

# Requirements and Analysis Document for Gastuen

Karl Gunnarsson, Jakob Ristner,  
Jakob Henriksson, Erik Berg, David Andreasson

2020-10-23

## 1 Introduction

The general purpose of the the application "Gastuen" is to act as a digital representation of a board game inspired by the physical board game "Betrayal at House on the Hill"[1]. The premise is that a group of adventurers are exploring a haunted house filled with traps, items and different events. After a while one of the players betrays the other players, thus taking the side of the monsters.

The game is played in turns and locally on one computer. The main stakeholders of this project are us students. The main focus is our learning process during the project.

The main benefit of this application is that it enables the user to play a boardlike game digitally. We believe that a digital version will improve the immersion during a game session.

### 1.1 Definitions, acronyms, and abbreviations

- rollDiceEvent - An event where the outcome affects the players stat. This event is based on a dice roll.
- moveEvent - An event where the outcome affects the players coordinates. This event is based on a dice roll.
- itemEvent - An event giving the player an item. This event is not based on a dice roll.
- gameWonEvent - An event that results in the player winning the game when entered. This event is not based on a dice roll.

- Tile map - a grid of squares representing the different rooms in the house.
- Sprite - Circles on the Tile Map representing the different players.

## 2 Requirements

### 2.1 User Stories

#### 2.1.1 Implemented User Stories

- As a player I want clear feedback in a way that makes me understand what is happening while the game is running.
  - Does the GUI tell the user what happened?
- As a player i want to be able to interact with the game so that it's playable.
  - Can I press buttons?
  - Can I move my character?
  - Can I interact with the GUI?
- As a player I want to be able to move on the board, thus enabling me to play the game.
  - Can I move through the rooms?
  - If I click on a door, do I enter the next room?
  - If I roll the dice based on my characters speed, can I move through as many rooms equal to the number I roll on the dice?
- As a player I want to be able to roll a dice so that I know how many steps I can take on the board.
  - Can I throw the dice?
  - Is the dice randomized?
  - Does the result enable you to walk accordingly?
- As a group of players we want to be ale to play together so that everyone can join in.
  - Can more than one player play through a session?
  - Can more than one player join the game?
  - Is the session turn based?
- As a player I want a new experience every time I play the game as this would prevent the game from feeling repetitive.
  - Is the game structure randomized?

- Are the tiles randomized every session, resulting in a unique map every time the game is played?
  - Are the events and items randomized every session?
- As a player I want to be able to win the game because it is fun to win.
  - Are the winning conditions clear?
  - Can I win?
- As a player I want the "Haunt mode" to start, thus changing the rules of the game and enables the game to end.
  - Can the game mode Haunt be activated?
  - Does the Haunt work properly?
  - When the "Haunt" triggers, does the GUI present the new rules to the "haunted" player?
  - When the "Haunt" triggers, does the GUI present the new rules to the "non haunted" players?
- As a player I want the ability to choose a character to play as and differentiate myself from the other players.
  - Is there a variety of characters?
  - Can I choose my character?
  - Does the characters have different stats?

### **2.1.2 Unfinished User Stories**

- As a player i want the ability to choose a character to play as and differentiate myself from the other players.
  - Does the characters look different?
- As a player I want a new experience every time I play the game as this would prevent the game from feeling repetitive.
  - Is the haunt effect randomized?
  - Is there a new path through the map every game?

## **2.2 Definition of Done**

For our project to be considered done, the following must be achieved:

- Code should be well documented
- Code should pass all tests
- Code is reviewed and tested
- Code is under version control

## 2.3 User interface

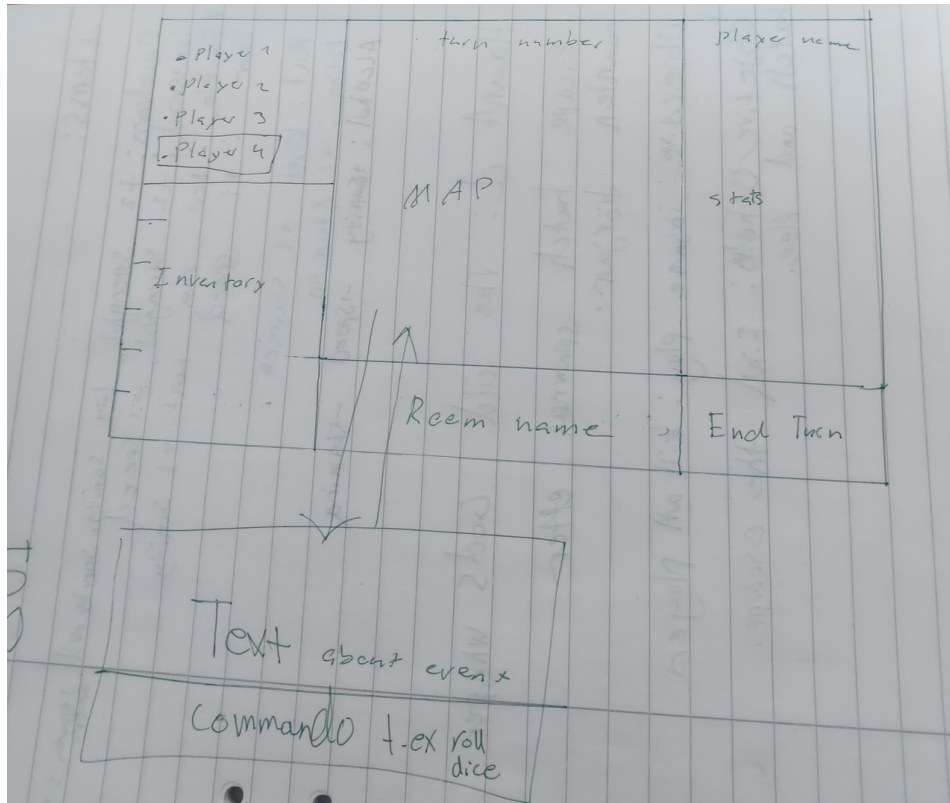


Figure 1: First draft of a User Interface

Figure 1 shows a rough sketch of the main view during a game session. The main view consists of a view that shows the current player, a view that shows the tile map, a view that shows the current room name, a view that shows the current players stats and a view for the current players inventory.

