cs2340 group 51

sprint 1 design deliverables

Domain Model

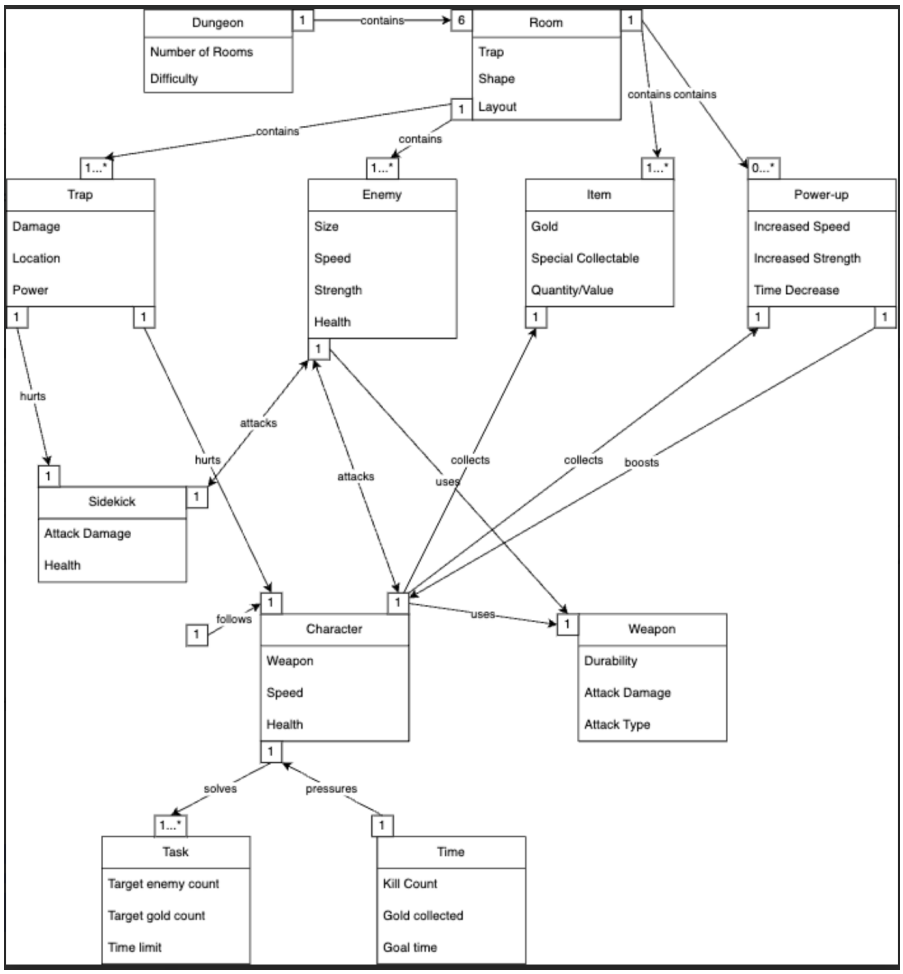
1. *Identify and list at least ten potential nouns* which could be used in your project. Some examples might be player, enemy, weapon, powerups, etc.
    1. Dungeon
    2. Player
    3. Layouts
    4. Enemy
    5. Attack
    6. Powerups
    7. Exits
    8. Rooms
    9. Health
    10. Score
    11. Leaderboard

2. *Identify classes and attributes:*
    a. *Classes (Game Objects) are nouns that require their own methods, attributes, and associations. Good examples of potential classes might include Player, Enemy, Weapon, etc.*
    b. *Attributes are descriptors of the identified classes (game objects). These are nouns that do not require a whole class to represent and describe a potential class. Good examples of potential attributes might include speed, direction, etc.*
    c. *List the identified classes and attributes on your submission PDF.*

