

Softwareprojekt

Inhaltsverzeichnis

Software Idee.....	1
Grundidee.....	1
Spielmechaniken.....	1
Leveldesign.....	1
Datenbank.....	1
Besprechungsnotiz.....	2
Projekt: Rain or Shine oder Puddle Partners.....	4
Softwarespezifikation.....	6
Einführung.....	6
Beschreibung.....	6
Anforderungen.....	7
Stakeholder.....	7
Funktionale Anforderungen.....	7
Rahmenbedingungen/Systemanforderungen.....	7
Betriebsbedingungen.....	7
Qualitätsmerkmale.....	7
Graphische Benutzerschnittstelle.....	8
Anforderungen im Detail.....	9
User Stories.....	9
Technische Beschreibung.....	10
Systemübersicht.....	10
Softwarearchitektur.....	10
Technologiewahl.....	10
Schnittstellen.....	10
Ereignisse.....	10
Datenmodell.....	10
Abläufe.....	10
Entwurf.....	10
Fehlerbehandlung.....	11
Validierung.....	11
Projektorganisation.....	12
Annahmen.....	12
Verantwortlichkeiten.....	12
Grober Projektplan.....	12
Anhänge.....	14
Glossar.....	14
Referenzen.....	14
Index.....	14

Software Idee

Grundidee

- Kooperationsspiel
- Jump n run
- Puzzle
- Spielaufbau als Level
- Unity als Entwicklungsplattform
- 2D Side view

Spielmechaniken

- Verschiedene Wettermodi:
 - o Normaler Regen/Matsch Regen (Verringerung der Geschwindigkeit)
 - o Säure Regen (Schaden an Spielern)
 - o Frost Regen/Slide Regen (Charaktere rutschen und halten nicht direkt an)
- Regenschirm, um Wettereffekte zu verringern/negieren (jeweils nur von einem Charakter benutzbar)
- Flutung des Levels nach bestimmter Zeit
- Schrittzähler, damit die Wetterereignisse nicht „abgewartet“ werden, Wetteränderung erst ab einer gewissen Schritt Anzahl
- NPCs als zusätzlicher störender Faktor Schaden/Schubsen
- Items (Waffen, Tränke...)
- Interaktion zwischen den Spielern (hochheben, werfen...)
- Wind Mechanik (Speziell gegen den Regenschirm => Wegwehen)
- Regenschirm als Sammelbares Objekt, mit fester Nutzungszeit und eingeschränktem vorkommen pro Level
- Verschiedene schwierigkeitsstufen
- Fallen und Schalter
- Händler (Waffen, Tränke...)

Leveldesign

- Level Größe = ?
- Eigenes Level Design (kein zufälliger Level Aufbau)
- Versteckmöglichkeiten vor dem Regen? (Gelände)

Datenbank

- Scoreboard => Datenbank
- Spiel speichern => Datenbank/Datei?
- Spieler Namen => Datenbank?

Besprechungsnotizen

Besonderheiten Abhebung von der Konkurrenz:

- Neue Spielmechaniken und Level Ideen.

Anwendungsbereiche

- Entertainment

Was ist das Produkt nicht?

- Kein Action Kampfspiel (NPCs dezent angelegt und sollen bei Rätsel/Puzzle nicht groß stören)

Alleinstellungsmerkmal:

- Kein Realismus, Pixel Game, Regen, Mechaniken, die es in anderen Koop-Spielen nicht gibt.

Konkurrenz:

- Andere Koop Spiele:
- Portal 2
- We were here
- It Takes two
- A way out
- Unravel two

Anwendungsbereiche:

- Ein Spiel, das nur zu zweit gespielt werden kann und das alleine nicht funktioniert.

Marktanforderungen:

- Marktanteilsverlust, da das Spiel nicht allein gespielt werden kann.

Wie kann durch das Produkt Geld verdient werden:

- In game Goodies durch Shop, Werbung, Spielverlauf

Für wen ist das Produkt/der Service

- Zielgruppe: Spieler/Publisher (Steam, Epic Games)

Bedürfnisse:

- Für Menschen die etwas zu zweit spielen möchten

Zielgruppenmerkmale:

- Gängige Spielsteuerung Vertrautheit, ansonsten keine

Lohnt sich das Projekt?

