

# Almost Intelligent Machines



<b>1. Formal Game Proposal</b>	<b>4</b>
<b>1.1. Game Description</b>	<b>4</b>
1.1.1 Camera and game view	5
1.1.2 Win and lose conditions	5
1.1.3 Drops and pickups	5
1.1.4 Auto fighters (AIMs)	6
1.1.5 Boss	6
1.1.6 Arena	7
<b>1.2. Technical Achievement</b>	<b>7</b>
<b>1.3. "Big Idea" Bullseye</b>	<b>8</b>
<b>1.4. Development Schedule</b>	<b>8</b>
1.4.1 Layered Development Schedule	8
1.4.2 Milestones and tasks	9
<b>1.5. Assessment</b>	<b>11</b>

<b>2. Physical Prototype</b>	<b>12</b>
2.1 Goal of the Prototype	12
2.2 Prototype description	12
2.2.1 General rules	12
2.2.2 AIMS	13
2.2.3 The Boss	14
2.2.4 Items	16
2.2.5 The Arena	16
2.3 What we've learnt	17
<b>3. Interim results</b>	<b>19</b>
3.1 A.I.M Units	19
3.1.1 Player controls	19
3.1.2 AI	19
3.1.2.1 Confused behavior	19
3.1.2.2 Coward behavior	20
3.1.2.3 Aggressive behavior	20
3.1.2.4 Evil behavior	20
3.1.2.5 Stay and stare behavior	20
3.1.3 Behavior adoption	20
3.2. [L.A.M.P.] Lethally Armed Management Program	21
3.2.1 Currently implemented attacks	21
3.3. Models and Environment	23
3.4. Progress in the Development Schedule	24
3.5 Changes to Development Schedule	25

<b>4. Alpha Release</b>	<b>25</b>
4.1. A.I.M. Units	25
4.1.1 Player controls	25
4.1.2 AI	25
4.1.2.1 Confused behaviour	25
4.1.2.2 Coward behaviour	26
4.1.2.3 Aggressive behaviour	26
4.1.2.4 Evil behaviour	26
4.1.2.5 Stay and stare behaviour	26
4.2. [L.A.M.P.] Lethally Armed Management Program	26
4.2.1 Boss phases	26
4.2.2 Boss Attacks	27
4.2.3. Attack determination	29
4.3 Items	30
4.3.1. Weapons	30
4.3.2. Health Pack	30
4.3.3. Bomb	31
4.3.4. Digging Spots	31
4.4 Tutorial	31
4.4.1. Scene I	31
4.4.2. Scene II	32
4.4.3. Scene III	32
4.5 Sounds	32
4.6 3D Art & Effects	33
4.7 Progress in Development Schedule	33
4.8 Changes to Development Schedule	33

<b>5. Playtesting</b>	<b>34</b>
5.1 Tutorial	34
5.1.1 Scene IV	34
5.1.2 Scene V	34
5.1.3 Scene VI	34
5.1.4 Scene VII	35
5.2 Conducting the playtesting	35
5.2.1 Feedback form	36
5.3 Playtest results	37
5.3.1 Demographics	38
5.3.2 Gameplay	39
5.3.3 A.I.M. Bots	43
5.3.4 The Boss	45
5.3.5 Items	46
5.3.6 Overall experience	48
5.4 Issues and planned changes	49
5.4.1 Balancing issues	49
5.4.2 Tutorial Changes	50
5.4.3 Bugs and incomplete features	51
5.4.4 Controls	51
<b>6. Conclusion</b>	<b>52</b>
6.1 The Final Product	52
6.2 Experience	53
6.3 Course personal impression	54

**Keywords:** boss fight; dynamic gameplay; hot-seat multiplayer; AI-controlled NPCs; friendly fire;

# 1. Formal Game Proposal

## 1.1. Game Description

A.I.M. is an engaging, fun to play solo or with a group of friends boss-fight experience, that combines dynamic gameplay with strategic thinking.

The players are put in a pit against a powerful enemy, that is bent on the destruction of everything that moves in its reach. If something doesn't move, the Boss will come closer, push it and *then* destroy it. Players operate a squad of highly intelligent automated fighters and have the ability to take the direct control of one of the auto-fighters at any time they want. The only problem that makes this mission much harder than it was supposed to be is that the Boss is radiating a field that interferes with the functioning of the quantum brain of the fighters, making them *not that intelligent* on their own during the encounter. Help your squad to focus on the goal, avoid the damage and exploit the Boss's weaknesses. Non-forgiving friendly fire makes the players think twice when firing or distributing powerful weapons among the auto-fighters, as, after all, all the most potent and fun weapons have the "use only under competent supervision" tag on them.

The Boss is a smart and dangerous creature that knows its and the players' weaknesses. It has the ability to take control of the players' units and *will use it* to gain advantage and cause mayhem in the squad.

The squad consists of 5 to 10 auto fighters and it is up to players to be able to efficiently utilize their dumbed down AI in order to defeat the Boss. The players have freedom of choice when it comes to which auto-fighter they want to control at a time. The controlled fighter can shout out commands to the nearby allies, affecting their behaviour accordingly. And even being dumbed down, the auto-fighters are still smart enough to learn by example and will try to assist the controlled unit to the best of their abilities. Monkey sees, monkey does.

The action takes place in the ruins of an industrial building, where the Boss was built and went out of control during the testing and experimentations. Use the remains of the walls and foundations to your advantage to take cover from the enemy (*or friendly*) fire or look in the ruins for the experimental weapons that survived the collapse of the factory.

The hot-seat multiplayer feature makes it a perfect game to spend the time with your friends and will throw you back to the precious memories of playing the SEGA games during the hot summer days in your friend's house.

### 1.1.1 Camera and game view

The players have an isometric view of the game action. This is a well-proven approach for these types of games (e.g. Magicka, Alien Swarm, Helldivers, etc.).

### 1.1.2 Win and lose conditions

The game objective is to destroy the Boss. Players lose when all of the fighter units get destroyed. Each fighter unit has its own hit points pool. The Boss has multiple stages and is destroyed only when the last stage is completed. As the fight progresses, the Boss becomes more dangerous and powerful, which sets a soft cap on the maximum amount of time spent in one playing session.

### 1.1.3 Drops and pickups

During the fight, the pickups will be spawned in the area. Those pickups help the players and include (the list is not final):

- Weapons
- Buffs (temporary enhancements)
- Repair kits (health packs)

The spawn of the pickups is tied to the unfolding action and happens when the ground gets damaged with powerful weapons (mostly from boss attacks). The players will have a limited time to pick up the spawned item in order to prevent them from hoarding pickups until the later stages of the fight. This also adds an extra layer of complexity to the decision-making during the play.

Examples of weapons:

1. Assault rifle – high rate of fire, low damage
2. Sniper rifle – low rate of fire, high damage
3. Laser gun – beam-type weapon, high damage/sec
4. Rocket launcher – low rate of fire, slow missiles, AoE damage

Examples of buffs:

1. AI boost – for a short time, the auto fighter acts as an intelligent AI;
2. Damage UP – deal more damage for a short amount of time;
3. Armour UP – receive less damage for a short amount of time;
4. Camouflage – the Boss ignores the unit for a short amount of time.

### 1.1.4 Auto fighters (AIMs)

The auto fighters fight on their own trying to help the players to defeat the Boss, although their actions are usually suboptimal without the player's intervention.

Auto fighter capabilities:

1. Attack
  - a. Melee: default, available at all times;
  - b. Ranged: requires ammunition and a ranged weapon equipped.
2. Movement
  - a. Movement on the terrain (plane);
  - b. Climbing over obstacles;
  - c. Movement on the terrain (with different elevation levels) – for High Target.
3. Actions
  - a. Pickup item.

### 1.1.5 Boss

The Boss has the following pool of actions:

1. Shoot a ranged weapon;
2. Create an AoE danger effect, that damages the objects in it over time;
3. Fling fighter unit;
4. Take the fighter unit under control.

When taking the fighter unit under control, the Boss can make it do one of the following things:

1. Destroy the weapon it is holding;
2. Attack another unit;
3. Run around (or move towards the Boss);
4. Become temporary intelligent and fight on the side of the Boss

The players can take direct control of the “dominated” fighter. In that case the fighter is under control of the player, however, the input is randomly changed, which makes it challenging to control the dominated fighter for the players.

The Boss will have multiple stages that begin when it receives enough damage.

The Boss will be receiving a low amount of damage at all times from the attacks, and a higher amount of damage when it exposes itself after doing one of its powerful

actions. Specifics for the last two statements will be agreed upon later in the next stage of the project.

The key ability of the Boss that present the most challenge to the players and makes the game stand out is taking the fighter unit under its control.