- Dungeon (Class)
    - Number of Rooms (Attribute of Dungeon)
    - Difficulty (Attribute of Dungeon)
- Room
    - Trap
    - Shape
    - Layout
- Trap
    - Damage
    - Location
    - Power
- Enemy
    - Size
    - Speed
    - Strength
    - Health
- Item
    - Gold
    - Special Collectable
    - Quantity/Value

- Power-up
    - Increased Speed
    - Increased Strength
    - Time Decrease
- Sidekick
    - Attack Damage
    - Health
- Character
    - Weapon
    - Speed
    - Health
- Weapon
    - Durability
    - Attack Damage
    - Attack Type
- Task
    - Target enemy count
    - Target gold count
    - Time limit
- Time
    - Kill count
    - Gold collected
    - Goal time

3. Draw a Domain model for the classes and attributes you brainstormed.



4. *Connect each class within your Domain Model with at least one other class using associations.*
    a. *Example: Player "wields" Weapon*
    b. *Include multiplicities for each association, one on each side of the association. Please explicitly categorize the nouns as either classes or attributes somewhere in your deliverable in addition to including the domain model. The submission of the Domain Model itself does not suffice for the inclusion of the listed and categorized nouns.*

    For our game, we have many classes that associate with each other. In terms of map, there is a dungeon that contains different rooms, which are distinguished by the traps in the room, shape of the room, and the layout of the room. These rooms also contain multiple objects, such as traps, enemies, items, and powerups. Items and power ups are objects that the player can collect and use to enhance their gameplay and gain an advantage. Enemies are present in rooms and the objective for the player is to defeat all the enemies. In addition, there is a character and a sidekick. The sidekick follows the character around. Traps and enemies can hurt both the character and a sidekick. To defeat the enemies, the character can choose from a variety of weapons, which are differentiated by its durability, attack damage, and attack type. The score of this game is based on the time it takes for the character to finish the game. Lastly, while the character traverses through the rooms, there are tasks that the character must complete in order to move to the next room or finish the game.

Use Case Diagram

1. *Categorize the following actors as Primary, Supporting, or Off-Stage*
    a. **Player**: attempts to control the game character and reach the goal tile: **primary**
    b. **Game Admin**: uses the game's administrator screen to manage players: **primary**
    c. **Third-party database service**: services the game uses to store its state (i.e., for high scores): **supporting**
    d. **Shareholder**: provided financial backing to the software company that develops the game: **off-stage**
    e. **Contractor Company**: localizes the game to other languages: **supporting**
    f. **International Game Developers Association**: provides documentation on implementation of common game features: **supporting**

2. *Brainstorm one additional actor and categorize it like above*
    g. Additional Actor: **Customer Support**: offers support to players but does not directly interact within the system boundary: **off-stage**
    h. Additional Actor: **Leaderboard**: displays high scores and rankings of players, encouraging competition: **primary**

