

Projektdokumentation

Projektname: ProductivityGarden

Name

Jonas Huber

Matrikelnummer

IU14085128

Modul

Projekt: Software Engineering

DLMCSPSE01_D

Datum

25.01.2025

Inhalt

1. Projektbeschreibung	2
2. Benutzeranleitung	3
Installation und notwendige Ressourcen	3
Bedienung und Oberfläche	3
Hauptfenster (session.py)	3
Virtueller Garten (virtualgardens.py)	5
Anwendung beenden oder wechseln:	5
Datenpersistenz	6
3. Risikomanagement	7
Potenzielle Risiken	7
Technische Risiken	7
Benutzerbezogene Risiken	7
Sicherheitsrisiken	8
Projektbezogene Risiken	8
Bewertungstabelle	9
4. Zeitplanung	10
5. Literaturverzeichnis	11

Abbildungsverzeichnis

Abbildung 1 - Gantt-Diagramm	10
Abbildung 2 -Gantt-Diagramm aktualisiert	10

1. Projektbeschreibung

Im modernen Alltag sehen sich Menschen häufig mit der Herausforderung konfrontiert, sich zu konzentrieren und ihre Zeit effektiv zu managen. Studien belegen, dass Gamification einen positiven Einfluss auf Motivation und Produktivität ausübt (Matallaoui, 2016, S. 4). Gamification, auch Spielifizierung genannt, bezeichnet die Anwendung spielerischer Elemente und Mechanismen in einem nicht-spielerischen Kontext, beispielsweise im Bildungsbereich (Huseynli, 2024, S. 45). Die Integration eines flexiblen Pomodoro-Timers, eines klassischen Timers sowie einer Stoppuhr ermöglicht es den Nutzer:innen, ihre produktiven Phasen präzise zu planen und nachzuverfolgen. Dies erfolgt in einer entspannenden und spielerischen Umgebung, wodurch eine nachhaltige Steigerung der Produktivität gefördert und gleichzeitig eine Reduktion von Stress erzielt werden soll. Zur Optimierung des Zeitmanagements besteht die Möglichkeit, die aufgewendete Zeit einzelnen Projekten zuzuweisen.

Ein wesentlicher Bestandteil der Anwendung ist ein Belohnungssystem, bei dem Nutzer:innen für produktive Zeit Punkte sammeln können. Diese Punkte können dazu verwendet werden, virtuelle Gärten mit Pflanzen und Dekorationen zu gestalten. Die Nutzer:innen haben die Möglichkeit, mehrere Gärten anzulegen und individuell anzupassen. Dies fördert nicht nur die Motivation, sondern auch Entspannung und Kreativität.

Die Anwendung ist für das Betriebssystem Windows optimiert und bleibt im Hintergrund aktiv, um eine nahtlose Nutzung zu ermöglichen. Die Speicherung sämtlicher Daten erfolgt lokal, um die Privatsphäre der Nutzer:innen zu schützen. Die Oberfläche verzichtet auf ablenkende Elemente, um den Nutzer:innen eine ungestörte Konzentration auf das Spielerlebnis und die Produktivität zu ermöglichen. ProductivityGarden ist werbefrei und bietet keinen Zugang zu Paywalls oder Premium-Inhalten.

Im Gegensatz zu herkömmlichen Apps zur Prokrastinationskontrolle richtet sich ProductivityGarden an eine breite Zielgruppe und ist für alle geeignet, die ihr Zeitmanagement verbessern möchten, eine zugleich entspannende und kreative Herangehensweise.

2. Benutzeranleitung

Installation und notwendige Ressourcen

Um die Anwendung zum Laufen zu bekommen helfen:

1. Repository von GitHub pullen:
https://github.com/Xecu114/iu_pse
2. "README" lesen und beachten
3. Python Version checken: kompatibel ab 3.9
4. Es werden zwei weitere Python-Pakete benötigt, damit die Anwendung ausgeführt werden kann:

- a. PyQt6
- b. Pygame

Diese können auch mit Hilfe des „install_py_libs.bat“ Skripts automatisch installiert werden.