### 1.1.6 Arena

The action happens in the ruins of an industrial building. The area is a plane with obstacles lying around, which means that there are only 2 dimensions of movement in the game. In the high target the 3<sup>rd</sup> dimension will be added.

The fighter units and the Boss can freely move around the plane. Using the obstacles as a cover from the enemy fire is one of the key gameplay elements.

There is a special area on the arena, which plays an important role: the fighter taken to this zone loses the dominated status imposed by the Boss.

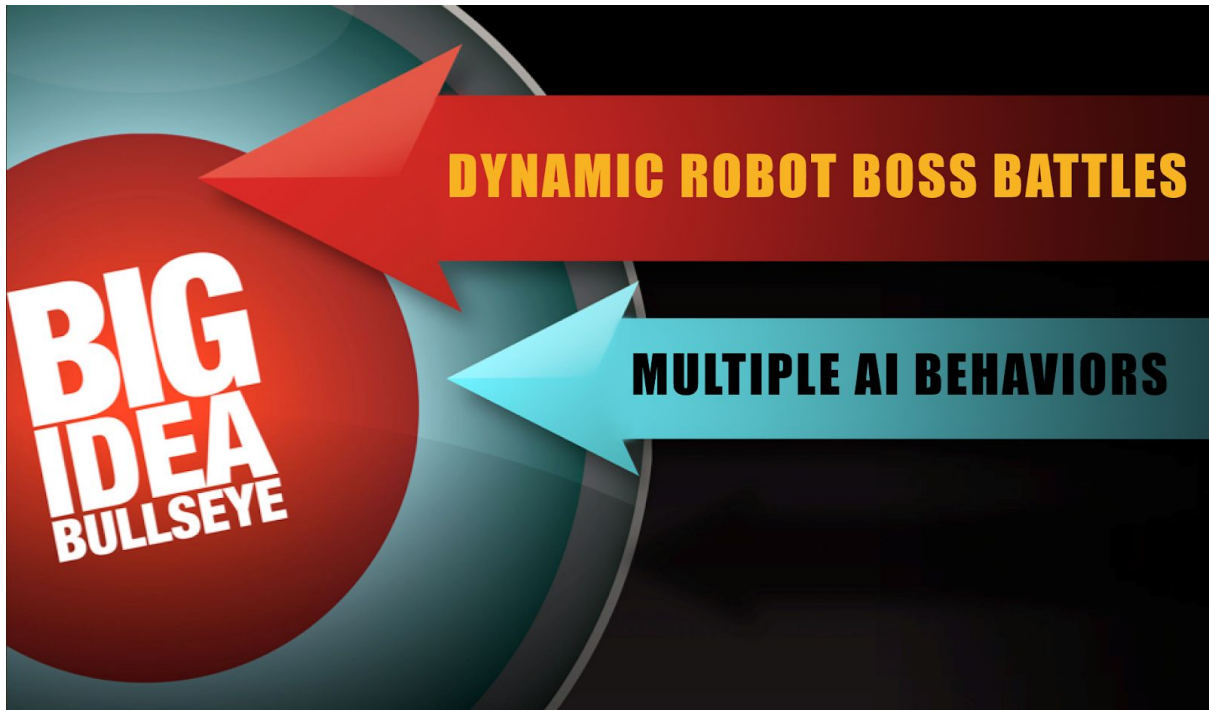
## 1.2. Technical Achievement

Technical achievement consists of making multiple AIs that behave in an interactive and fun way:

1. Boss;
2. Dumb fighter;
3. Smart fighter;
4. A smart fighter that is turned against the players.



### 1.3. "Big Idea" Bullseye



### 1.4. Development Schedule

#### 1.4.1 Layered Development Schedule

##### Functional Minimum:

- Simple environment: an empty room
- Simple melee robots: can move, do a melee attack and take damage
- Player: can select and control a robot, stop controlling a robot
- Xbox/mouse and keyboard input scheme
- Static boss: does not move, attacks random robot in range
- Models: no real models yet, only polygonal shapes to indicate objects

##### Low Target:

- Environment: industrial building
- Melee robot: can dig and pick up items, destroy/drop items
- Items: health pack, stronger melee weapon
- Boss: can influence AI of robots causing them to move around, attack each other or destroy their weapons
- Area to cleanse robots from the boss' influence
- Area to restore health to damaged robots
- Battle has time-limit

- Animation: attack animation for robots and the boss
- User Interface: health bars for robots and the boss
- 3D Assets: a simple model for robots(1 robot type) and boss

Desirable target:

- Boss: can launch a big area of effect attacks
- Boss: has a weak point where it will take additional damage
- Items: ranged weapons with a certain amount of ammunition
- Ranged robot: can pick items, shoot from range, destroy/drop items
- Main menu, win screen, lose screen
- Animation: destruction animation, item spawn animation, pick up animation
- 3D Assets: enhanced robot model with dynamic head lighting(colour indicates the state of the robot)
- Audio: background music and sound effects

High target:

- Boss: can throw robots around / knock them back
- Player ability: giving commands to robots nearby
- 2nd boss with different abilities and behaviour (multiplayer needed)
- Local multiplayer mode: Each player can command one robot
- Different difficulties
- Higher quality models
- More robot types

Extras:

- 3-dimensional playing field with different altitude levels
- Networking: multiplayer mode over the network
- Different boss types

### 1.4.2 Milestones and tasks

1. Milestone: Game Idea Pitch(29.04)

Task Name:	Who	Hours
Brainstorming	All	5
Documentation and Presentation	All	5

## 2. Milestone: Prototype(13.05)

Task Name:	Who	Hours
Defining Type of Prototype	All	3
Creating a Prototype	All	7
Documentation and Presentation	All	5

## 3. Milestone: Interim Results(03.06)

Task Name:	Who	Hours
Melee Robot AI(normal and influenced)	N	10
Ranged Robot AI(normal and influenced)	N	10
Boss AI + Abilities	L	12
Input Manager	L	2
Player Control	L	5
Model Creation: Robot	K	12
Model Creation: Boss	K	12
Model Creation: Environment	K	12
Simple User Interface	N	5
Main Menu	L	3
Level Layout	J	10
Items(Spawn, Pickup, Use)	J	12
Heal and Cleanse Areas	J	4
Documentation and Presentation	All	5

## 4. Milestone: Alpha Version(24.06)

Task Name:	Who	Hours
Boss: Area of Effect Attacks with Weak Point Exposure	L	14

Robots: Additional Behavioral Patterns	N	14
Player: Command Ability	J	10
Background Music and Sound Effects	J	8
Enhanced Robot Models	K	7
Animations	K	10
Balancing of health and damage	All	5
Documentation and Presentation	All	5

#### 5. Milestone: Playtesting Results(08.07)

Task Name:	Who	Hours
Testing and Evaluation	All	10
Bug Fixing	All	4
Implement Feedback	All	2
Documentation and Presentation	All	5

#### 6. Milestone: Final Release(22.07)

Task Name:	Who	Hours
Trailer	All	10
Documentation and Presentation	All	5

## **1.5. Assessment**

A.I.M. is going to be a fun experience for fans of fast-paced, action-packed gameplay that also involves a strategic component as it is found in MOBAs for example. Due to the relatively short play-time, A.I.M. will be playable when you just have half an hour of time, but there will also be achievements and unlockables for more invested players.

The player will have to play strategically in order to defeat the boss in the given time limit, carefully evaluating which items to go after and decide when it is advisable to take corrupted bots to the healing station or when this just wastes valuable resources.

In order to make each fight different, there will be several types of items, each with its own effect, to enhance your bots during their fight in various different ways.

In order to become successful, A.I.M. needs to

1. contain many types of unique items to enhance your legion
2. incorporate the corruption mechanic of bots in a fun and interesting way
3. provide lots of replay value so that it doesn't get boring after your first couple of playthroughs

## **2. Physical Prototype**

### **2.1 Goal of the Prototype**

The main goal of our prototype is to observe if the various behaviours and actions we thought of would be realizable within our game world.

It should let us observe the movement of the robots in our simulation and help us determine what information the robot has, which behaviour it should display and if their desired actions are currently executable. We want to verify that the actions we planned for each behaviour type do not cause any unwanted interactions and that without player interference the behaviour of a robot can change, as we defined state transitions depending on what the robot perceives.

Also, the attacks and patterns of the boss should be tested with the prototype. The focus for the boss lies in which target they will pick, how the attack will be executed and the area which it affects. Moreover, we want to test if the boss can move within our fighting arena considering the possibility of colliding against walls and obstacles.

Another important part of the prototype was to determine if the player has fun playing the game. Because the robots are only chess figures on a paper field, we do not test the actual controlling of the robot but by simulating the robot AIs, the player has to decide which robot they should take control of and what actions they should do with the robot.