The sketch also includes a lightbox that is supposed to display text about an event if one exists in a room. The lightbox also contains a commando prompt where the player can roll the dice to perform the event.

The current ideas for navigating through the rooms on the tile map is that either the player presses a door in current tile to enter the next room, or that the player uses the arrow keys to enter the next room. The main game view also includes a button for ending the turn.

Furthermore is the application planned to have two more views. One view for selecting how many players the game will have, and one view where the players can select a character.

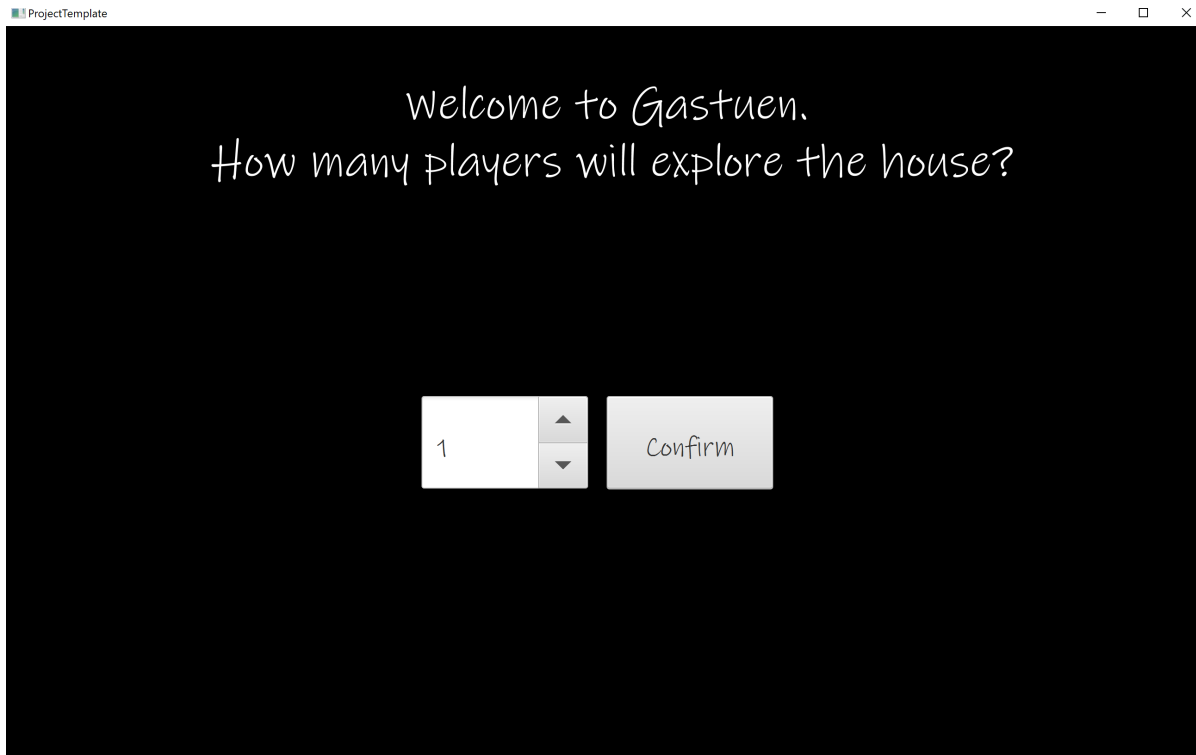


Figure 2: Final draft of the start view

Figure 2 shows the current version of the start screen when you run the game. The start screen consists of a header asking how many players will play the game, a spinner for choosing the amount of players that are going to play and a confirm button for navigating to the next view.