- Weil es etwas neues ist und der Markt noch nicht ausgeschöpft ist.

Aufwand:

- Grafisch simpel
- Frei für den Markt

Risiken:

Das Spiel verkauft sich schlecht, wenige Spieler

Ein Größeres Unternehmen veröffentlicht zeitnahe auf ein Koop Spiel

Probleme mit OS oder Hardware

Softwarespezifikation

1 Einführung

Ein Level basiertes Kooperationsspiel mit Rätsel/Puzzle-Mechaniken. Die Spieler müssen sich durch verschiedene Level kämpfen, es gibt Wetterereignisse sowie feindliche gesinnte NPCs, die die Spieler am Fortschritt behindern. Durch Abschluss aller Level oder dem vorzeitigen Ende (Zeit abgelaufen/Spieler-Tod) wird ein Punktwert ermittelt und am Scoreboard veröffentlicht.

1.1 Kurzbeschreibung formal:

Ein Level basiertes Kooperationsspiel mit Rätsel/Puzzle-Mechaniken. Die Spieler müssen sich durch verschiedene Level kämpfen, es gibt Wetterereignisse sowie feindliche gesinnte NPCs, die die Spieler am Fortschritt behindern. Durch Abschluss aller Level oder dem vorzeitigen Ende (Zeit abgelaufen/Spieler-Tod) wird ein Punktwert ermittelt und am Scoreboard veröffentlicht.

1.2 Kurzbeschreibung nicht formal für Zielgruppe:

Als Spieler müsst Ihr euch in verschiedenen Leveln dem unbeugsamen Wetter und starken Feinden stellen. Löst die Rätsel und Puzzle so schnell Ihr könnt, braucht Ihr zu lange, werdet ihr „untergehen“. Je weiter Ihr kommt, desto höher euer Team Punktzahl auf dem Scoreboard!

1.3 Beschreibung

Geplant ist ein 2D side view Pixel Kooperations Spiel in dem es darum geht, in möglichst kurzer Zeit zu zweit Rätsel und Puzzles zu lösen. Das Spiel wird Level-basiert aufgebaut sein, sodass die Schwierigkeit mit steigendem Level stetig ansteigt. Zu den Besonderheiten unseres Kooperations Spiels zählt:

- Regen mit unterschiedlichen negativen Auswirkungen
- NPCs die die Spieler beim Lösen der Rätsel und Puzzles stören
- Ein Regenschirm um den Wetterbedingungen zu trotzen, das jeweils nur ein Spieler benutzen kann, aber beide darunter geschützt sind
- Damit die Wetterereignisse nicht abgewartet werden wird es eine Zeiteinschränkungen geben, wenn die Zeit abgelaufen ist wird das Level geflutet und das Spiel ist beendet

Das Spiel wird kein explizites Kampfsystem erhalten, dennoch haben wir vor, den Spielern Lebenspunkte zu geben. Die Lebenspunkte werden zum größten Teil durch Wetterbedingungen und Level Ereignisse beeinflusst. Die Rätsel und Puzzles werden so ausgelegt, dass diese nur zu zweit gelöst werden können. Am Ende des Spiels wird ein Punktestand ermittelt und auf einer Bestenliste veröffentlicht.

1.4 Risiken

Unsere direkten Konkurrenten sind andere Kooperations Spiele wie:

- Portal 2, We were here, It Takes two, A way out, Unravel two

Sollte ein anderes größeres und bekanntes Entwicklerstudio zur gleichen Zeit ein neues Kooperationsspiel auf den Markt bringen, wäre dies unser größtes Einnahme-Risiko.

Entwicklungsprobleme zählen auch zu den Risiken einer Software:

- Stürzt ab
- Spielfehler
- Zu hohe Systemanforderungen

können und werden dafür sorgen, dass unsere Kunden von unserem Produkt enttäuscht sind und das Vertrauen verlieren, um dies zu vermeiden, ist eine sorgfältige und gründliche Planung vonnöten.