### **2.2 Prototype description**

#### **2.2.1 General rules**

Due to the constraints of the table-top nature of the physical prototype, the time flow of the action-based game had to be discretized and the action took the place rounds

that consisted of turns. While this approach takes away from the dynamic action facet of the game, it allowed us to focus on the main objective of our prototype test.

Each round consisted of 3 stages:

1. Player's turn: during the turn, the player is able to pick one of the bots and specify where the bot should move and what action it should take.
2. AIMS' turn: during the turn, the gamemaster is making a movement and an action for the bots that are not controlled by the player based on the current behaviour mode of the bots.
3. Boss' turn: during the turn, the gamemaster makes a movement and an action of the boss based on the current stage of the boss fight and the corresponding available actions set.

Turn structure:

1. Move: the bot can move up to 20 units. The Boss can jump to any part of the Arena.
2. Action: see corresponding sections for the Boss and the Bots.

### 2.2.2 AIMS

Each bot has 5 HP in the beginning and is removed from the Arena after receiving 5 or more points of damage over the game.

Bot's actions:

1. Melee attack
2. Use the terminal
3. Use the item (weapon/healing kit) in hands
4. Dig
5. Throw the item in hands for up to 20 units

Bot's independent thinking:

The malfunctioning AI is represented by 4 behaviour modes. The uncontrolled bot can only be in one of the states at a time.

The player loses after all the bots were removed from the Arena.

#### **Angry**

This mode is activated if the last action made by the player was one of the following: melee attack, use the weapon in hands, use the terminal.

In this state, the bot will continue attacking its last target as long as it's in the range or continue pressing the button.

## **Coward**

This mode is activated if the last action made by the player was one of the following: break the line of sight with the Boss, move away from the boss and no damage to the Boss was dealt.

In this state, the bot will try to increase the distance from the threat and hide behind some obstacle.

*This mode can be activated on the bots that are not controlled by the player every time a low HP bot receives damage.*

## **Digger**

This mode is activated if the last action made by the player was digging an item.

In this state, the bot will be digging for items as long as there are places to dig in its vicinity. When an item is found, the bot doesn't try to make use of the new asset but rather throws it away as it's not interested in items, only the digging activity.

*If there is nothing to dig, the bot goes into Confused mode.*

## **Confused**

This mode is activated if the last action made by the player doesn't fit other modes.

In this state the bot will walk around and trying to pick up items in its close proximity, using them immediately, choosing a random direction if the item needs a target to function.

### 2.2.3 The Boss

The boss fight happens in 3 stages, with each stage being more dangerous and difficult than the other.

#### *0. Warm up (not a real stage)*

At the start of the game, a wall separates the Arena in 2 parts: one with the friendly AIMS, the other one with the Boss. The player has to interact with a terminal on the left wall in order to remove the wall and start the actual fight. This is done in order to give the player a chance to gather all of its AIMS and place them strategically before the actual fight begins.

As soon as the wall is removed, the stage 1 begins and the Boss starts to take its turns.

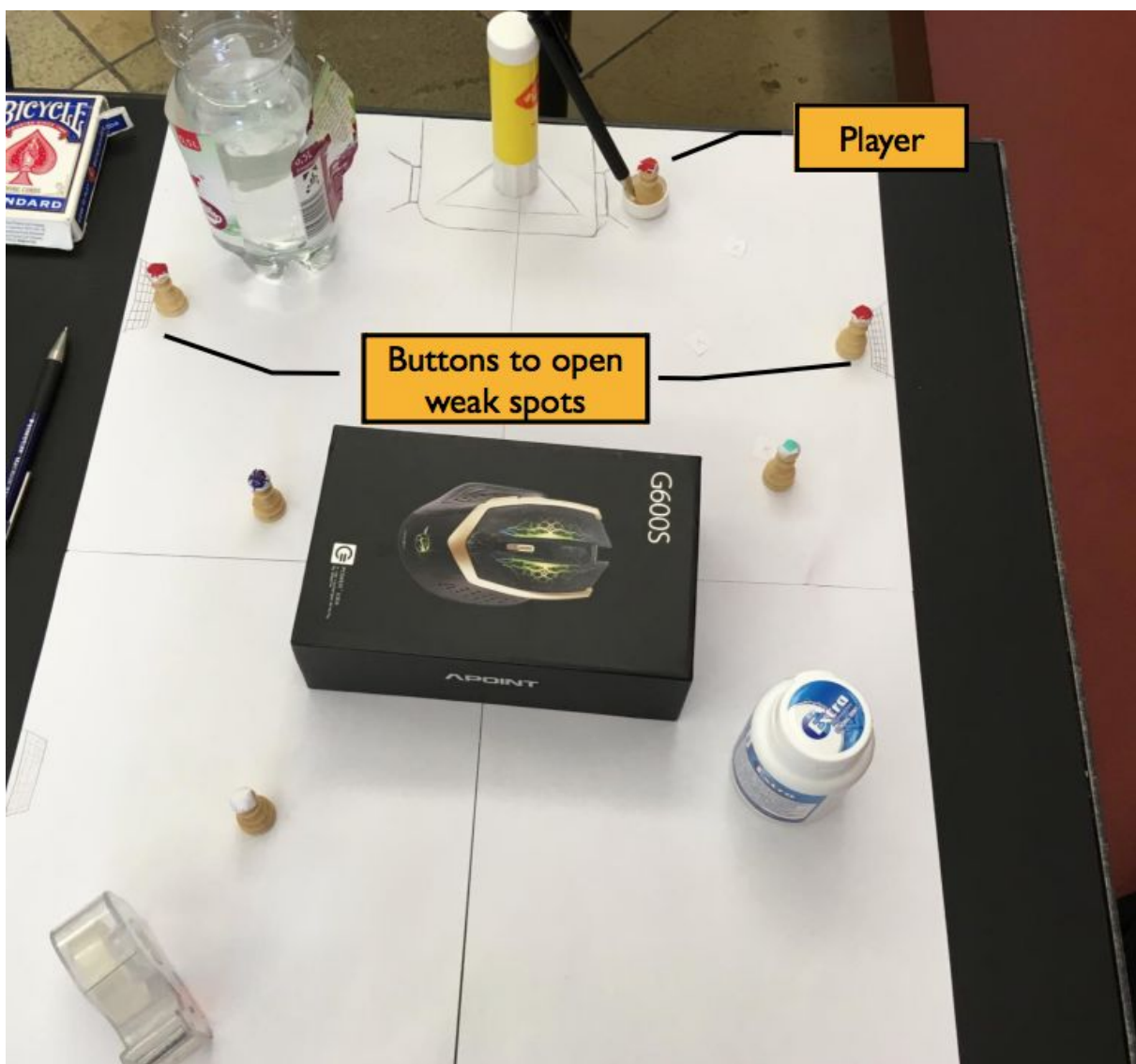
## 1. Overture

In the first stage, the Boss is stationary and is covered with a piece of heavy metal, that protects it from the damage. In order to make the Boss vulnerable to the attacks, the player has to interact with the 2 terminals that are on the walls near the Boss.

The boss always skips the movement part of its turn and does one of the following actions:

1. Machine gun: deal 1 point of damage to all the AIM bots in a cone in front of it
2. Confusing ray: force one of the AIMs to skip its turn
3. Swipe: deal 1 point of damage to all the AIM bots in melee range and knock them back 10 units.

After the 2 consecutive rounds that the Boss gets attacked in its vulnerable points, the stage 2 begins.





## 2. *Main phase*

As soon as the Boss goes into the Main phase, on its turn, it uses its movement action and jumps in the middle of the Arena. This destroys the pile of rubble in the middle and creates 4 additional walls for the player to hide behind.

On its action it can do everything from the previous stage, but gets access to the extra actions:

1. **Recalibration:** take one of the AIM bots in the line of sight under its control. The controlled unit will use its turn to harm the other AIMS to the best of its ability. If the player takes control of such a unit, during the turn the player can only do the movement part of the turn, the action is taken by the gamemaster and is done in such a way that puts the player at the most disadvantage as possible. The effect ends in 4 rounds or if the bot enters the safe zone. The bot has to skip 1 turn in the safe zone before being able to take part in action again.
2. **Large Area Laser Attack (LALA):** deal 2 points of damage to all the bots in a cone area in front of it. This action is taken instead of the Machine Gun.

This stage ends after the Boss has taken 10 points of damage.

## 3. *Endgame*

At this stage, the boss gets access to the movement part of its turn, which gives it an ability to jump at any point of the Arena except for the safe zone.

The player wins after 6 or more points of damage are dealt during this Boss stage.