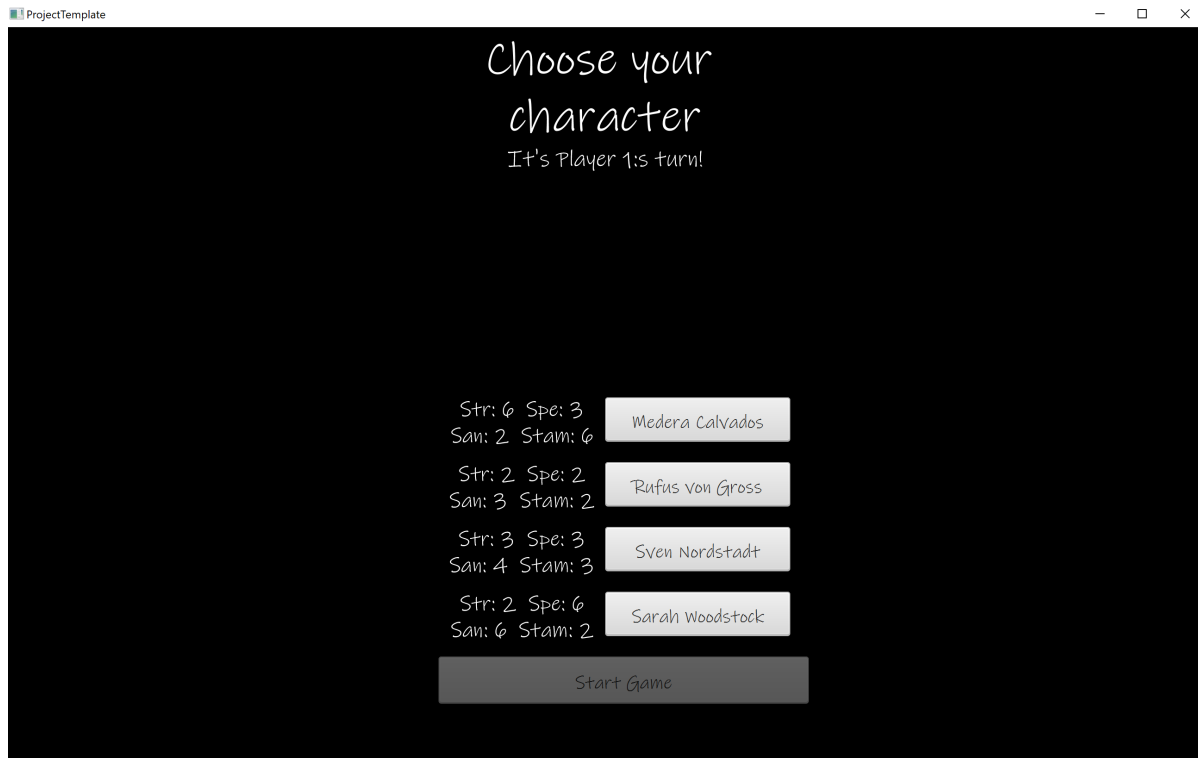


Figure 3: Final draft of the choose character view

Figure 3 shows the next view after pressing confirm in the start screen(see Figure 2). This view functions as a character select screen and figure 3 shows us the view before any characters have been picked. This view consists of a header prompting the players to choose a character. The text under the header shows which player's turn it is to pick a character. The character is chosen by pressing the the button with their names on and the characters stats are displayed to the left of the button.

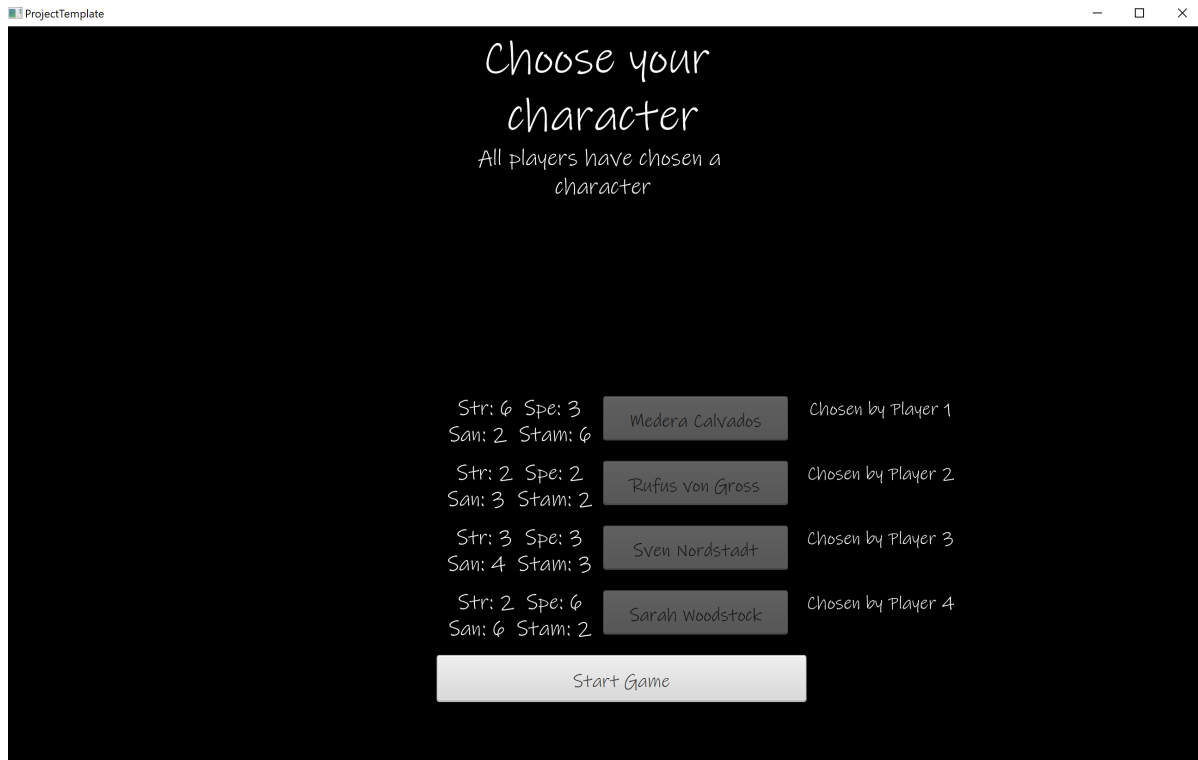


Figure 4: Final draft of the choose character view, after characters have been picked by the players

Figure 4 shows the choose character view after characters are picked. Note how the buttons for the characters are greyed out, implying that they are not clickable anymore. The text under the header displaying each characters turn to pick a character now also says "All players have chosen a character". After a character is picked it is also displayed in text to the right of the button, which tells which player choose that specific character. Also note how the button that says "start game" is now lit up, implying that it is now clickable. Pressing the start game button starts the game and takes you to the next view.