Der Entwicklungsaufwand und das Risiko unseres Produkts wird durch die Art des Spiels minimiert, unser Ziel ist es nicht, ein Spiel zu entwerfen, das möglichst realistisch und visuell anspruchsvoll gestaltet ist, deshalb konzentrieren wir uns auf die wesentlichen Inhalte des Produkts und entwerfen die Anwendung als 2D und nicht als 3D Spiel.

Wir möchten unser Produkt auf verschiedenen Vertriebsplattformen für einen Festpreis anbieten, mit steigendem Erfolg können noch weitere Spiele Inhalte (DLCs) oder Shops für Echtgeld implementiert werden, um den Umsatz zu steigern.

2 Anforderungen

2.1 Stakeholder

Spieler:

Setzt sich mit der Software auseinander (spielt) und möchte ein fehlerfreies aufregendes Spielerlebnis genießen.

Investoren/Teilhaber:

Sind auf Profit aus und erwarten ein Produkt, das die Spieler kaufen möchten und finanzieren das Produkt.

2.2 Funktionale Anforderungen

2.2.1 Use-Case-Diagramme

Spieler möchte:	Funktion benötigt:
Figur bewegen	Eingabeerkennung + Spielphysik
Objekte greifen	Fixieren des Objekts am Spieler
Objekte werfen	Objekt in Blickrichtung beschleunigen
Türen öffnen/Geräte benutzen	Aktivierbare Schaltflächen
Regen abwehren	Tragbarer Regenschirm
Zu zweit spielen	Kontinuierliche Netzwerkverbindung und Synchronisation der Aktionen
Rätsel lösen	Trigger zum Abschließen des Levels, Datenbankverbindung zum Speichern des Punktestands

2.3 Nicht-funktionale Anforderungen

2.3.1 Rahmenbedingungen/Systemanforderungen

Client:

Betriebssystem: Windows 10 und höher, Mac, Linux

Prozessor: -

Grafikkarte: DX11 fähige Grafikkarte

Arbeitsspeicher: -

Speicherplatz: -

Netzwerkanbindung: DSL

DB Server:

Betriebssystem: Linux Ubuntu 24.04 LTS, Debian Bookworm 32/64 Bit (Raspberry PI 3 B),

Debug unter Windows 10/11

Prozessor: 1,2 GHz Quadcore

Arbeitsspeicher: 1 GB

Speicherplatz: 19 KB (Skripte), 150 MB (Webserver)

Netzwerkanbindung: Abhängig von Spieleranzahl

Software: Apache Webserver (Version 2.4.58), MariaDB (10.4.32), PHP (Version 8.2.12),
phpMyAdmin (Version 5.2.1)

2.3.2 Betriebsbedingungen

Programmiersprache C#

Angeboten als Native App, Desktop Applikation.

2.3.3 Qualitätsmerkmale

- Qualitätsmerkmal	Sehr gut Gut Normal Nicht relevant
- Fehlertoleranz	- X - -
- Wiederherstellbarkeit	- - - X
- Ordnungsmäßigkeit	- X - -
- Richtigkeit	- X - -
- Konformität	- - - X
- Installierbarkeit	X - - -
- Verständlichkeit	X - - -
- Erlernbarkeit	X - - -
- Bedienbarkeit	X - - -
- Zeitverhalten	- - X -
- Effizienz	X - - -

2.4 Graphische Benutzerschnittstelle

- Die gesamte Anwendung ist eine Benutzerschnittstelle.

Web Server Einrichtung

Installation Linux (Ubuntu 24.04 LTS) / Debian Bookworm:

- Port 80 auf dem Router freigeben (TCP/UDP)
- Konsole öffnen
- Den Befehl "sudo apt install apache2" ausführen
- Den Befehl "sudo apt install php libapache2-mod-php" ausführen
- Für Ubuntu den Befehl "sudo apt install mysql-server" ausführen für Debian Bookworm "sudo apt install mariadb" ausführen
- Den Befehl "sudo systemctl start mysql.service" ausführen
- Den Befehl "sudo apt install phpmyadmin" ausführen (Dialog 1: apache2 auswählen, Dialog 2: Ja, Dialog 3: Passwort setzen)
- Den Befehl "sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf" ausführen
- Den Befehl "sudo a2enconf phpmyadmin.conf" ausführen
- Den Befehl "sudo systemctl reload apache2.service" ausführen
- Der Web- und Datenbankserver ist nun vollständig installiert
- In der Konsole den Befehl "sudo mysql -u root" ausführen
- Datenbank Befehl "ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'yourpassword';" ausführen
- Den Befehl "sudo chmod 777 /var/www/html" ausführen (Schreibrechte)
- Den Ordner unter "/var/www/html" öffnen und alle Dateien entfernen