### 2.2.4 Items

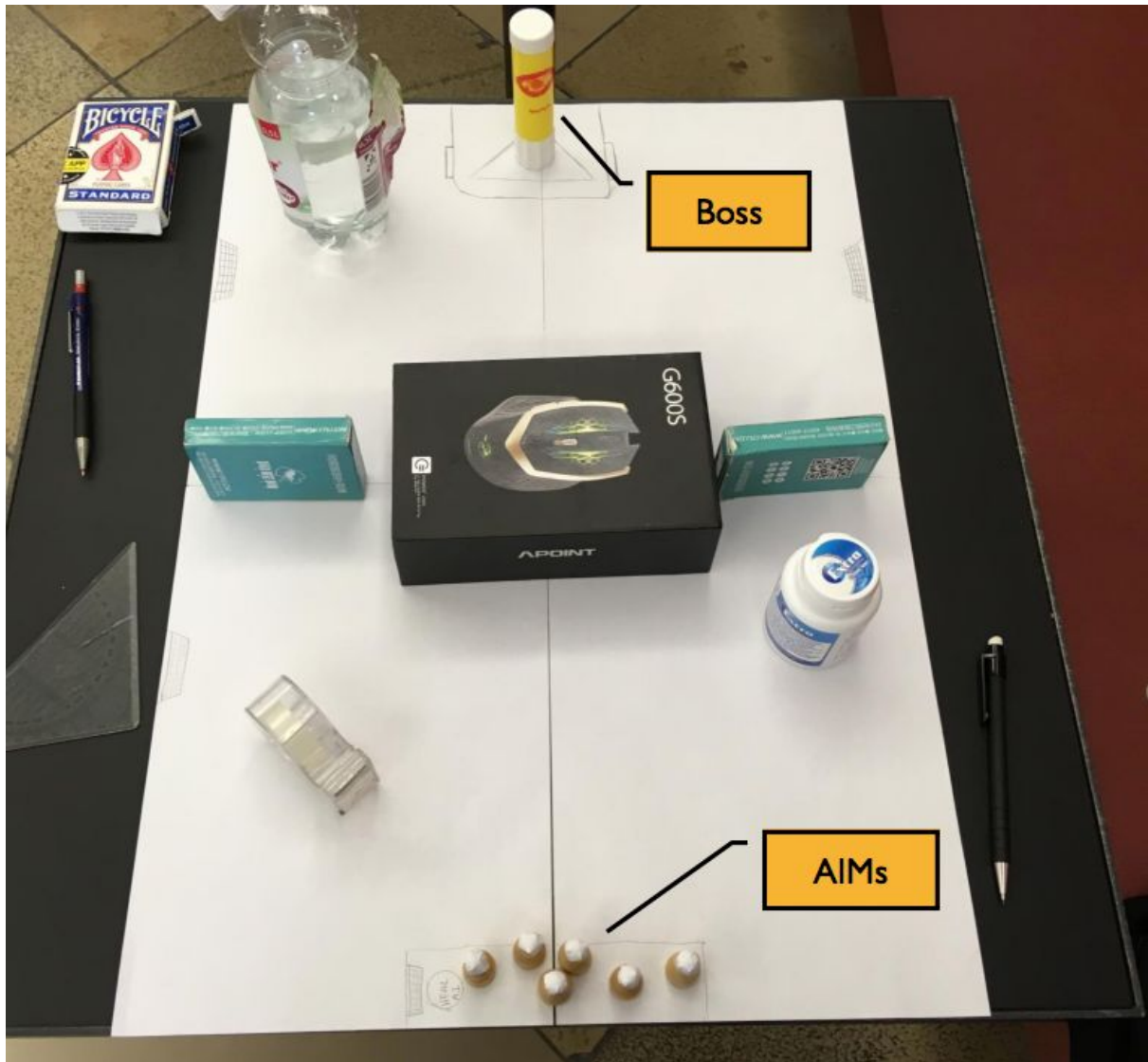
Items that can be found by digging. If the bot doesn't hold any item it automatically picks up the item laying on the ground if it moves over it or if the bot was the one doing the digging. At a time, a bot can hold only 1 item.

1. **Health pack.** Gives the holding bot an action "repair" that allows it to regain up to 3 HP (no more than the maximum amount of 5). The item is destroyed after this action is taken.
2. **Assault rifle.** Gives the player the ability to make an attack action from up to 30 units away. Such attacks deal 1 point of damage and require a direct line of sight to the target. The item is destroyed after the player uses the action 3 times.

When the player completes the "dig" action, 50% of the chance is that nothing would be found, and 25% chance to find a health pack, 25% to find the assault rifle.

### 2.2.5 The Arena

The arena has a rectangular shape. On the short sides of the rectangle, opposing each other, are the Safe Zone and the Boss' base. On the arena, there are various structures that can be used to break the line of sight between actors. The layout of the arena can be seen in the image below.



### 2.3 What we've learnt

Creating and playing our prototype taught us quite a few things about the playability and enjoyability of our game and made us think about how exactly certain parts of A.I.M. would have to play out.

During our main playthrough of the prototype we noticed the following things:

1. In the beginning, it's really hard to get all bots out of the starting position.

When the game starts, we plan to have all bots in the bottom and the boss at the top of the arena. Since they haven't learnt behaviour, yet, they would just wander around aimlessly (no pun intended).

This behaviour isn't necessarily bad or unwanted, since the game is all about training the bots to do the right things, by controlling them, but needs the boss-fight to start out slowly as well. Otherwise, the boss may attack the still behaviour-less bots and deal a lot of damage before the player can do anything about it.

In order for the player to be able to take their time teaching a certain behaviour to each bot, we decided to add a wall in-between the starting position and the boss, so that the behaviour-less bots aren't attacked immediately. However, if the player takes too long, they may themselves randomly wander out of this safe zone keeping the player on their toes.

As we noticed this in the very beginning, the fix was still included in the prototype.

2. The bots, that adapt to aggressive behaviour, tend to in-fight a lot.

Bots, that deal a lot of damage or accidentally hit another bot, while controlled by the player, adopt an aggressive behaviour and attack everything in their vicinity. This feature was meant to be a way of getting a bot into berserk-mode and dealing lots of damage, with the down-side, that if one isn't careful enough, said bot starts attacking other nearby bots if no boss is in sight.

However this in-fighting happened a lot more, than expected and often resulted in clusters of bots fighting each other. Especially the transition from first to second phase, where the boss jumps away from its former position and the third phase, where it moves around quickly, were prone to the currently attacking bots staying behind and battling each other.

The prototype showed us that, in order to make this mechanic fun rather than frustrating, we need to be careful how and when a bot starts attacking other bots. Two options we came up with are either just making a bot get "mindlessly" aggressive, i.e. also attack friends, if the player, while controlling said bot, accidentally hit another one or having two different levels of "raging": A weaker one, where the bot just attacks the boss and a stronger one, where

the bot dealt so much damage when controlled by the player that they're now so blood-thirsty they even attack allies.

## 3. Interim results

### 3.1 A.I.M Units

A.I.M. units carry out both the avatar of the player and NPC functions. When the player doesn't control the unit, it is left to its own suboptimal or even malicious reasoning.

#### 3.1.1 Player controls

The player controls are very simple to grasp. There are 2 distinct modes that the player can stay in: bot selection and direct control. During the bot selection the camera zooms out, giving the player an overview of the battle and the ability to choose the bot to control. After selecting the bot, the camera starts to follow the selected unit and the player goes into the direct control mode in which the player gets an access to all the actions a bot can do:

1. Move (run or walk)
2. Dodge (the bot makes a roll)
3. Make melee attack
4. Use a held item (shoot a rifle or use the health pack)
5. Throw away the held item
6. Dig near the digging spot

#### 3.1.2 AI

For AI pathing we are using Unity NavMesh system. Each bot has a NavMeshAgent component attached to it, that we query to get the next step of movement. The current implementation has a room for improvement, as sometimes the actual model of the bot gets left behind by the NavMeshAgent and the bot can get stuck by a wall or a laying ragdoll. This difficulty comes from the fact that we are using root motion animation for a smoother feel of bot movement and is next on the "to be fixed" list. Otherwise the pathing and collision avoidance provided by the Unity is sufficient for our game and works fine.

Currently there are 5 AI behaviors implemented for our AI bots.

##### 3.1.2.1 Confused behavior

This is the default and useless behavior. The bot slowly wanders around the arena aimlessly, making stops to look around and relax a bit. It might choose to pick up an item if it feels like it (random based).

### 3.1.2.2 Coward behavior

In this behavior the bot is scared of everything. As soon as it takes damage it will run away from the source of damage as far as possible. If there is no source of damage, the bot will run around in panic. If the bot hasn't received damage in a while (currently it's 20 seconds) it will go to the default state.