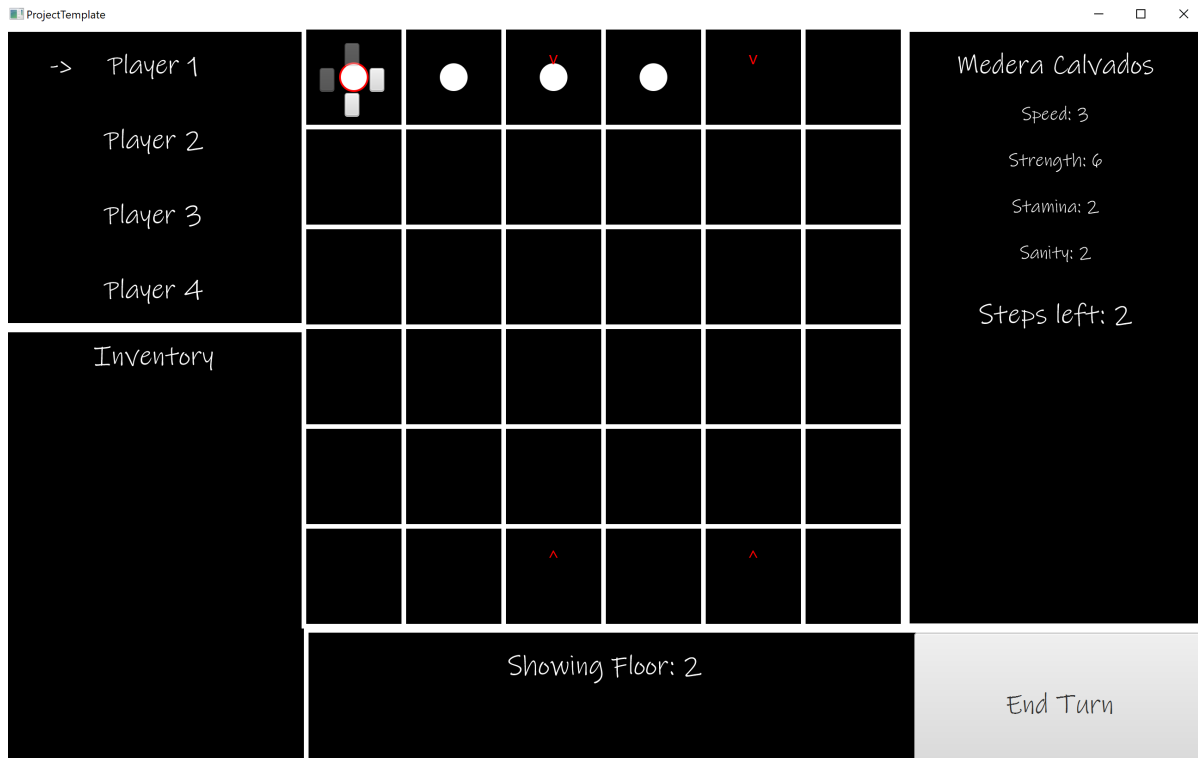


Figure 5: Final draft of the main game view

Figure 5 shows the main game view. This view consists of the following: A list of players and a current player indicator(the arrow to the upper left). An inventory. A grid representing the different rooms in the house. Player sprites consisting of white circles. A view for displaying the current players stats and steps. A end turn button and a view for showing the current floor. The current players sprite is marked with a red circle. The four buttons around the current active sprite are used to navigate through the different rooms on the map. The tiles on the map(representing the rooms in the house) can either be empty or contain an event. After the current player has run out of steps you end the turn by pressing the "End Turn" button, thus starting the next turn for the next player.