- Im Ordner "/var/www/html" einen neuen Ordner mit dem Namen "puddle_partners" anlegen und die PHP Dateien aus dem Ordner "files" in den Ordner "puddle_partners" kopieren
- Im Webbrowser die Seite "localhost/phpmyadmin/" aufrufen
- Mit "root" und "yourpassword" anmelden
- Neue leere Datenbank puddle_partners anlegen
- Klicken Sie auf die Datenbank "puddle_partners", klicken Sie oben in der Menüleiste der Datenbank "puddle_partners" auf "Importieren", unter "Zu importierende Datei" auf "Datei auswählen" klicken, wählen Sie die Datei "puddle_partners.sql" aus, anschließend unten rechts auf den Button "OK" klicken

Tests Linux:

- Nach der erfolgreichen Installation befinden sich die Test PHP Skripte im Ordner "var/www/html/puddle_partners/test/"
- Starten Sie das Skript test_register.php in dem Sie in der Browser Adresszeile die Webseite "localhost/puddle_partners/test/test_register.php" aufrufen
- Starten Sie das Skript test_login.php in dem Sie in der Browser Adresszeile die Webseite "localhost/puddle_partners/test/test_login.php" aufrufen

Installation Windows (nicht empfohlen):

- Port 80 auf dem Router freigeben (TCP/UDP)
- Port 80 als Firewall Ausnahme konfigurieren (wird bei der installation von Apache Web Server automatisch abgefragt)
- Xampp 8.2.12 herunterladen und installieren (Apache Webserver, PHP, phpMyAdmin, der Rest wird nicht benötigt)
- In den Xampp Installationsordner gehen "C:\Xampp\" den Ordner "htdocs" öffnen und alle Dateien entfernen
- Im Ordner "htdocs" einen neuen Ordner "puddle_partners" anlegen und die PHP Dateien aus dem Ordner "files" in den Ordner "puddle_partners" kopieren
- Xampp Controlpanel öffnen und Apache Webserver und MySQL starten
- Im Xampp Controlpanel unter MySQL auf den Button "Admin" klicken, es öffnet sich der Webbrowser => phpMyAdmin
- Neue leere Datenbank puddle_partners anlegen
- Klicken Sie auf die Datenbank "puddle_partners", klicken Sie oben in der Menüleiste der Datenbank "puddle_partners" auf "Importieren", unter "Zu importierende Datei" auf "Datei auswählen" klicken, wählen Sie die Datei "puddle_partners.sql" aus, anschließend unten rechts auf den Button "OK" klicken

Tests Windows:

- Nach der erfolgreichen Installation befinden sich die Test PHP Skripte im Ordner "C:\xampp\htdocs\puddle_partners\test/"
- Starten Sie das Skript test_register.php in dem Sie in der Browser Adresszeile die Webseite "localhost/puddle_partners/test/test_register.php" aufrufen
- Starten Sie das Skript test_login.php in dem Sie in der Browser Adresszeile die Webseite "localhost/puddle_partners/test/test_login.php" aufrufen