### 3.1.2.3 Aggressive behavior

In this behavior the bot is feeling like taking an action. It will choose the closest target it can interact with (the interactable objects like console, the friendly units or the boss) and will try to attack it with what it has. If it holds a harmless object, it will drop it and go with melee attacks. When the bot is melee, it will first close the distance with the target and the attack it. If the bot has a ranged weapon it will try to get in range with the target and the proceed to shoot, not paying attention to the possible obstacles between the target and itself.

If it wasn't able to attack anything in 3 seconds it will look for a new closest target, potentially changing its attention. If it wasn't able to attack anything in 15 seconds it will go into the default mode.

### 3.1.2.4 Evil behavior

Currently evil behavior acts just as the Aggressive behavior except that it picks only other bots as the victims.

### 3.1.2.5 Stay and stare behavior

This hardly an AI behavior: under this behavior the bot will stay in one place and follow the provided target with its rotation.

The AI logic is implemented as a state machine with defined transition from one state to the other and a set of scripted actions performed in each step.

As the game action takes places in a limited space arena and at all times a big part of the arena is visible to the player, we that it is okay for the bots to have all information about the level without masking based on vision, proximity, etc. This makes the development easier and doesn't take away from the game quality.

## 3.1.3 Behavior adoption

When the player has a direct control over the bot the bot pays attention to what the player does. Depending on the last actions of the player the bot will choose the behavior to follow after the player released the direct control.

It is implemented in a form of "behavior-meters". Each potentially obtainable behavior has its own progress scale that decays over time. If the player takes an

action that corresponds to the behavior, the progress will increase. For example, if the player uses an attack action, the value of the aggressive behavior's progress will increase.

Currently there are 3 behaviors that can be adopted:

1. Aggressive behavior: using attacks increases the progress towards this behavior.
2. Coward behavior: running away from the boss increases the progress, running towards the boss decreases it. If the player breaks line of sight with the boss, the progress increases dramatically. The opposite happens when the player brings the bot in the line of sight with the boss.
3. Confused behavior: if none of the potential behaviors exceed the necessary threshold value for the progress the confused behavior will be chosen.

## **3.2. [L.A.M.P.] Lethally Armed Management Program**

Our antagonist is the Lethally Armed Management Program who was the overseer of a robot producing facility. Their task was optimizing the production pipeline and ensuring the quality of the products. But due to the extreme policies of the company and several failed products the facility was abandoned. Now the Management Program still dwells in the facility and tries to eliminate all the defective products remaining there. It has a variety of different attack options and decides, depending on the situation, which ones are more advantageous for them.

### 3.2.1 Currently implemented attacks

#### **1. Spread Shot:**

When many bots are clumped up, the L.A.M.P. rotates its gun at a certain angle around the center of the group. The area in which A.I.M. bots can be hit by bullets is shown in the form of a triangular danger area on the ground.

#### **2. Snipe:**

This attack is more likely to appear when there is a bot with low HP. The L.A.M.P. aims with his gun at the bot with the lowest HP for a while and then shoots out a sniper bullet which is faster and does more damage compared to the normal gunshots. While aiming the trajectory of the projectile will be shown through the laser of the sniper. The player can avoid this attack by moving the targeted bot out of range or block it by moving another bot in front of the target.

### 3. Eye Slam:

As bots are continuously attacking the L.A.M.P. they will slam their eye in rage on the ground. Which does little damage but knocks bots away and (sends them into a confused state).

### 4. Bouncy Projectile:

A large ball like projectile flies out of the eye of the L.A.M.P. and bounces slowly around the battlefield. In the beginning, it is quite harmless as it is slow and does not much damage but over time the bouncing projectile accelerates and also its damage increases. Therefore, it is advised to get hit by it as soon as possible. This means the player needs to take control of a unit and intercept the bouncing trajectory of the ball, especially because the L.A.M.P. aims the projectile in a way where it tries to avoid hitting A.I.M. bots.

### 5. Electrical field:

Within the abandoned facility are still some power generators hidden. The L.A.M.P. can activate them and with those generators, they can charge up an electrical field which increases in size over time. The player has to hit the generators in order to deactivate them, but this is easier said than done because they have a security measure which only allows them to be deactivated if they are hit quickly in succession in a specific order. We plan on randomizing the order and displaying it on the eye of the L.A.M.P., but currently, we have a fixed order of pink->blue->green.



### 6. Sawblades:

As the name suggests the Lethally Armed Management Program is equipped with a multitude of dangerous weapons. One of them are the sawblades hidden within their



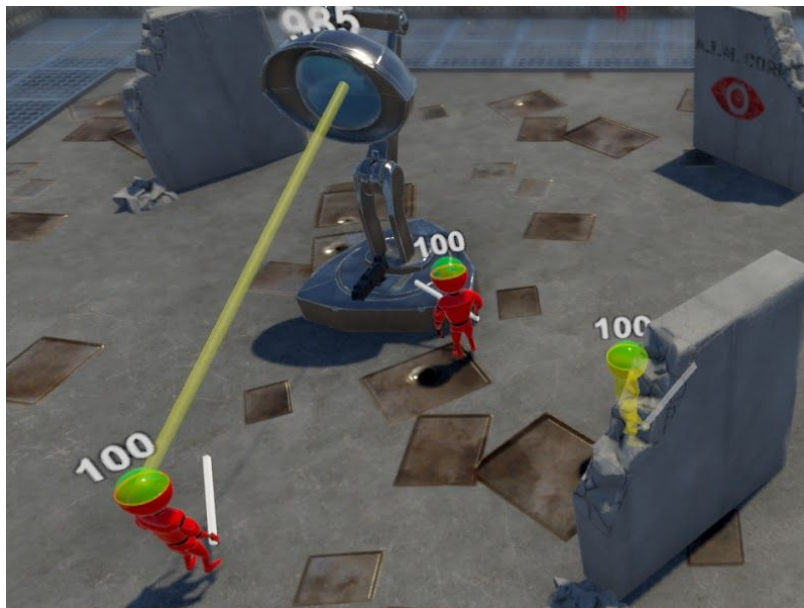
base. They continuously move the sawblades in and out of the damaging and pushing back bots. This punishes melee bots who are in an aggressive state. For the interim, we do not have a saw blade model yet that is why we are using a mock saw blade, built from unity objects.

### 7. Corruption beams:

The L.A.M.P. can also corrupt the behavior of the A.I.M. bots. For this they will look at the bot they want to corrupt and a beam will be shot from their eye to the head of the bot. The whole corruption process takes a certain amount of time depending on the corruption. During the procedure, the affected bot will do nothing and only stare at the boss. After finishing the corruption procedure the bot will adopt a new malicious behavior.

DoNothing beam: This beams stops any behavior of the bot and forces them to stand still and do nothing.

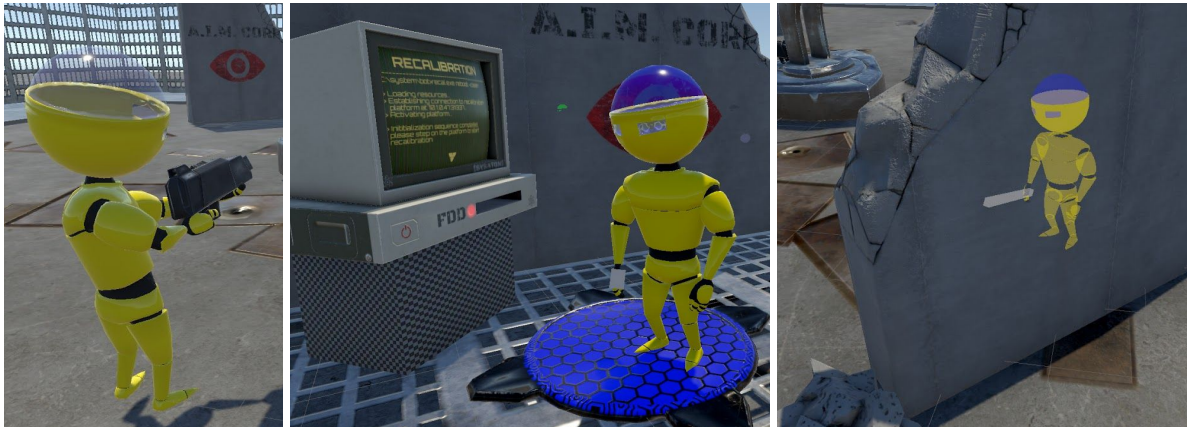
Evil beam: This beam fully corrupts the bot making them attack friendly bots



### 3.3. Models and Environment

There already exist models as well as some animations for the important game objects, i.e. a model for bots, the boss as well as an arena with obstacles. The bot model has been animated using Mixamo ([www.mixamo.com](http://www.mixamo.com)), an free online service providing automated rigging and high-quality retargetable animations. The arena also contains a functioning recalibration terminal, that can cure bots hit by the L.A.M.P.'s 'Evil Beam', as well as digging spots where a bot can dig up a random item. The latter ones spawn randomly in walkable spots of the arena every ten to

thirty seconds. So far the items that can be dug up are a health pack and a rifle. To make it easier for the player to keep an overview, a shader has been implemented to visualize bots that are occluded by obstacles in the arena.