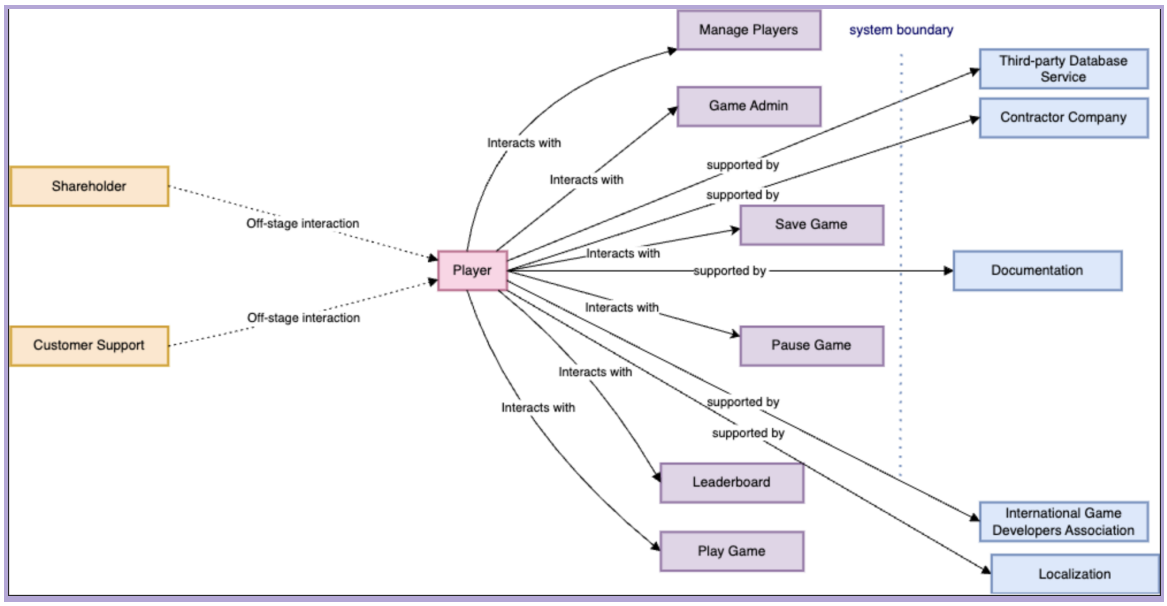
3. *Draw a Use Case Diagram for the game application*
    a. *Include the player and at least three other actors*
    b. *Place Primary Actors on the left of the system boundary*
    c. *Place Supporting Actors on the right of the system boundary*
    d. *Place Off-Stage Actors near but unconnected to the system boundary and place a "Off-stage" label under them*
    e. *Include six or more Use Case Titles (functional requirements within the system boundary). These functional requirements should define the ways in which Primary and Supporting Actors may interact with the system*

https://drive.google.com/file/d/1-1kDekU7MHeDBVZKAe1L85lEo8-342EB/view?usp=sharing

Use Cases Added

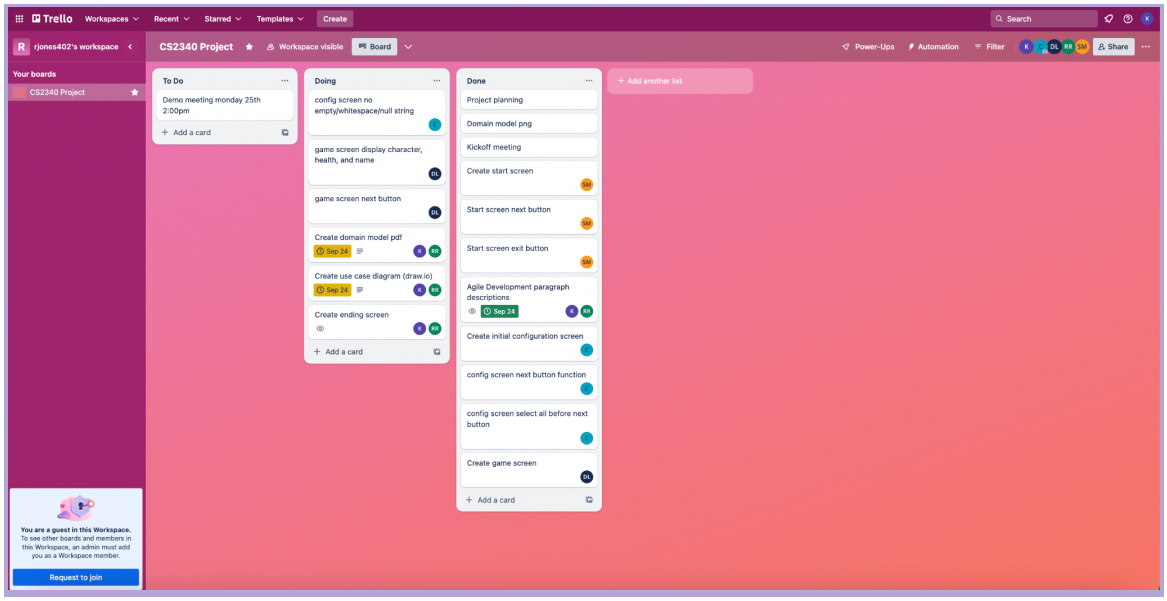
- *Play Game*
- *Manage Players*
- *Save Game*
- *Pause Game*
- *Localization*
- *Access Documentation*

