Alpha release

Wallther, don't let the wall falter!

Towsif Zahin Khan, Tobias Heller, Leon Stier, Karl Zakhary

June 17, 2024

1 Development Progress

At this stage, it is safe to say that we have implemented the core elements and mechanics to realize our initial model of the game. We have not faced too many technical or game design issues. We wanted to try several ideas including using diegetic UI where it made sense, but we always had a "Plan B" just in case our initial ideas did not bare fruit. These ideas and fall back plans were discussed during intense team meetings. These meetings have also lead to us creating the software architecture of the entire game, which includes not only the proper separation of concerns and functionalities, but also how these separate concerns communicate each other. For every detail of the architecture we discussed the most intuitive, clean and performant ideas, which we believe led to less coupled and more extensible components down the line.

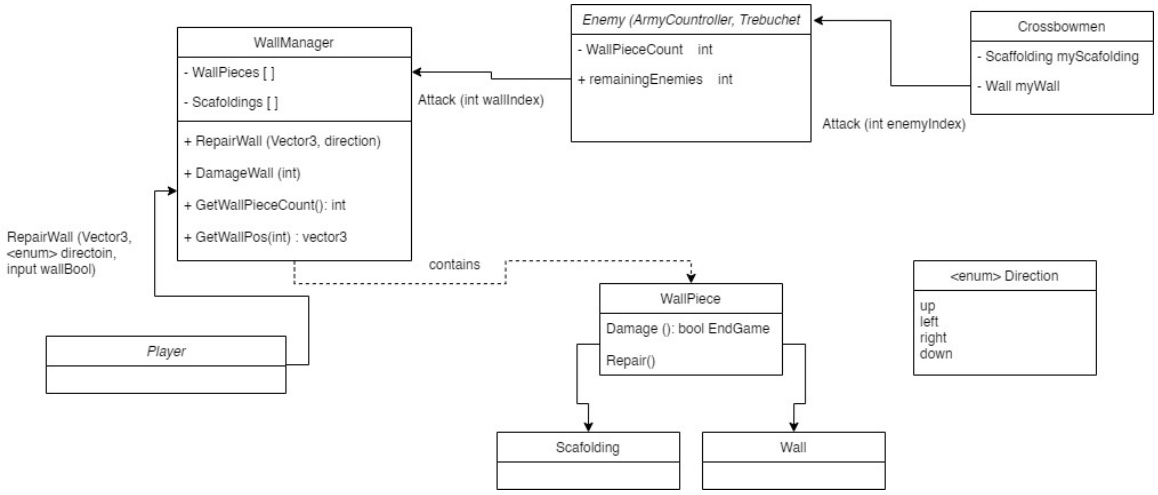

Figure 1: Architecture for the Games

1.1 Repair System

As the central mechanic of our game it was important to design and implement a working system to repair the wall where it's damaged. Since the last report we have done just that with the player now being able to walk up to a wall segment and repair either the wooden scaffolding or the stone wall. To avoid clunky material selection schemes we opted for two dedicated buttons, the bumper buttons on the game pad, to directly repair the chosen piece.

1.1.1 Selection and Highlighting

The player can choose to repair the closest segment, the adjacent segment to the left or right or even the one above. The segment to be repaired can be selected with the right analog stick. Whenever the player uses the bumper buttons the selected segment is repaired (if possible).
To give the player a visual indication what segment is selected at the moment we implemented a preview of what the wall piece would like like when it's repaired. This is done by showing a half-transparent version of the repaired mesh over the selected segment.

1.2 Resource Replenishment

Having enough resources within the player's inventory is a precondition for fixing a damaged wall segment or a burnt scaffolding. Wallther can now interact with two resource stores, one for wood and another for stones. The interaction results in refilling completely Wallther's inventory so that he can carry out the repairing actions. The amount of resources that Wallther has available is displayed diegetically by adding them to the actual player model.

1.3 Crossbowmen State Machine

Crossbowmen represent our non playable army that we have to protect by fixing the wall. We implemented the system responsible for synchronising the current state of the wall that influences the states and behaviours of individual crossbowmen. We have four states available for the crossbowmen:

- **Walk:** When the correct triggers are found (scaffolding.health $\geq$ 1 AND wall.health == fullHealth s) each crossbowmen is responsible for getting themselves to their preassigned wall segment.
- **Dead:** This state is reached from any other state when the following triggers are reached (scafolding.health == 0 AND wall.health != fullHealth).
- **Revive:** The crossbowmen are queued when they are ready to be revived. So they do not reach the walk state all at the same time, but one after another, for a nice sequence of marching soldiers.
- **Attack:** The attack state is reached simply when the crossbowmen reach their preassigned WallSegment so they can start shooting down the enemy forces.