### 3.4. Progress in the Development Schedule

Summarizing the above we managed to completely cover the functional minimum and low target we set ourselves at the game-pitch except for the time-limit on the boss-fight, which we dropped. Also the display of a bots health as numbers over their heads is temporary as we want to include the information of a bots remaining hitpoints in the model.

As to be expected, we didn't start implementing the items on our high target or extras list, yet.

Concerning the desirable target, we managed to implement several of the there listed points:

- Boss: can launch a big area of effect attacks

As seen above a lot of attacks (including an AoE attack) for the boss are already included in our current version

- Items: ranged weapons with a certain amount of ammunition

The rifle obtainable from the digging spots is fully functional, although this point can't really be called 'complete' as of now as there exists only one ranged weapon

- 3D Assets: enhanced robot model with dynamic head lighting (colour indicates the state of the robot)

For now the behavioural state of a bot is represented by a material color change of its head. Later the color of a light source inside the head will indicate the bot's state. As mentioned we intend to add a visual way of indicating a bots health as well, which could be the intensity of the light.

### **3.5 Changes to Development Schedule**

The aforementioned health pack replaces the healing area mentioned in the low target. Also, the time limit for the battle has been discarded since a playthrough is expected to be short and fast paced anyway.

## **4. Alpha Release**

### **4.1. A.I.M. Units**

#### 4.1.1 Player controls

There were no radical changes to the player controls, only improvement on the initial implementation: the support of X-BOX controller was added. Also the visual part of camera transitions and marking the selected bot was improved: now it feels more natural and less obstructive to the actual gameplay.

#### 4.1.2 AI

The problem with NavMeshAgent position getting separated from the actual bot model was solved by creating a wrapper around the Unity's NavMeshAgent, that handles all the calls to the AI pathfinding and takes care of synchronisation between model position and NavMeshAgent position. Also the NavMesh generation parameters were improved, which leads to cleaner pathfinding. The bots still get stuck on the scene objects in rare cases however in the interest of time saving and higher priority tasks the work on the obstacle avoidance had to be skipped. Even if the bots get stuck it is not game breaking and the player can always take over and help the bot with its task.

The behaviour logic was significantly improved and made more complex. One of the significant changes that affect the gameplay is that the bots now can switch to the Coward behaviour when being hit, with higher chance of transition when the bot is low on health. For the corrupted behaviours the chance is greatly lowered but the transition can still happen.

#### 4.1.2.1 Confused behaviour

Now the confused bots have higher chance to pick up a lying item. If they do so, after some time they will use it pointing at random direction and throw it away afterwards.

#### 4.1.2.2 Coward behaviour

The change since the last report was to improve the running away part. Also the scared bots if they have not received damage in a while will try to pick up and consume health packs.

#### 4.1.2.3 Aggressive behaviour

The logic behind the aggressive behaviour hasn't changed, but the implementation was improved. To perform melee attacks now the aggressive bots ask their target to provide them "parking spots", which they use to determine from which position they can hit the target. This change was necessary to give us more precise control on how the bots will be hitting each other, interactable objects and so on. Also the aiming of the bots with ranged weapon was reworked. Now the proper rotation of the bot in order to hit target is computed using homemade inverse kinematic algorithm and the bots lead with their shots on the moving target.

#### 4.1.2.4 Evil behaviour

Evil behaviour has been changed quite a lot. In the current version evil behaviour has its own custom logic for ranged attacks: it takes into account the obstacles and tries to find a position from which the target is visible. Also if there is a weapon nearby the evil behaviour will pick it up in order to deal more damage to the friendly bots. When the player tries to control the evil bot the movement direction input is rotated with random angle, making it much harder to control the corrupted bot. If the evil behaviour has a gun when the bot is controlled by the player, it will override the rotation of the bot trying to shoot the friendly bots.

#### 4.1.2.5 Stay and stare behaviour

No changes were made for this behaviour.

## **4.2. [L.A.M.P.] Lethally Armed Management Program**

### 4.2.1 Boss phases

Now the boss fight starts after the player opens the door and steps outside with at least one bot. The whole battle is divided into three phases. With each phase transition, the attack rate of the boss increases and new attacks will be unlocked.

**Phase 0:** Before the player opens the door the L.A.M.P. is still inactive as it did not detect the presence of the A.I.M. units yet.

**Phase 1:** The boss notices the defective units and starts its execution program. In this phase, he has 6 different attacks to choose from and takes full damage.

**Phase 2:** After taking a third of its maximum hit points the boss transitions into the second phase. They gain protective armor which halves all damage. Fortunately for the player buttons exist in the arena which can lift up those protections for a short while. The boss also gains 4 new attacks, including its ultimate attack called LALA.

**Phase 3:** When the boss reaches its last third of health they start jumping around the arena causing chaos and confusion on the remaining A.I.M. bots.

## 4.2.2 Boss Attacks

### 1. Spread Shot:

This attack did not change much from the interims. The gun rotates in a certain angle around and shoots slow bullets each dealing 4 damage to a bot.

### 2. Snipe:

This attack also stayed the same. The speed of the bullet was increased as it was to easy to dodge. The sniper bullet does 10 damage.

### 3. Eye Slam:

The force for the knockback has been adjusted and it also stuns bots now because they need a certain amount of time to stand up. When bots get hit by the Eye Slam they take 5 points of damage.

### 4. Bouncy Projectile:

The initial speed and damage of the bouncy projectile have been reduced but the maximum stayed the same. This means it takes longer to reach its full potential, but

the maximum damage of 29 damage is a lot. Also to indicate the increase in damage not only does the projectile increase in speed but also in size;

## **5. Electric Field:**

Now the pillars for the electrical field do not have to be hit in a certain order. As long as the bots activate all pillars at the same time the field will vanish. During the development playthrough, we found that the player could ignore the electrical field and just keep attacking. To give the player the ability to ignore the field for a short time, but force them to deactivate it in the long term, the field has 3 areas now. The outer area which damages the bot only by 0.5 per second, the medium one which does 0.8 damage per second and the inner one doing 2 damage per second.

## **6. Destruction disks**

The saw blade attack which was supposed to damage and push away bots was changed into the destruction disk attack. Through the long reach of the melee weapons, the bots usually do not stand in the range of the sawblades rendering the attack useless. Now the boss instead shoots multiple waves of destruction disks with a higher range and with a faster speed. These disks do not push to bots away but deal 3 damage per wave. This attack is only available during the first phase.

## **7. Corruption beams:**

The basic corruption beam attack of the boss did not change, but we refined the malicious behaviors.

## **8. Stomp**

The new Stomp attack knocks all bots around the L.A.M.P. back and deals 5 damage. Each bot hit by this attack will also become confused.

## **9. Jump**

In the third phase, the boss starts jumping around the arena. They can jump to one of the 4 center walls or the arena center. Upon landing it knocks back and confuses bots.

### **10. Large Area Light Attack(LALA)**

The Large Area Light Attack which was previously referred to as Large Area Laser Attack is the ultimate attack of the L.A.M.P. It needs 10 seconds of charge up time. Afterward, it illuminates the whole arena with a blinding light causing all the bots to panic and become confused. Bots in the targeted zone which got hit by the LALA take 30 damage, adapt the evil behavior and become uncontrollable by the player. If bots are in the area the player can hide them behind walls so they do not get hit by the attack. After the light fades the intense energy of the attack leaves small electrical fields on the ground. These electrical fields cause the bots to do nothing and become uncontrollable by the player. But this attack also offers a golden opportunity when handled correctly. The attack overloads the processor of the boss causing the eye to fly onto the ground. The player can target the eye to deal extra damage to the boss. After the boss recovers the electrical fields vanish and the influenced bots wake up from their malicious behavior confused.



### 4.2.3. Attack determination

The L.A.M.P boss uses a point based system which determines the likelihood of each attack. Depending on the positioning of the A.I.M. bots, their current health, the health of the boss or the amount of damage they took in the last x seconds, the boss has different priorities to each attack. Also after using an attack, they go on a cooldown meaning the boss cannot use the same attack in immediate succession. Also, some attacks need certain conditions to be fulfilled in order to be used, for example, the boss can only activate its LALA attack when it is in the middle.

