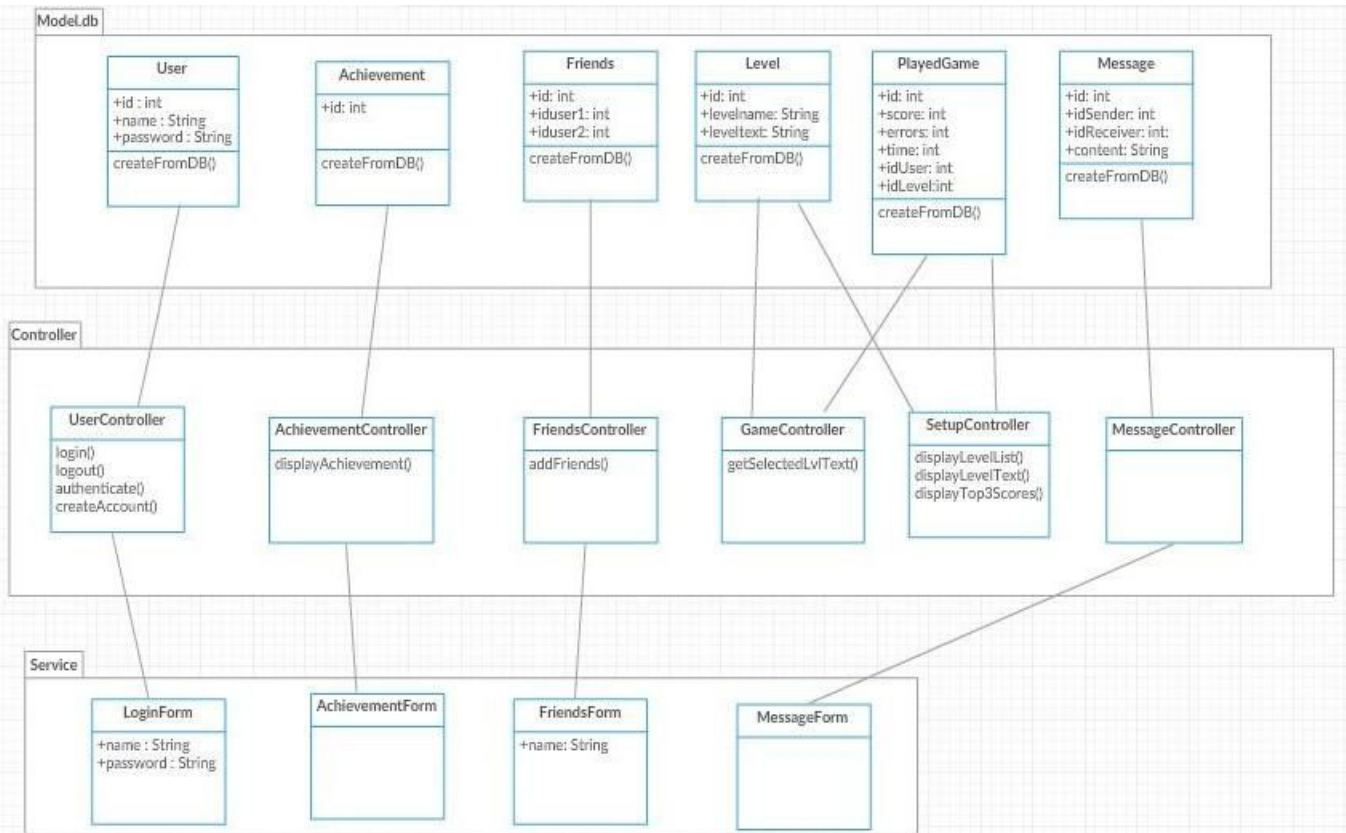
Sound on Stroke

Correct

Not Correct

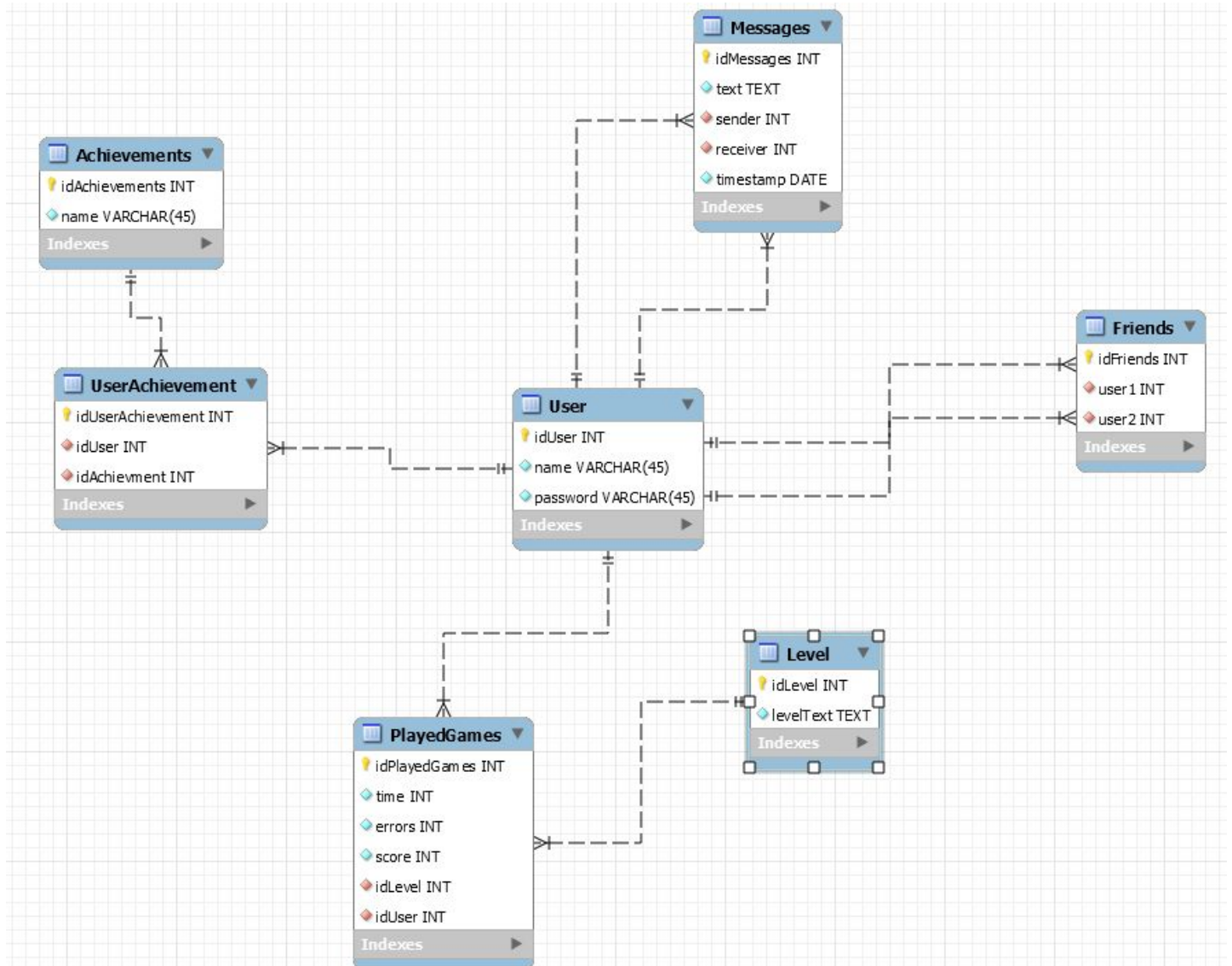
Combo increase

We also created a class diagram containing all classes and methods already defined in the concept stage:



Database setup

Our program needed to have a connection to a MySQL database to function. We first created an Entity Relationship Diagram to model the relationships between the individual tables that were needed for our database.



Final Preparations

As the basics of the software we were going to create were defined, we broke everything we had to do into individual tasks, further detailing the design.

JavaFX	<p>Level Creation</p> <ul style="list-style-type: none"> • Textfield LvL • Textbox LvL Name <p>User Manipulation</p> <ul style="list-style-type: none"> • Create User • Delete User •
Icons Bar	<p>Icon Profile</p> <ul style="list-style-type: none"> • New Window on Click, different content <p>Icon Friends</p> <p>Icon Logout</p>
Menu	<p>Login</p> <ul style="list-style-type: none"> • Textfield Name • Textfield PW • Button „Create New“ • Button „Login“ <p>Profile</p> <ul style="list-style-type: none"> • Box Achievements • Picture • Name • Stats <ul style="list-style-type: none"> ◦ Average Error Rate ◦ Average Score • Best Score <ul style="list-style-type: none"> ◦ Text Name ◦ Play Level Button • Worst Error Rate <ul style="list-style-type: none"> ◦ Text Name ◦ Play Level Button <p>Start/Play/Prep</p> <ul style="list-style-type: none"> • Play Button • Level Selection Scroll-List • Score current selected Level • Current selected Level Text <p>Friends & Chat</p> <ul style="list-style-type: none"> • Friends Scroll List <ul style="list-style-type: none"> ◦ Name ◦ Last Message • Chat <ul style="list-style-type: none"> ◦ History

	<ul style="list-style-type: none"> o Timestamps • Search Friend • Add Friend
Game	<p>Game Pause</p> <ul style="list-style-type: none"> • Current written text • Global Highscore for this Level • Friends Hiscore for this Level • Personal Best for this Level • Resume Button