1.4 Visual Effects

We have implemented so far some visual effects via leveraging the Unity particle system. The following list is the current state of achievements:

- **Fire Particles:** A visual element that simulates fires resulting from burning scaffolding.
- **Repair Result Highlights:** A translucent preview of the final state of the wall segment after repairing, so that the player has an idea about the final result when the segment is fixed and also doubles as a visual selection mechanic of which wall is going to be repaired if the repair button is pressed.
- **Smooth Texture Blending:** A visual effect that blends between the different meshes based on the change in health points.
- **Repair Particle Effects:** A visual effect that simulates smoke and debris flying out when the wall is being repaired.

2 Future Targets

In this section the main targets we will focus on for the next milestone are described.

2.1 Implement feedback

After presenting our game for alpha testing we want to collect as much feedback as possible to be able to implement it and focus on entirely new feedback in the playtesting milestone. We are anticipating insights on the balancing, especially Wallthers speed and capacities in the initial game state.

2.2 Sound Design

An important cornerstone of game development is of course sound design which we put in the back log for now. For the next milestone we will start implementing sounds to make the gameplay more satisfying and give better feedback to the player.

2.3 Level Progression

The precondition here is still not fully realized. Once the first level's vision is fully implemented and the feedback from the alpha release is collected and integrated, then we will start on working on the level progression.

2.4 Upgrade System

Playing the game should give us a sense about the appropriate balance we would like to achieve, hence, the difficulty of the game resulting from the level progression is in balance with the upgrade system.

2.5 Polishing

There is still some improvement when it comes to integrating the visual effects properly. Some adjustments to the player locomotion and the interactivity could be considered to increase the smoothness of the gameplay.

3 Development Timeline

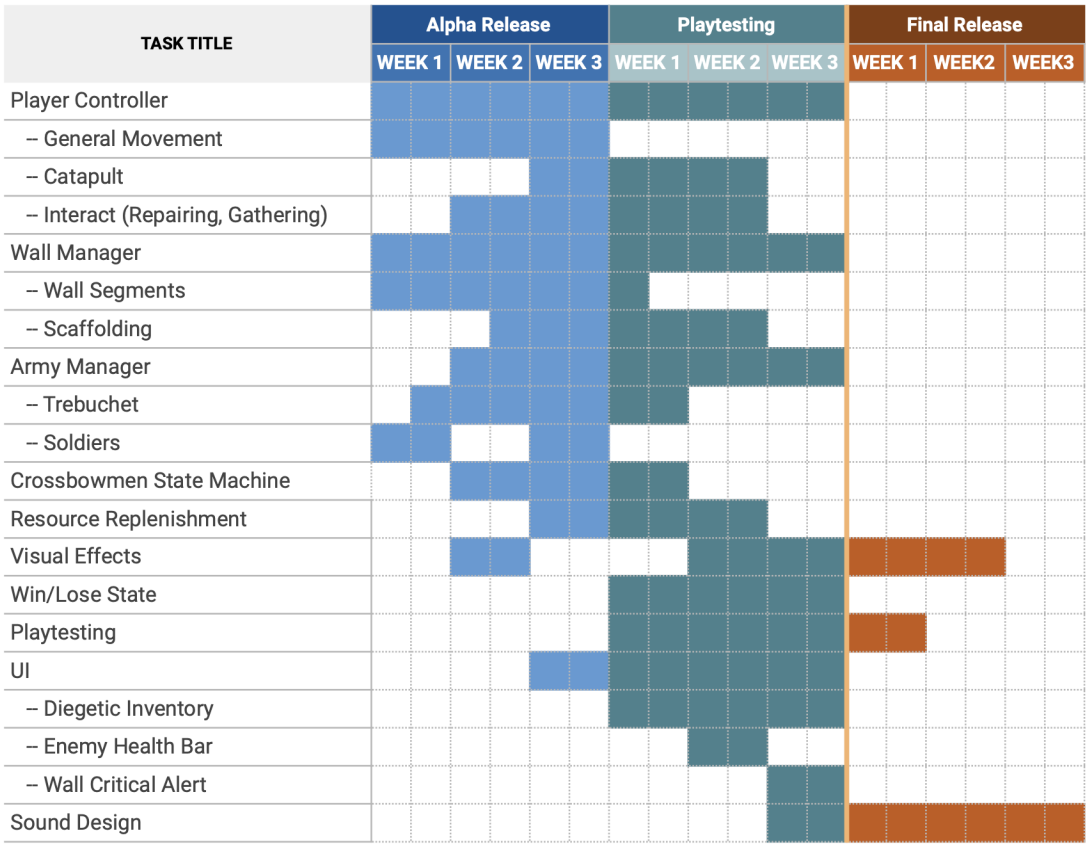

Figure 2: Task timeline breakdown and achievement