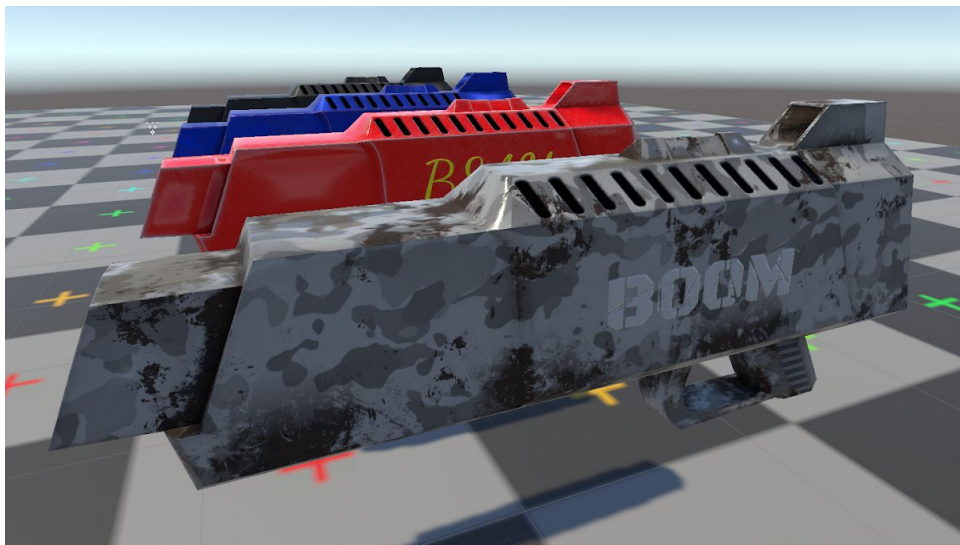
## 4.3 Items

There are currently six items obtainable in the game.

### 4.3.1. Weapons

Four of them include a rifle, a sniper, a rocket launcher and a laser gun. While the former two shoot regular bullets, the third will fire a rocket that explodes on contact with another object and cause massive damage to all nearby entities. The laser gun continuously fires a laser beam when the attack key is held down.

All items have an ammunition count / a maximum number of uses after which they disappear.





### 4.3.2. Health Pack

A simple item granting 25 health points to each bot that uses it.

### 4.3.3. Bomb

This item will explode 5 seconds after spawning.



### 4.3.4. Digging Spots

Digging Spots have been changed in that they are now spawned with a random item out of a given pool with a given probability distribution over the items of how often they will show up.

## 4.4 Tutorial

In order to make it easier to learn the core controls and goals of A.I.M., a tutorial has been added. It consists of several small scenes, each intended to teach one (or a couple of) concept(s) of the game to a new player. There's usually a small task to solve making use of only some gameplay elements / controls, task and needed knowledge are presented to the player in a small textbox.

Currently, there are three tutorial scenes available. Solving one will progress the player to the next.

### 4.4.1. Scene I

Teaches:

- switching between bot control mode and bot selection mode

- hitting (buttons in this case)

Setup:

Three bots are in three different cells, each with a button, one with a door. Task is to open the door for which each button needs to be hit. In order to do this, one needs to play as each bot once and hit the respective button.

#### 4.4.2. Scene II

Teaches:

- digging
- fighting
- malicious behaviour

Setup:

One bot is in the starting room, which contains three digging spots as well as a door to the next room. The latter contains an uncontrollable, malicious bot that has a weapon and needs to be killed to progress. Therefore, a weapon from the digging spot would be handy.

#### 4.4.3. Scene III

Teaches:

- bot moods
- attacking the boss

Setup:

Five bots are in a big arena with an invincible, non-attacking boss in the middle. Some of them are aggressive, some confused, some craven. Task is to get them all aggressive and targeted on the boss at the same time, before they kill each other.

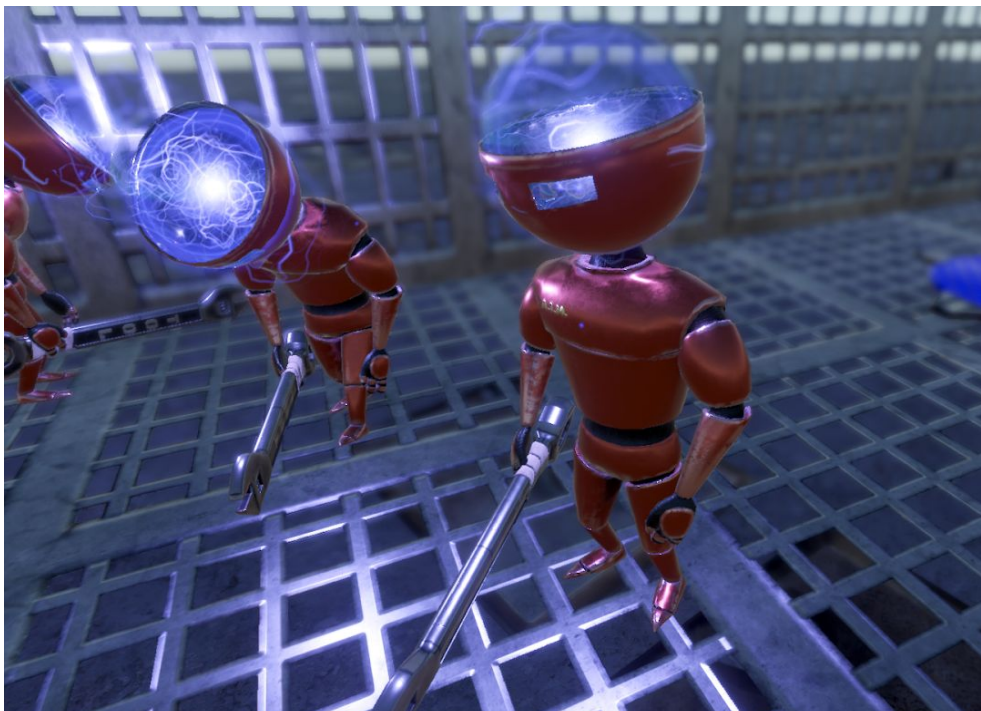
## 4.5 Sounds

Sound effects were added to the game. Currently in the game the boss and bots voice overs are integrated to the gameplay. The sounds for gunshots and environment events are played, although only with placeholder audio clips.

To overcome the limitation of the Unity's audio API an extra layer for sound management was implemented. The sound management subsystem takes care of scheduling sound clips, managing different audio sources (tracks) and checks that the sound clip collections are properly configured in the scene.

## 4.6 3D Art & Effects

Almost every object in the game now has a dedicated 3d model with textures. A particles systems for the bot head, boss eye, and boss corruption beam have been created from scratch. Additional particle effects have been added from Unity's particle pack ([Unity Particle Pack](#)), and modified for our needs.



## 4.7 Progress in Development Schedule

Our development pretty much exactly hit the desirable target from the schedule. Weak point of boss, menus, several missing animations and textures as well as sounds were all added to the game. Our main priority in the second half of the development was to polish everything we had plus what we added to obtain a playable game. We added a couple of extra items and boss attacks, for example,

and improved on the bots' AI systems. Therefore, we didn't venture far out into the high target, yet. We only have better quality models as well as the boss knocking around bots.

## **4.8 Changes to Development Schedule**

Several adjustments were made to the layered development schedule, differing from the original concept. The ability to shout commands to nearby bots (desirable target) has been discarded, since it counteracts one of the game mechanics, that is having multiple bots that all can be directly controlled by switching between them. Also, the idea of having multiple robot types was replaced by expanding on the bot AI with a multitude of different behaviours as described above.

## **5. Playtesting**

Before starting with the playtesting, we corrected some faulty behaviors, added a tutorial in order to improve the experience of the playtester and created a feedback form with the most important questions.

### **5.1 Tutorial**

In section 4.4 we explained what we had of the tutorial at the time of the alpha release. The basic game-play was being explained, including selecting and switching bots, movement, attacking and digging up and using items as well as the behaviour system described in section 3.1.2.

In order for the game to be understandable for new players, though, we felt it necessary to have the more complicated boss attacks explained, so there we're four more tutorials added, three of which made it into the build used in our tests.

#### **5.1.1 Scene IV**

At the beginning, a single bot is in the scene, together with the L.A.M.P. The latter uses its corruption-beam attack on the former, making it evil and thus hard to control. At the end of the attack, three additional bots spawn that the corrupted bot (if it has a weapon at hand, which it spawns with) attacks.

Task is to get the corrupted bot to the recalibration terminal, purifying it from the L.A.M.P.s influence.

### 5.1.2 Scene V

The electric field attack is cast in the beginning of the scene and task is, to use three of the four available bots to break it.