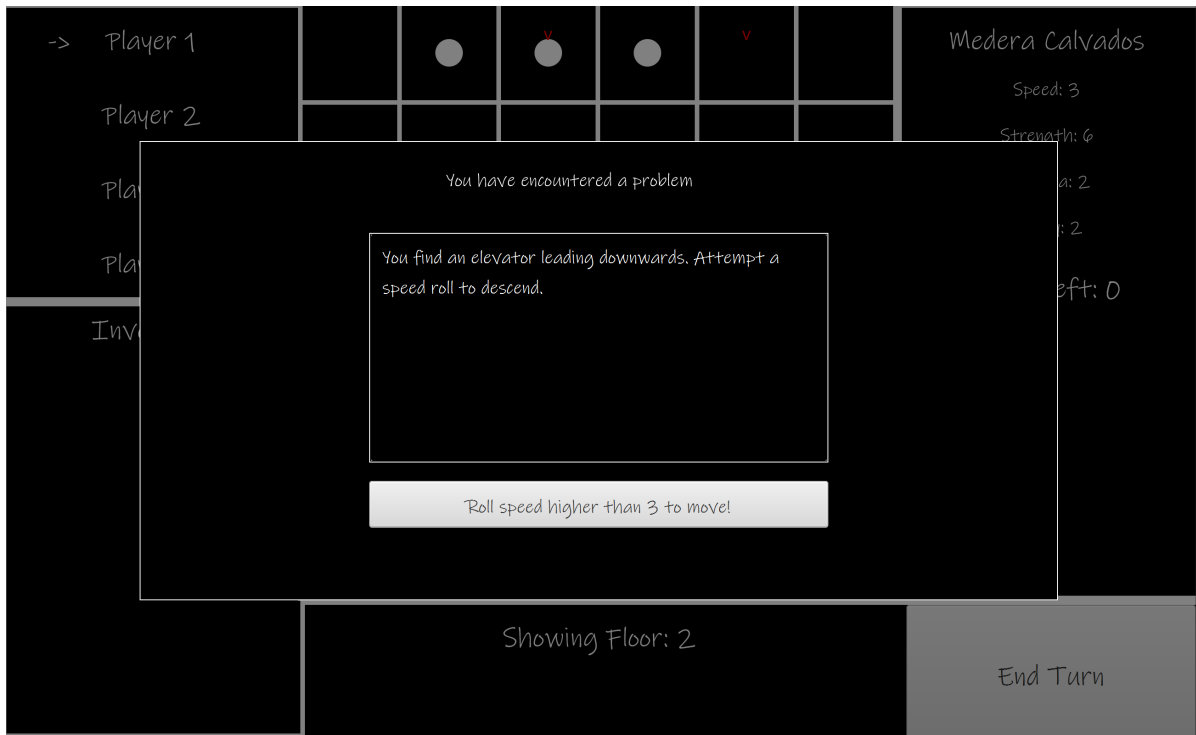


Figure 6: Final draft of an event view

Figure 6 shows an example of an event view and in this case the view for a moveEvent. The event view consists of a light box that greys out the background when present. An event is triggered when a player steps on a tile containing an event. The modal panel in combination with the light box is used to put emphasis on the event at hand, thus drawing the attention of the player to the event. The modal panel consists of a header, a text area describing what the event is about and a button that tells the player what they need to roll to pass the event. The roll is done by pressing the button and also is used to navigate back to the main view. After going back to the main view the result is displayed under the current floor text in red text. The red text then fades after a few seconds.

This particular event is a moveEvent(finding an elevator) but the view for a rollDiceEvent looks the same way. Therefore there is no figure of the view for the rollDiceEvent included in this section.

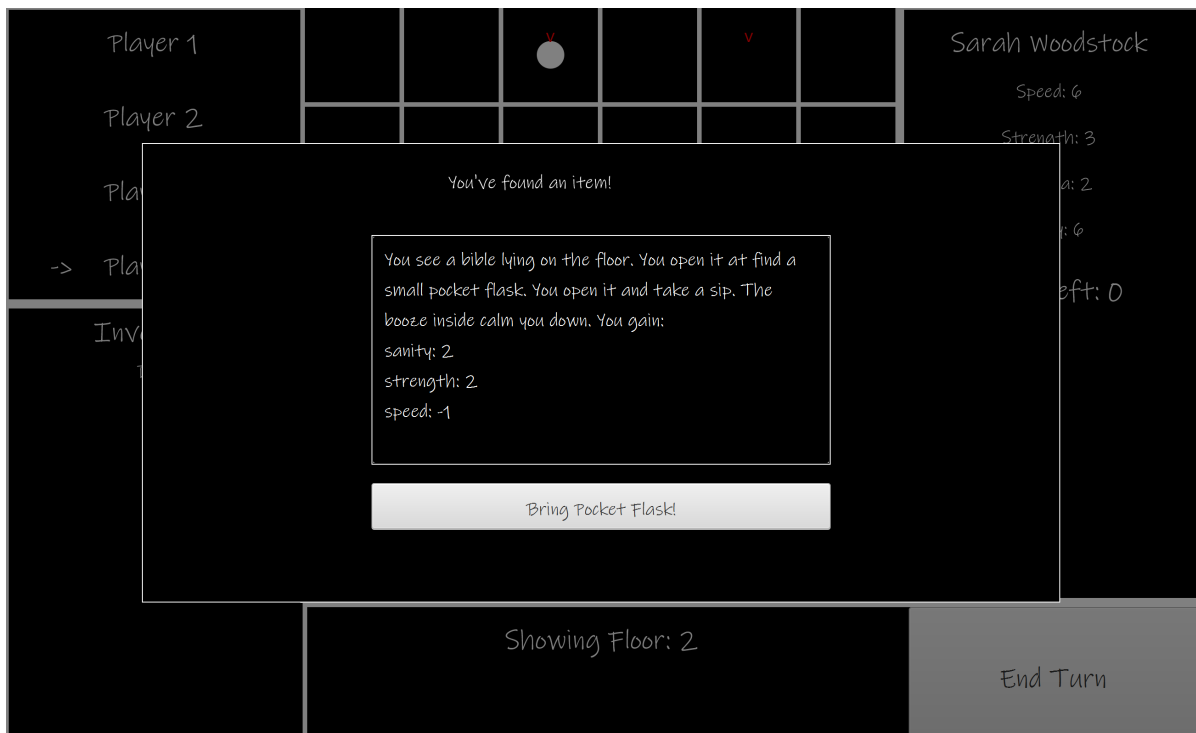


Figure 7: Final draft of the item event view

Figure 7 shows the view for an itemEvent. This view is almost identical to the other eventViews, with the exception that you do not have to pass a roll to obtain an item. The items stats are also displayed in the text area. The button is used for accepting the item and navigating back to the main view.