5. Anwendung starten:
`python main.py`

Bedienung und Oberfläche

Anwendung starten über „main.py“ (z.B. mit: `python main.py`)

Damit öffnet sich das Hauptfenster (in den weiteren Dokumenten auch oft als „Produktivanwendung“ bezeichnet):

Hauptfenster (session.py)

Das Hauptfenster ist in drei Spalten aufgeteilt:

Erste Spalte

- Punktestand (zwei Kreise mit „available“ und „total“ Punkten).
- Button „TO THE GARDENS“: Öffnet den virtuellen Garten in einem separaten Fenster/Prozess.
- Darunter ein Bild von einer Blumenwiese

Zweite Spalte

- Zeitmanagement-Bereich mit Timer-Modus und digitaler Uhr
- Steuerknöpfe: Start, Pause/Resume, Stop
- Eingabefelder für Pomodoro-Arbeitszeit, -Pausenzeit, sowie klassischen Timer
- Pomodoro-Mode (Voreinstellung: 25 Min. Arbeit, 5 Min. Pause)
- Ein Button „Switch to Timer“, „Switch to Stopwatch“ oder „Switch to Pomodoro“ zum Wechseln zwischen den verschiedenen Zeitmess-Modi
- Fehlermeldungen bei falscher Zeiteingabe (bspw. Zeit < 1 Min. oder > 24 Std.).
- Textfeld im unteren Teil für Notizen, Aufgaben etc.

Dritte Spalte

- Projektverwaltung: Auswahl eines Projekts (Drop-down-Menü), Hinzufügen und Löschen von Projekten.
- Bearbeiten: Name, Beschreibung, Kategorie, Start- und End-Datum eines Projekts.
- Kreis mit getrackter Zeit (Minuten) für das aktuell ausgewählte Projekt.
- Kreisdiagramm (Pie-Chart) aller Projekte und deren getrackter Gesamtzeit.
- Option zum manuellen Hinzufügen von Zeit in Minuten (z.B. wenn man offline gearbeitet hat).

Funktionsweise:

Timer/Stopwatch/Pomodoro:

- Beim Starten (Start-Button) läuft die gewählte Zeitmessung.
- Bei Pause kann man den Timer anhalten, Resume führt ihn weiter. Stop setzt alles zurück.
- Jede getrackte Produktivzeit addiert sich intern in minute_counter. Alle 10 gesammelten Minuten gibt es automatisch 1 Punkt (erhältlich im „available points“-Kreis).
- Die Zeit wird dem aktuell ausgewählten Projekt in der Projektauswahl gutgeschrieben.

Projekte:

- Über den Add-Button wird ein neues Projekt in der lokalen Datenbank (sqlite) und im Drop-down-Menü angelegt.
- Delete entfernt das aktuell ausgewählte Projekt aus der Datenbank.
- Im Drop-down-Menü kann man zwischen Projekten wechseln.
- Die Felder Name, Description, Category, Start/End Date sind editierbar und werden regelmäßig in die Datenbank geschrieben.
- Das Kreisdiagramm und die Zeit-Anzeige werden automatisch aktualisiert.

Punkte:

- Neue Punkte erscheinen in „available“, gleichzeitig erhöht sich auch „total“.
- In der JSON-Datei (JSON_FILE) wird der Punktestand regelmäßig gespeichert (siehe Abschnitt Datenpersistenz).

Virtueller Garten (virtualgardens.py)

Die Spielkomponente, also die virtueller Garten Anwendung wird über den Button „TO THE GARDENS“ im Hauptfenster gestartet.

Alternativ besteht auch die Möglichkeit die Anwendung manuell zu starten mit bspw. folgendem Befehl: `python virtualgardens.py`

Beim Start wird ein Pygame-Fenster geöffnet und ein Hauptmenü dargestellt mit folgenden Optionen:

- „Create Garden“: Neues Garten-Layout anlegen. Hier wählst du:
 1. Vegetationsart (z.B. City Park, Desert, Rainforest)
 2. Einen Namen für den Garten
 3. Dann wird eine leere Karte erzeugt, die du bepflanzen kannst.
- „Load Garden“: Vorhandenes .map-File aus dem Ordner map_data/ laden. Du kannst so einen bereits angelegten Garten weiterbearbeiten.
- „Back to Productivity Window“: Wechselt zurück zum Hauptfenster

Gärten und Objekte

Die Gärten sind in Kacheln unterteilt. Du hast eine Inventarleiste am unteren Bildschirmrand, in der du verschiedene Objekte (Blumen, Bäume, Wege, etc.) auswählen kannst.

