# brainstorming

Prepare to party - incremental game

Normal home as the location. A party is to be prepared for a festival like easter or christmas, family get together

Limited resources of time, money, stresslevel. The goal is to prepare the party and get a high score by finishing within time, preparing well, not spending too much money and having a low stress level

Final score screen shows different images(ai generated) and a score for each category which leads to a final screen

**Limited resources** are time, money and you have a stress level, which you want to keep low

## UI

- List of actions related to the below mentioned categories. Each item has a time needed, money needed, stress reduced or increased number and you can either start or queue each action. The actions also have a description, that will be shown in a different ui part and a number of points they will add to the final result, which is hidden from the player

- Ui has a textlog, where the description gets shown of what happened after an action was completed

### Score Categories

### Food & Drinks

- Catering

- Buy food

- Cook food

- Ask everyone to bring some food

**Decorations**

- Order on creative platform

- Order on huge shopping site

- Buy in town

- Craft yourself

**Activities**

- Crafting something

- Playing games

- Relax and talk

**Datenstruktur**

**Model-Klassen**

Action (mit Eigenschaften wie Zeitkosten, Geldkosten, Stressänderung, Punktzahl, Beschreibung usw.)

Category (entweder als Kategoriegruppe mit Unterkategorien oder direkt mit einer Liste von Action-Objekten)

**UI-Klasse**

Für die Darstellung (z. B. Listenansicht der Aktionen, Ressourcenzähler, Log usw.)

**UI-Controller**

Für alle visuellen Reaktionen, Event Listener, Aktualisierung der UI bei Interaktionen

**ResourceController**

Verwalten und Berechnen von Zeit, Geld und Stresslevel

Enthält Methoden zur Ressourcenerhöhung/-verringerung und zur Validierung (z. B. ob genug Zeit/Money da ist)

**GameController**

Hauptlogik: Spielstart, Ausführung von Aktionen, Abbruch, Ende berechnen, Steuerung der anderen Controller

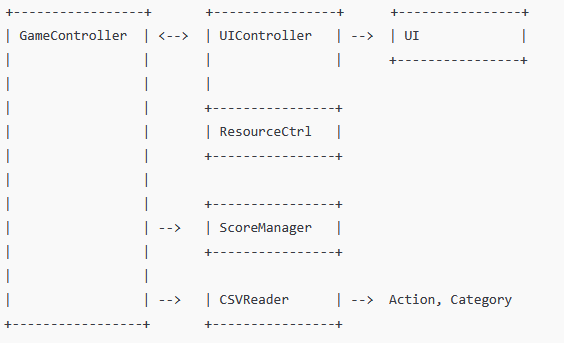
**CSVReader / DataLoader**

Einlesen von Aktionen und ggf. Kategorien aus einer externen CSV-Datei

**ScoreManager / ScoringController** (optional, aber sinnvoll)

Eine eigene Klasse zur Berechnung und Speicherung der Endwertung

Ermöglicht das leichte Austauschen oder Anpassen der Bewertungslogik (z. B. „Restzeit senkt Stresslevel → beeinflusst Score“)



**Goal**

Get as high a score as possible to

Expansions

- Different locations like garden, home, restaurant

- Achievements for finding special actions like self brewed alcohol or statues(adding easter eggs)

- Gifts for the guests can be added as another score category later on