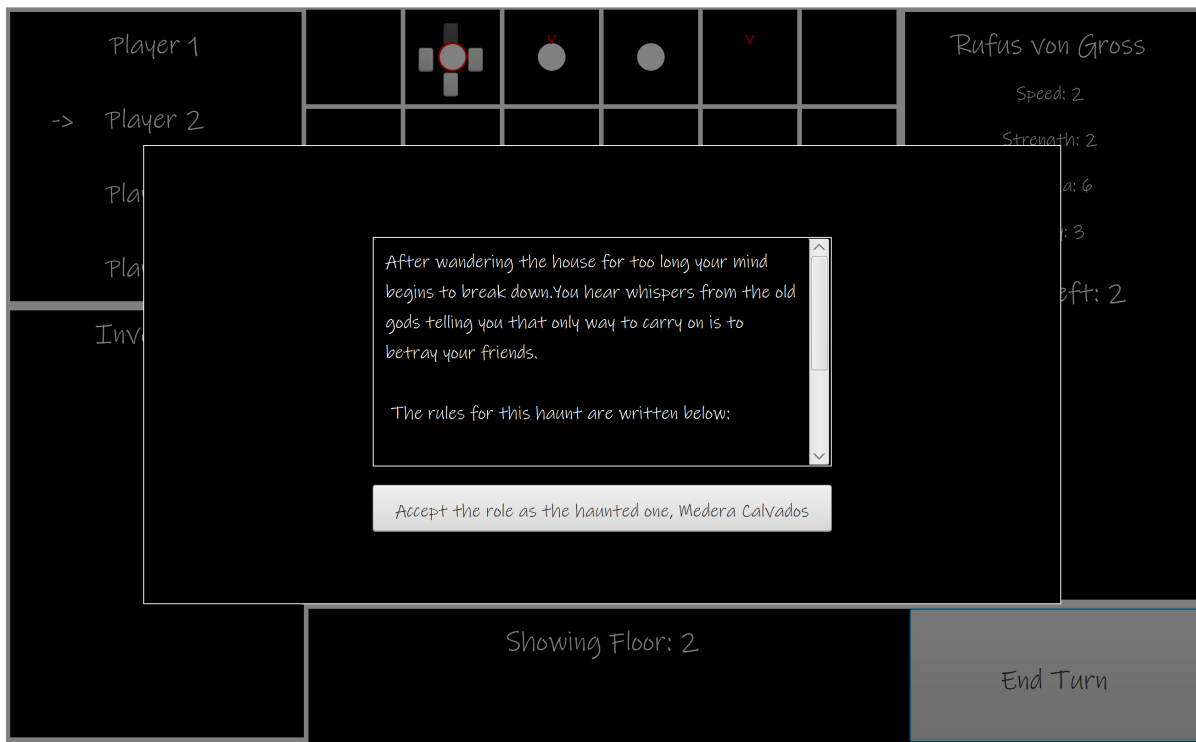


Figure 8: Final draft of the haunt event view

Figure 8 shows the view for haunt event. The haunt event is activated after a set number of events have been done (currently 8 events). The player that activates the last event is the one that ends up haunted. The haunt event view is constructed similar to the other event views. The difference is that the text area contains information about the haunt and the different rules for the haunted player and the other non haunted players. The name of the haunted person is also displayed on the button which as per usual is used to navigate back to main game view.

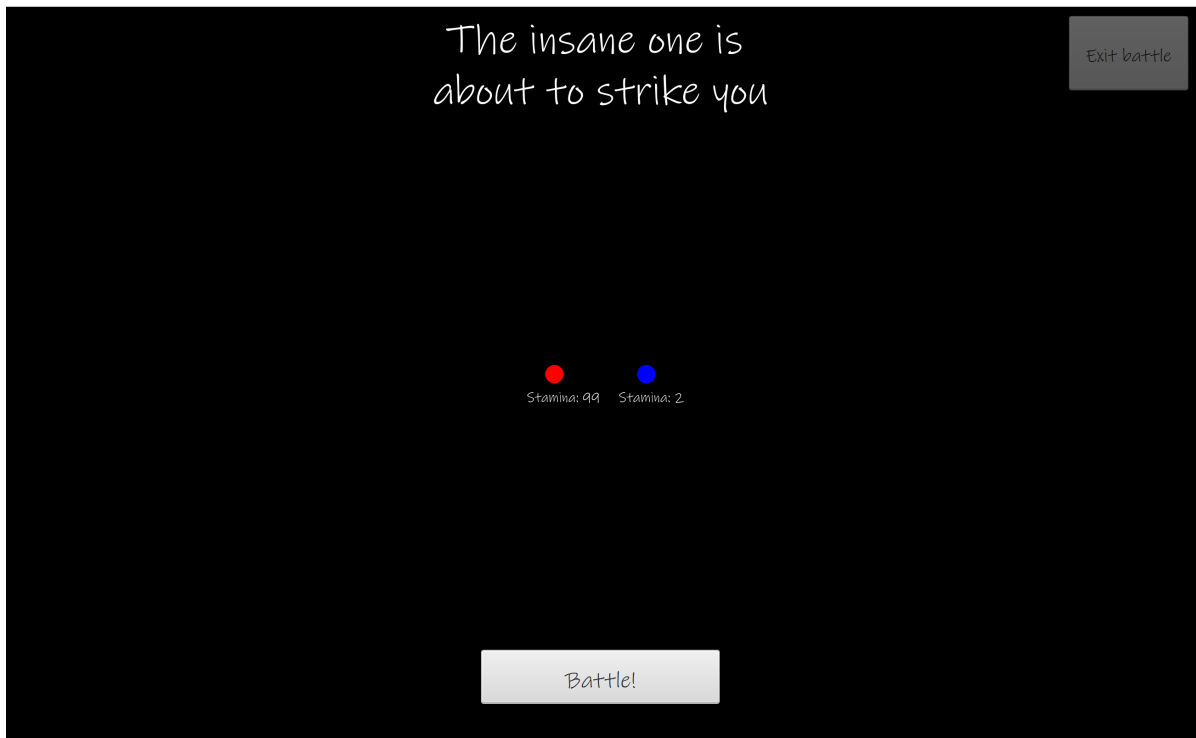


Figure 9: Final draft of the combat view

Figure 9 shows the view for combat during the insanity haunt before a battle. The view consists of the following: A header. The sprites that are in the same room (red representing the haunted player, blue representing a non haunted player). The players stamina and a button to execute a battle. The combat supports battle for more than two players in a room. If there are more non haunted players in the room they will show up as another blue circle sprite. The one attacking is always the haunted player. If there are multiple non haunted players in the room they will all get hit by the same attack. There is also a button for exiting battle which currently is grayed out as there hasn't been a battle yet.