Über Mausklick auf ein Kachel-Feld wird das aktuell gewählte Objekt platziert, sofern genug Punkte vorhanden sind.

Jeder Gegenstand hat Kosten (z.B. 2 Punkte für Blumen, 8 für einen großen Baum). Beim Platzieren eines Objekts werden Punkte von deinem „verfügbaren Punktestand“ abgezogen.

Tipp: Falls du nicht genug Punkte hast, um ein Objekt zu platzieren, kannst du zurück ins Hauptfenster, dir durch weitere Arbeitsminuten Punkte verdienen und erneut in den Garten wechseln.

Anwendung beenden oder wechseln:

In egal welchem Fenster kann über das Windows Fenster x-Symbol die Anwendung komplett beendet werden.

Über die zwei Buttons „TO THE GARDENS“ (Hauptfenster) oder „Back to Productivity Window“ (virtueller Garten) kann beliebig häufig zwischen den beiden Teilanwendungen gewechselt werden

Zusätzlich gelangst du in der virtueller Garten Anwendung jederzeit über die „Escape“ Taste zurück ins Hauptmenü der virtueller Garten Anwendung.

Datenpersistenz

Alle Daten werden lokal gespeichert im Ordner „resources“ und werden an keinen Server übermittelt.

Speicherung der Nutzerdaten:

Die Anwendung speichert fortlaufend folgende Daten:

- Punktestände (total_points, available_points)
- Timer-Einstellungen (z.B. Zeitwerte für Pomodoro-Phasen und normalen Timer)
- Freitext (das große Textfeld).
- Informationen der angelegten Projekte (Name, Beschreibung, Kategorie, Start- & Enddatum, gesammelte Minuten)

Speicherung der Gartendaten:

Beim Schließen eines Gartens oder der Anwendung werden folgende Daten gespeichert:

- Objekt-Positionen (in .map-Dateien)
- Garten-Metadaten (z.B. welche Vegetationsart) in gardens_data.json
- Ebenso wird der verbleibende Punktestand in deiner JSON_FILE (z.B. user_data.json) aktualisiert.

3. Risikomanagement

Potenzielle Risiken

Bei der Risikoanalyse werden potenzielle Risiken identifiziert, deren Eintrittswahrscheinlichkeit und Schadensausmaß bewertet und geeignete Gegenmaßnahmen vorgeschlagen. Die verschiedenen Risiken werden für mehr Übersichtlichkeit in Bereiche kategorisiert.

Technische Risiken

1. Leistungsprobleme bei hoher Last oder Datenmengen
 - Beschreibung: Die Anwendung könnte bei intensiver Nutzung oder Speicherung vieler Daten langsam werden, was die Benutzerfreundlichkeit beeinträchtigt.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Hoch (Nutzer:innen könnten das Programm frustriert abbrechen)
 - Gegenmaßnahmen:
 - Nutzung performanter Python-Bibliotheken und Implementierung von Optimierungsmöglichkeiten (Recherche benötigt)
 - Optimierung der Datenbankzugriffe und Minimierung redundanter Prozesse.
 - Vorab Lasttests durchführen.
2. Datenverlust durch unerwartetes Beenden der Anwendung
 - Beschreibung: Fortschritte oder Einstellungen der Nutzer:innen könnten bei einem Absturz verloren gehen.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Hoch
 - Gegenmaßnahmen:
 - Regelmäßige Autosave-Funktion implementieren.
 - Backup-Mechanismen für kritische Daten einbauen.

Benutzerbezogene Risiken

3. Mangelnde Akzeptanz der Benutzeroberfläche
 - Beschreibung: Die minimalistische Oberfläche oder die Visualisierung des virtuellen Gartens könnten als unattraktiv empfunden werden.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Mittel
 - Gegenmaßnahmen:
 - Feedback einholen und iterativ verbessern.
4. Fehlende Motivation zur Nutzung durch schlechte Umsetzung der Gamification
 - Beschreibung: Nutzer:innen könnten die Anwendung nicht regelmäßig verwenden, da die Gamification nicht motivierend genug ist.
 - Eintrittswahrscheinlichkeit: Mittel
 - Schadensausmaß: Hoch
 - Gegenmaßnahmen:
 - Belohnungssystem abwechslungsreich gestalten (z. B. freischaltbare Inhalte, Fortschrittsanzeige).