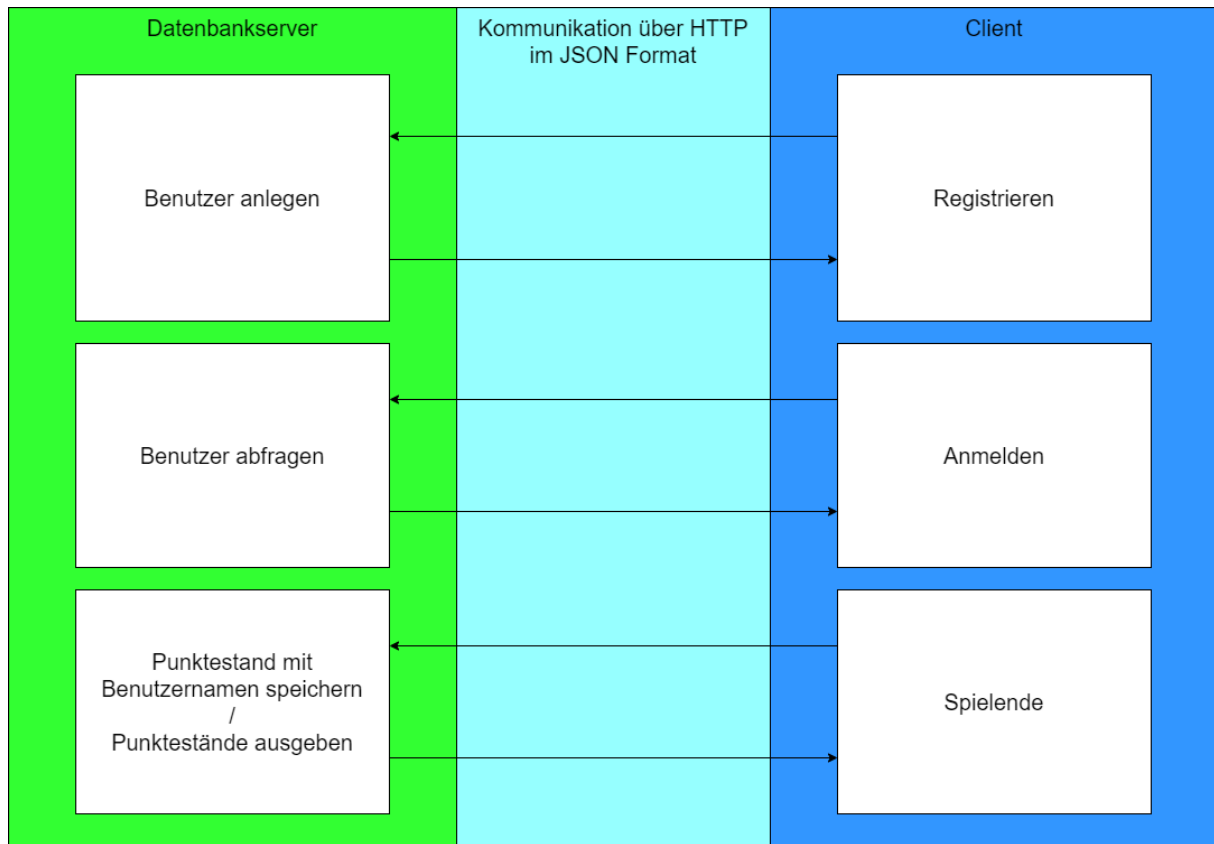
2.5 Anforderungen im Detail

User Stories

Als	möchte ich	sodass	Akzeptanz
Spieler	die möglichkeit haben das scoreboard zu sehen	ich mich mit anderen vergleichen kann	Scoreboard anzeigen
Spieler	meinen Partner sabotieren	wir, oder nur ich was zu lachen habe	Spieler können sich gegenseitig sabotieren
Spieler	mit meinem Partner zusammen Herausforderungen meistern	wir dieses Erfolgserlebnis zusammen teilen können	Level nur in kooperation abschließbar
Spieler	einen flüssigen Spielfluss haben	das erreichen des Highscores Spaß macht	Flüssige Steuerung und Reaktionsschneller Spielfluss
Spieler	kreative Puzzles lösen	Das Spielen viel Spaß macht	kreative Ideen für Hindernisse
Spieler	mit den Puzzlen auf die Probe gestellt werden	Ich mich gut fühle, wenn ich die Lösung finde	Anspruchsvolle Level haben (Low Prio)
Spieler	dass das Spiel flüssig läuft	Das Spielen nicht frustrierend ist	effiziente Implementierung
Spieler	mit meiner Umgebung interagieren	ich Puzzle auf verschiedene Arten lösen kann	Greifmechanik und Kollisions- Bedingungen einbauen
Spieler	einen Timer oder alternativen Indikator für die verbleibende Zeit des Levels sehen	ich abschätzen kann, ob ich mich beeilen muss, das Rätsel zu lösen	Zeitablauf darstellen
Investor	ein gutes Spielerlebnis für den Spieler	das Spiel gut bewertet und öfter gekauft wird	Spieler ist zufrieden
Investor	möglichst niedrige Systemanforderungen	auch Spieler mit alter Hardware das Spiel kaufen können	effiziente Implementierung