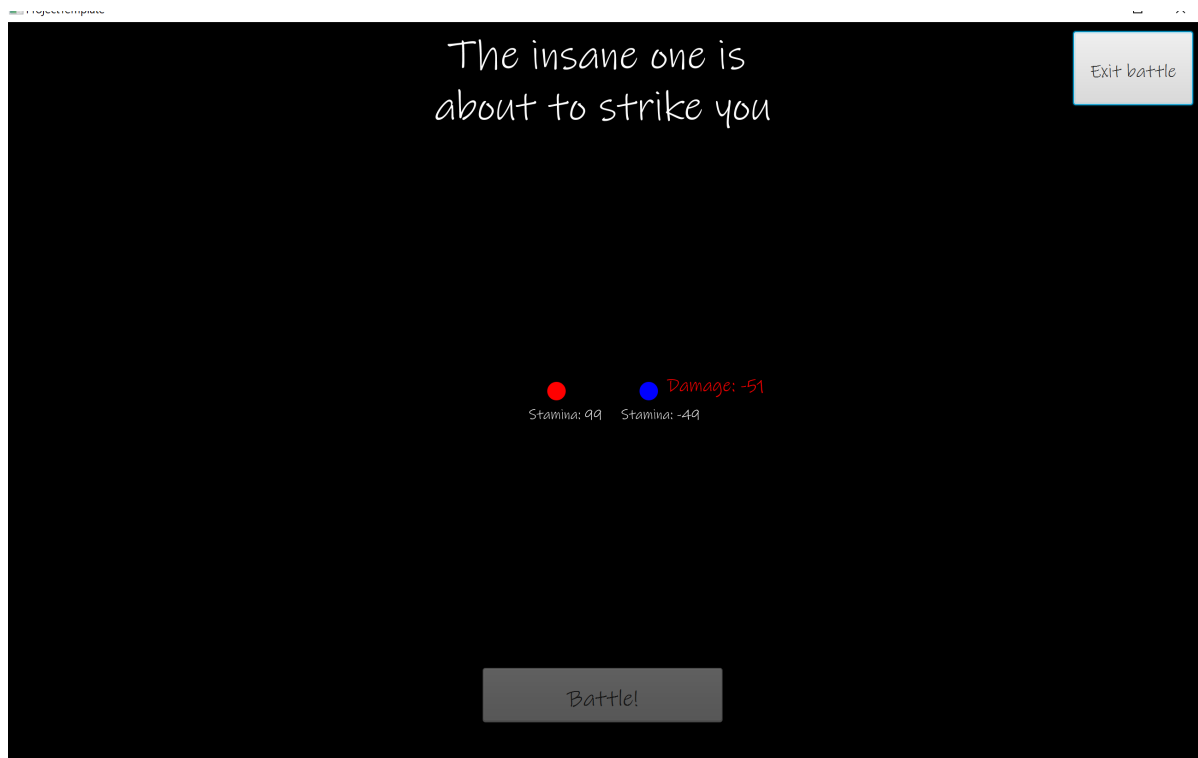


Figure 10: Final draft of the combat view, after a battle

Figure 10 shows the combat view after a battle has been executed. The damage to the non haunted players are displayed in red text. The battle button is now grayed out and the Exit battle button is lit up to indicate that you can now leave battle and navigate back to the main view.

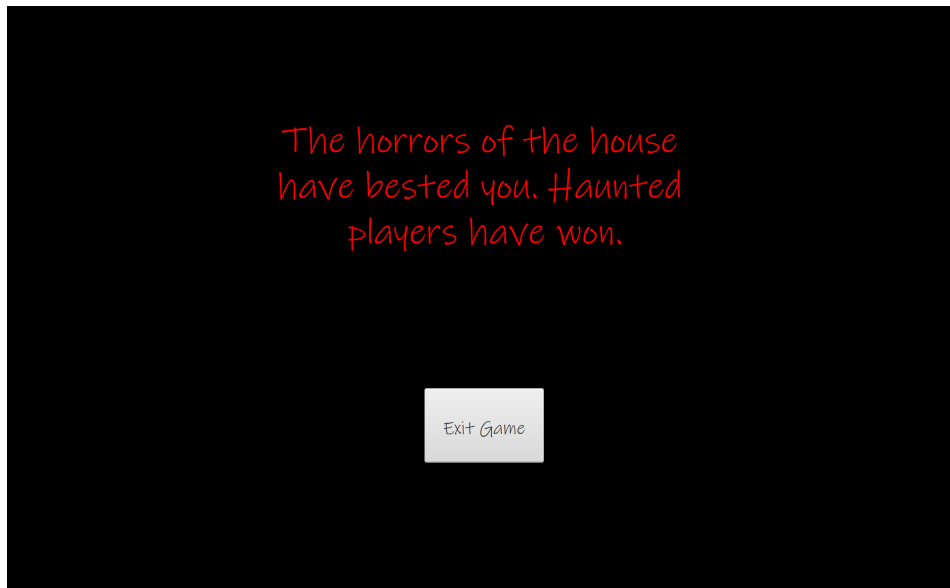


Figure 11: Final draft of the haunted side winning view

Figure 11 shows the view for when the haunted player wins the game. It consists of a header telling all the players that the haunted player/players have won and a button for exiting the game.