- Regelmäßige Erfolgserlebnisse integrieren, die den Nutzer:innen einen Nutzen aufzeigen.

Sicherheitsrisiken

5. Datenschutzverletzungen

- Beschreibung: Speicherung personenbezogener Daten könnte Sicherheitslücken enthalten oder DSGVO-Vorgaben nicht vollständig erfüllen.
- Eintrittswahrscheinlichkeit: Niedrig
- Schadensausmaß: Sehr hoch (rechtliche Konsequenzen, Vertrauensverlust)
- Gegenmaßnahmen:
 - Speicherung ausschließlich lokal durchführen.
 - Anonyme Datenerfassung sicherstellen.
 - Code- und Sicherheits-Audits vor Veröffentlichung.

Projektbezogene Risiken

6. Zeitüberschreitung oder Nichterfüllung von Anforderungen

- Beschreibung: Das Projekt könnte mehr Zeit als geplant in Anspruch nehmen oder einige Features könnten nicht rechtzeitig fertiggestellt werden.
- Eintrittswahrscheinlichkeit: Hoch
- Schadensausmaß: Hoch
- Gegenmaßnahmen:
 - Projektplanung in Milestones aufteilen.
 - Priorisierung der wichtigsten Features vornehmen.

7. Unerwartete technische Schwierigkeiten (z. B. mit Python-Bibliotheken)

- Beschreibung: Verwendete Bibliotheken könnten Bugs enthalten oder nicht die gewünschten Funktionen bieten.
- Eintrittswahrscheinlichkeit: Mittel
- Schadensausmaß: Mittel
- Gegenmaßnahmen:
 - Vorab sorgfältige Auswahl der Bibliotheken treffen.
 - Alternativen evaluieren und Backup-Lösungen bereithalten.

Bewertungstabelle

Tabelle 1 fasst die identifizierten Risiken zusammen, indem sie diese nach Eintrittswahrscheinlichkeit und Schadensausmaß kategorisiert. Auf Basis dieser Bewertung wird die Priorität der Maßnahmen abgeleitet, um die wichtigsten Risiken gezielt zu adressieren.

Risiko	Eintritts- wahrscheinlichkeit	Schadensausmaß	Maßnahmenpriorität
Leistungsprobleme	Mittel	Hoch	Hoch
Datenverlust	Mittel	Hoch	Hoch
Mangelnde Akzeptanz der UI und Visualisierung	Mittel	Mittel	Mittel
Fehlende Nutzungsmotivation	Hoch	Hoch	Hoch
Datenschutzverletzungen	Niedrig	Sehr hoch	Hoch
Nutzer-Manipulation (Cheating)	Mittel	Mittel	Mittel
Zeitüberschreitung	Hoch	Mittel	Hoch
Technische Schwierigkeiten	Mittel	Mittel	Mittel

Tabelle 1 – Risikoanalyse-Bewertungstabelle

4. Zeitplanung

Bei der Zeitplanung sind einige Kriterien zu berücksichtigen. Zum einen ist es schwierig, den Aufwand für die einzelnen Aufgaben im Voraus abzuschätzen, zum anderen ist es schwierig abzuschätzen, wie viel Zeit an welchen Tagen zur Verfügung steht. Daher ist es wichtig, die Zeitplanung so zu gestalten, dass der aktuelle Fortschritt ständig mit den Zielen verglichen werden kann und gegebenenfalls kleine Anpassungen vorgenommen werden können.

Deswegen wird die Projektzeitplanung kontinuierlich mithilfe eines Gantt-Diagramms überwacht, das zugleich zur Fortschrittskontrolle dient. Dieses ist in Abbildung 1 zu sehen.