3 Technische Beschreibung

3.1 Systemübersicht



3.2 Softwarearchitektur

Die gesamte Software läuft auf dem System des Benutzers, eine Kommunikation findet beim Registrieren, Anmelden und am Spielende mit einem Datenbankserver für das Hinterlegen der erreichten Punkte und das Anzeigen der Platzierung im Scoreboard statt.

3.2.1 Technologiewahl

Unity (2022.3.26f1) als Entwicklungsplattform, C#, JSON

MySQL (MariaDB) als Datenbank Server, PHP als zusätzliche Sicherheitsschicht

3.3 Schnittstellen

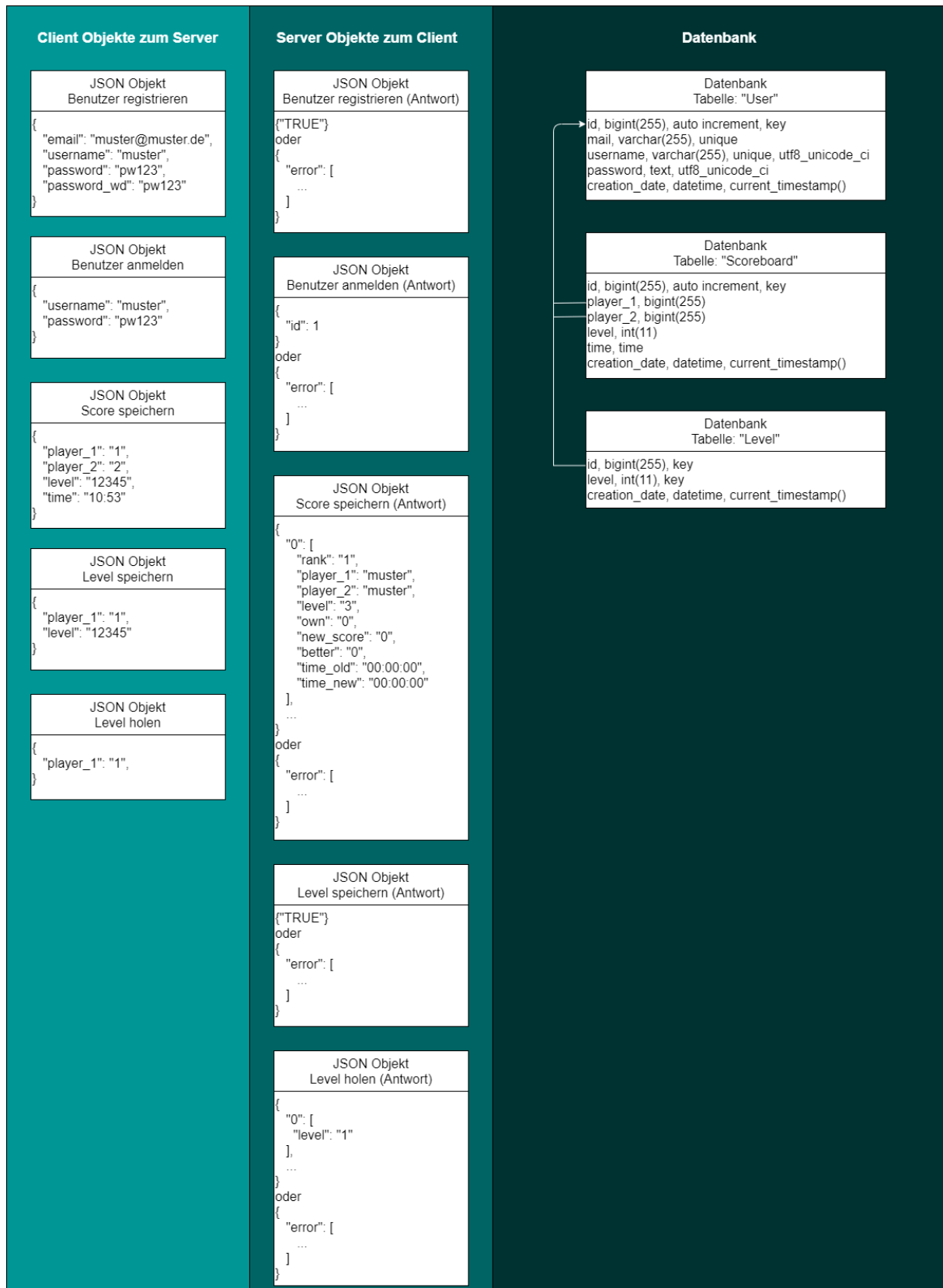
Intern:

- UnityEngine.UI, um das User Interface zu verwalten.
- UnityEngine.SceneManagement, um die Levels während der Laufzeit zu verwalten.
- Netcode for GameObjects
- Multiplayer Tools

Extern:

- Multiplayer Samples Utilities

3.4 Datenmodell



3.7 Fehlerbehandlung

- NullReferenceException
- OutOfMemoryException
- Web Server Timeout
- Web Server Errors

4 Projektorganisation

4.1 Annahmen

Technologien:

C#, Unity

Git:

Repositories nach Meilensteinen

Entwicklungsumgebung:

Unity (2022.3.26f1), Visual Studio

Erweiterungen:

Ingame Shop mit Skins und DLCs (herunterladbare Inhalte)

4.2 Verantwortlichkeiten

Pro Meilenstein

Pro Meilenstein-Unteraufgabe je zwei Personen, wenn möglich ("Pair programming")

Rollen:

Product-Owner: (nicht notwendig für unser Projekt)

Scrum-Master: Manuel Wiebe

Softwarearchitekt: (nicht notwendig für unser Projekt)

Frontend-Entwickler: Michael Herber, Fabian Pechta, Alexey Khokhlov, Manuel Wiebe, Abdul-Kerim Gerikhanov

Backend-Entwickler: Alexey Khokhlov (Web Server Skripte, Web Server Rückmeldungen, Web Server Einrichtung)

DevOps-Engineer: Alexey Khokhlov, Manuel Wiebe

Github Accountnamen	Reale Namen
GoAlldex	Alexey Khokhlov
fpechta	Fabian Pechta
Skrpop	Abdul-Kerim Gerikhanov
IXetnosl	Manuel Wiebe
Adonski	Michael Herber

4.3 Grober Projektplan

KW 17:

- Projektidee Entwurf
- Grobe Sammlung von Features der Anwendung

KW 18:

- Erstellung Softwarespezifikation
- Team strukturieren

KW 19:

- Einarbeitung in Unity
- Erste Grundkarte (ohne feste Objekte)
- Zwei Bewegliche Objekte (Spieler)

KW 20:

- Hindernisse (Bäume, Wände...)
- NPCs mit movement (ohne einflüsse auf Spieler)
- Texturen/Objekte zeichnen/suchen

KW 21:

- Wetterereignisse

KW 22:

- Erstes Rätsel/Puzzle
- NPCs mit störfaktor

KW 23-25:

- Tests
- Weitere Level
- Spielende

KW 26:

- Datenbank (MySQL)
- Punktestand ermitteln/anzeigen
- Registrier/Login Interface

KW 27-28:

- Feintuning
- Wenn Zeitlich möglich portierung auf Mobile
- Abschluss des Projekts

5 Anhänge

5.1 Glossar

DLC = Downloadable Content / Herunterladbare Inhalte

NPC = Nicht Spieler Charakter

5.2 Referenzen

Impressum

Icon "Schlüssel" registrier Formular:

https://www.flaticon.com/de/kostenloses-icon/schlüssel_680415

Icon "Eintreten" login Formular:

https://www.flaticon.com/de/kostenloses-icon/eintreten_8677615

5.3 Index