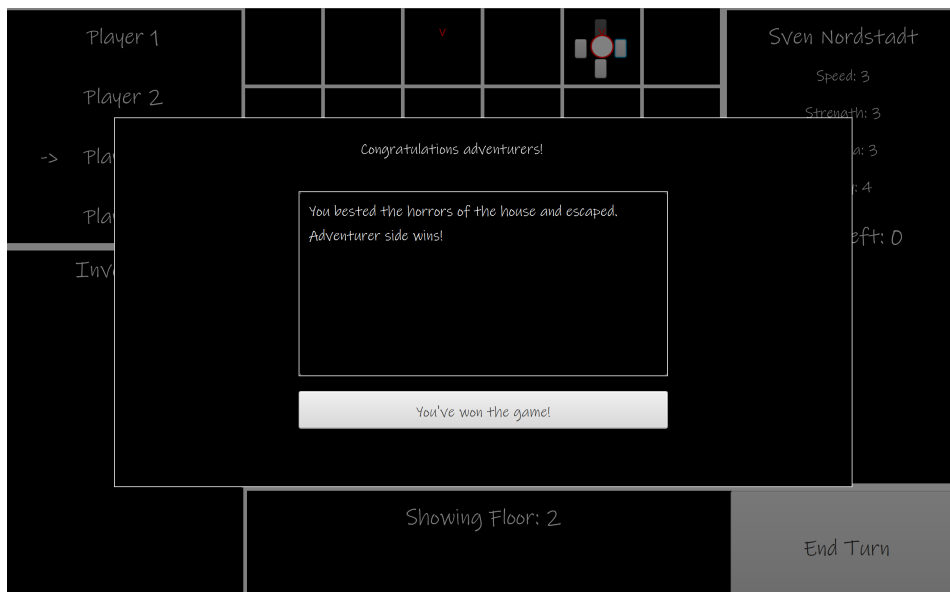


Figure 12: Final draft of the adventurer side winning view

Figure 12 shows the view for when the non haunted players win the game. It has a similar appearance to the other event views. Pressing the button will exit the game.

## 3 Domain model

The domain model shows how the application is modeled.

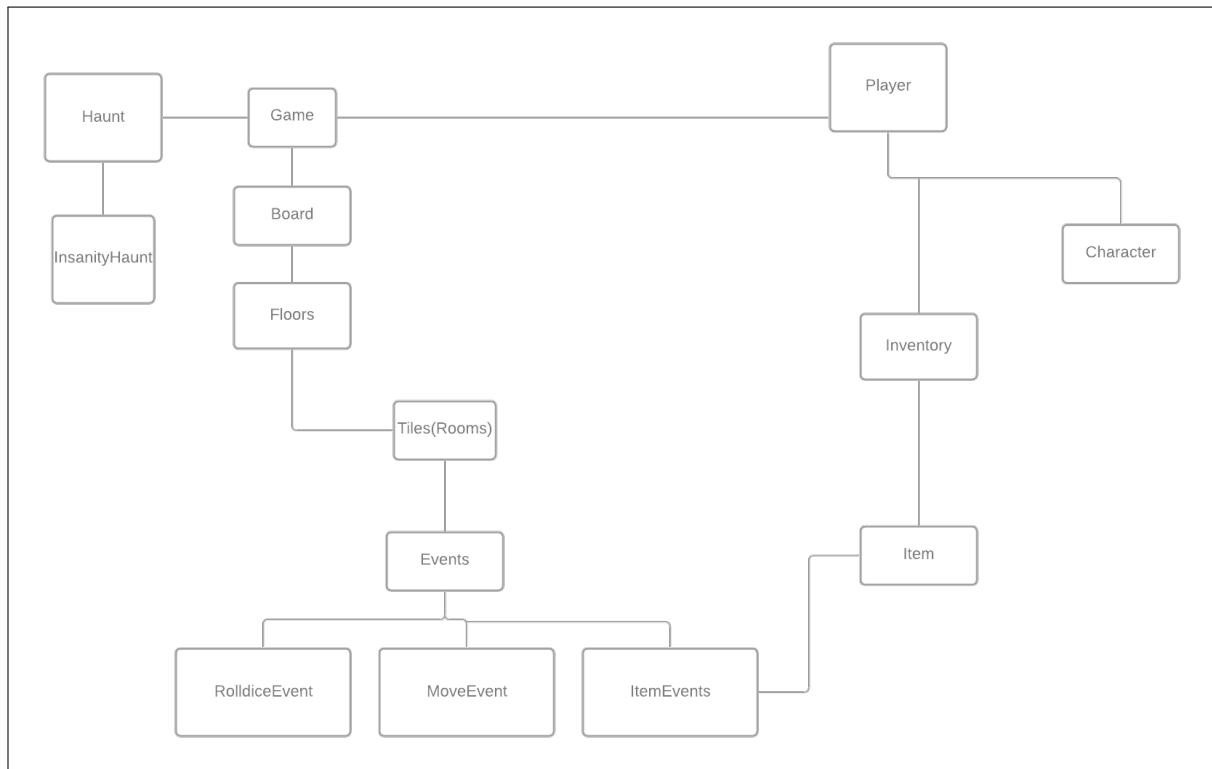


Figure 13: Domain model of Gastuen application

### 3.1 Class responsibilities

Explanation of class responsibilities in the domain model.

- The **Game** class is a representation of the entire game that handles most of the game logic and sends updates and data between classes.
- The **Haunt** is a representation of different game states that the game can enter.
- The **Board** class is a representation of the game board that hold the floors and tiles. It also gives events to the different floors.
- The **Floor** class is a representation of a group of tiles. It represents the different floors of the haunted house.
- The **Tile** class is a representation of a room. The tiles can also hold events which triggers when player walk on the tile.



- The **Player** class is a representation of a player. It handles how the player moves and where the player is on the game board.
- The **Character** class is a representation of the characters that a player can choose. It handles the character stats and changes to the stats.
- The **Inventory** class is a representation of a bag that can carry items. It shows what items a character holds.
- The **Item** class is a representation of an item.
- **Events** is a high level representation of all the events that can happen in a room. It is the interface towards the rest of the game when the game interacts with the events.
- The **ItemEvent** class is an event that gives you an item.
- **RollDiceEvent** is an event involving a task that can succeed or fail based on a dice roll.
- **MoveEvent** is an event that moves the player based on if the fail or succeed a dice roll.

## 4 References

### References

- [1] (Oct. 2020). Betrayal at house on the hill, Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Betrayal\\_at\\_House\\_on\\_the\\_Hill](https://en.wikipedia.org/wiki/Betrayal_at_House_on_the_Hill) (visited on 10/21/2020).