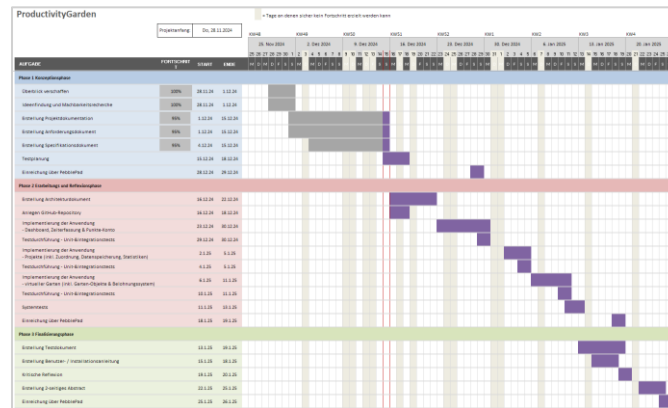


Abbildung 1 - Gantt-Diagramm

Stand 25.01.2025:

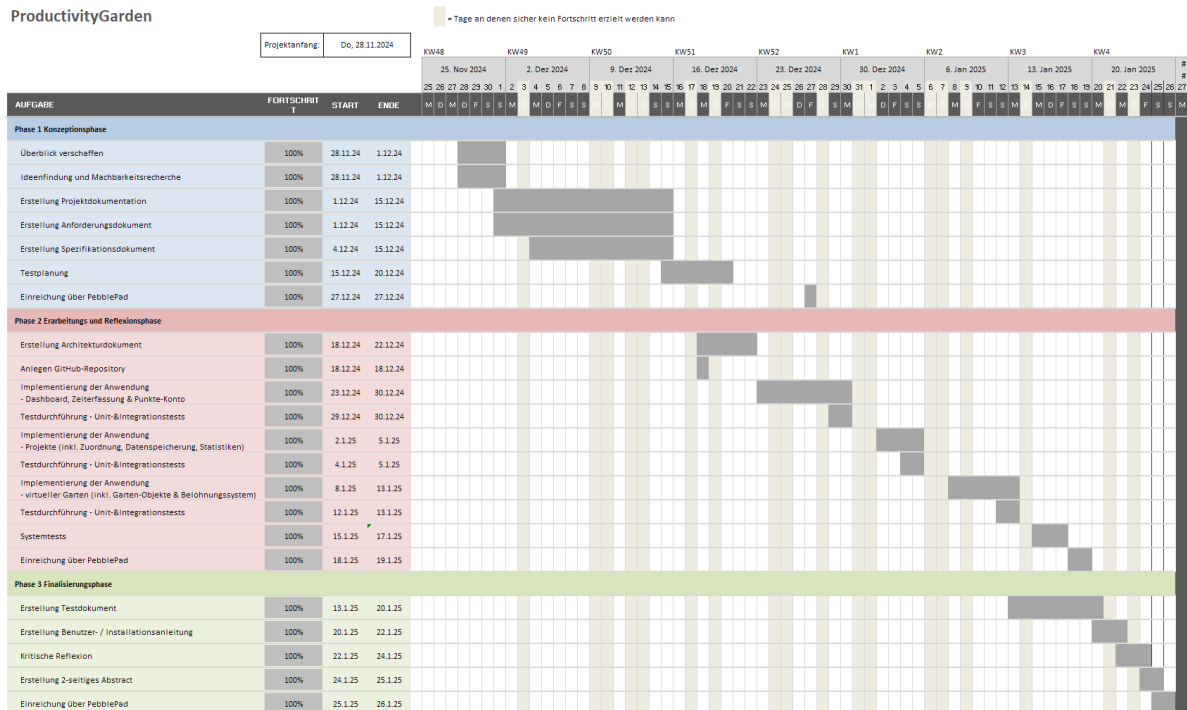


Abbildung 2 -Gantt-Diagramm aktualisiert

5. Literaturverzeichnis

Huseynli, B., & Uslu, A. (2024). A Qualitative Study on the Definition and Concept of Gamification. *Journal of Economic Sciences: Theory & Practice*, 81(1), (S. 40–50).
<https://doi.org/10.61640/jestp.2024.81.01.03>

Matallaoui, A., Hanner, N., & Zarnekow, R. (2016). Introduction to Gamification: Foundation and Underlying Theories. In S. Stieglitz, C. Lattemann, S. Robra-Bissantz, R. Zarnekow, & T. Brockmann, *Gamification: Using Game Elements in Serious Contexts* (S. 3–18). Springer International Publishing AG.
<http://ebookcentral.proquest.com/lib/badhonnef/detail.action?docID=4710247>