Case Diagram:



Agile Development

1. *Submit a screenshot of the project management tool (Trello, Jira, Rally, etc.) used in Sprint 0.5 and Sprint 1. The screenshot should capture a clear view of the tool's board, showing columns/categories such as "Sprint Backlog", "In Progress", and "Done". Each task within these categories should have an assignee, indicating which team member is responsible for its completion. For best practices, it's advantageous if tasks also have associated "points" that represent the estimated effort required for completion. This serves as a visual indicator of the team's workflow and progression through the sprints. Some helpful examples are linked here: Ref1, Ref2*



2. *Write a paragraph description of how the team utilized the project management tool in Sprint 0.5 and Sprint 1.*

Sprint 0.5 had our team collaboratively work with the project management tool of our choice, Trello, for the first time collectively. Having used the tool before, some of us spearheaded the initiative to put our smallest tasks and to-dos in the right categories as we

all caught up with our collective working style. The team Trello played a pivotal role in laying the foundation for our development project. During the initial stages, we used Trello to establish a primary timeline, allocate resources, and define the scope of our project as broken down into smaller and easy-to-interpret tasks. To kickstart development, we created specific milestones we aimed for each team member to individually and collaboratively complete within a mutually agreed-upon timeframe. Team members were assigned their responsibilities based on preference and previous expertise, task dependencies were identified and visualized within the tool's interface. Additionally, we utilized Trello's collaboration features to facilitate communication and foster a shared understanding of the project's goals and objectives.

Approaching Sprint 1, the team Trello continued to be our central hub for planning and execution. We utilized it to prioritize and schedule tasks, ensuring that our development efforts aligned with the overall project vision. Progress tracking became a core function, since the tool allowed us to update task statuses, record team effort, and monitor the completion of features to streamline workflow and ensure smooth collaboration. The tool's reporting capabilities were utilized to generate sprint progress reports and burndown charts, offering valuable insights into our team's productivity and helping us make informed adjustments to our plans. Overall, the project management tool played a crucial role in guiding our project from inception to execution, enabling effective collaboration and transparency and ensuring that we remained on track to meet our goals and deadlines.

3. *Write a paragraph description of how the team conducted scrum meetings for Sprint 1.*

In the first week of team meetings, we agreed upon a common time for our weekly meetings. In our days moving forward, we selected our scrum master who assigned, updated, and shared weekly updates and progress reports to us based on how far along we were on our discussed timeline. With an emphasis on maintaining effective communication and transparency, we started with weekly stand-up meetings alongside in-class meetings, as suited to everyone's schedules. During these brief sessions, team members shared their progress, plan ahead, and any roadblocks faced thus far. This allowed us to identify impediments and offer assistance where needed, ensuring that the sprint's objectives remained on track.

Before the start of the first sprint, we conducted a Sprint Planning meeting, during which we determined the amount of work we could commit to for Sprint 1 and set clear goals.

Midway through Sprint 1, we held a Sprint Review meeting to showcase the work completed to our teammates and gather feedback. This feedback loop was essential for making any necessary adjustments and ensuring that the project aligned with the expectations we began with.

At the end of Sprint 1, we held a Sprint Retrospective meeting to reflect on our progress, working styles and teamwork. We discussed what went well, what could be improved, and identified any action items to implement in the next sprint to improve our productivity and collaboration.

The Sprint format and the routine Scrum meetings served as crucial touchpoints for the team, encouraging adaptability, transparency, and dynamic improvement throughout the development process.