### 5.1.3 Scene VI

The L.A.M.P. is spawned with its protection from the second phase. There are three buttons in the scene, each of which, similar to the three buttons in the main scene, lifts the bosses protection from one side for several seconds. Goal is to lift the protection off of each side once and hitting the bosses weak point while it is up.

### 5.1.4 Scene VII

Three bots and the boss are spawned inside a circle of several broken wall pieces. The boss performs the L.A.L.A. In order to complete the scene, the player needs to get all bots to safety, either hiding behind a wall or behind the boss and then hitting the bosses eye-weakpoint, that comes down for a while after each perform of the L.A.L.A., once.

If a bot is damaged or gets corrupted by the electric residue (see section 4.2.2.10 for an explanation of the L.A.L.A.), the scene needs to be retried.

However, since the L.A.L.A. is our best attack, we decided to remove this scene in order for it to be a surprise.

## 5.2 Conducting the playtesting

We conducted the playtesting mostly with friends and colleagues in private but also prepared a major playtesting day on July 5th from 1 pm to 6 pm at the university in Garching. We advertised our playtesting session in chat channels of the Games Engineering study and also talked to various people in the main hall of the university. For this playtesting session, we prepared snacks such as chips and cookies, and various soft drinks. We also prepared multiple stations where the testers could play. Unfortunately not a lot of people appeared on that day, but all in all, we managed to conduct 17 playtests.

The general structure of the playtest looks like this:

Welcome and short introduction	~5min
Playing the tutorial	~5min
Playing the main game	~10 - 25min
Filling out the feedback form	~5min

Casual talk about the game experience	~5min
---------------------------------------	-------

In the beginning, a short introduction of the game is given. We explain the general setup of the game and what the goal of the player is. We specifically did not mention any details of any game mechanics as they should learn these by playing the tutorial. The duration of the main gameplay phase varied a lot between playtests depending on the abilities of the player but we gave every tester at least 2 tries to defeat the boss. We also noted when a player had issues during the play session or moments where they had to ask us in order to understand a mechanic. After completing the gameplay phase they were given the feedback form and afterward they had the chance to talk to us about their playing experience. This talk gave us further insight into the thoughts of the playtesters and they gave us more direct feedback.

### 5.2.1 Feedback form

The feedback form was created with Google form which offered different answer types such as multiple-choice-questions, text answers, time answers, linear scales. The whole form is split into 6 different sections. The following is a summary of the questions in the form including the answer type in the brackets.

1. User Data
  - a. Gender [Multiple-Choice]
  - b. Age [Multiple-Choice]
  - c. Rate how well versed you are with video games [Scale]
2. Gameplay
  - a. Were you able to defeat the L.A.M.P.? [Multiple-Choice]
  - b. If yes, how long did it take? [Time]
  - c. How many bots were alive at the end? [Multiple-Choice]
  - d. How much HP did you have left [Short Text]
  - e. How difficult was the game? [Scale]
  - f. Why do you think the game was too easy/difficult? [Long Text]
  - g. If there was one, explain a situation during the playtest which confused you. [Long text]
  - h. How did you like the pace of the game? [Scale]
  - i. If you did not like the pace, how would you change it? [Long Text]
  - j. How skill or chance based do you think the game is? (According to the matrix above) [Scale]

- k. In which way does the game challenge the player? (According to the matrix above) [Scale]
  - l. You have placed the game on the matrix above. Is this where you would like the game to be? If not explain where it should be. [Long Text]
  - m. Describe how you would change the game in order to improve it. [Long text]
3. A.I.M. Bots
- a. How many different A.I.M. behaviors did you see? [Multiple-Choice]
  - b. Were you able to understand the factors for some of the behavior transitions? [Multiple-Choice]
  - c. Describe what could trigger a behavior transition. [Long Text]
  - d. Rate the Almost Intelligent AI of the bots. [Scale]
  - e. How would you improve the AI behavior? [Long Text]
  - f. Rate how good the controls felt. [Scale]
4. L.A.M.P Boss
- a. Mark the attacks you liked. [Checkboxes]
  - b. Mark the attacks you disliked. [Checkboxes]
  - c. Explain why you did not like certain attacks. [Long Text]
  - d. Rate how much you liked the LALA attack. [Scale]
  - e. Tell us which aspects of LALA you liked or disliked. [Long Text]
  - f. Describe how you would improve the boss. [Long Text]
5. Items
- a. Did you dig up any items during the boss fight? [Multiple-Choice]
  - b. Rate how much you liked the digging feature. [Scale]
  - c. Which item did you like the most? [Multiple-Choice]
  - d. Why did you like it? [Long Text]
  - e. Which item did you like the least? [Multiple-Choice]
  - f. Why did you not like it? [Long Text]
6. Final Remarks
- a. Rate your overall game experience. [Scale]
  - b. Which aspect of the game did you like the most? [Long Text]
  - c. Which aspect of the game did you like the least? [Long Text]
  - d. Describe how you would improve the game. [Long Text]
  - e. If you have any other remarks which did not fit to any questions above, please state them here. [Long Text]

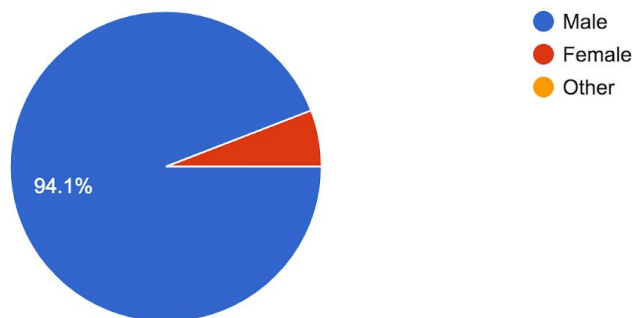
## 5.3 Playtest results

In the end 17 people have tried to play our game and filled out the feedback form. The statistics from the feedback form are presented and contemplated in this section.

### 5.3.1 Demographics

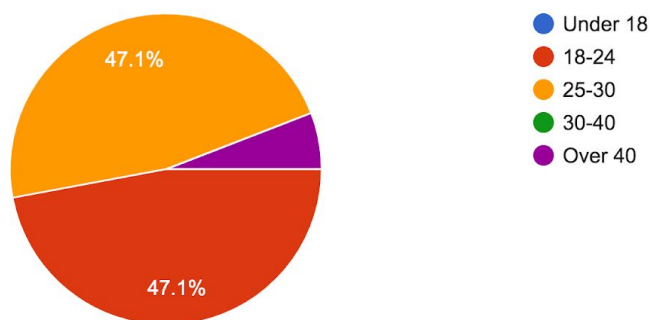
#### Gender

17 responses



#### Age

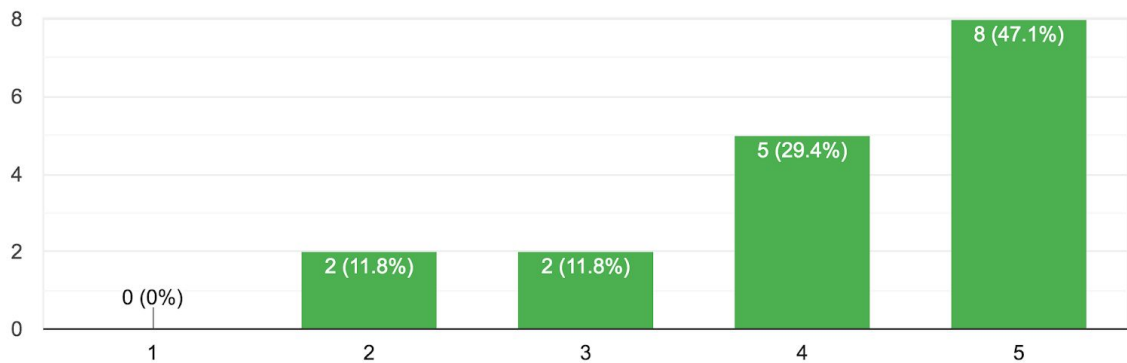
17 responses





## Rate how well versed you are with video games.

17 responses

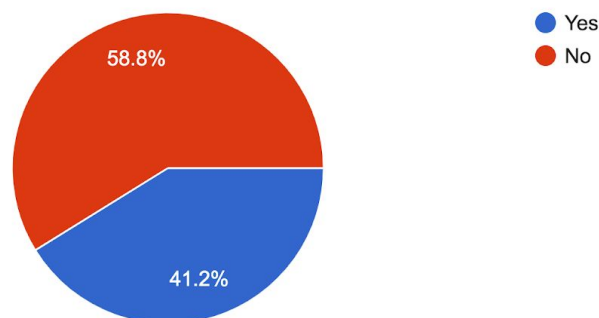


The majority of the respondents were male, with the age between 18 and 30. Most of the participants were gamers with medium to high experience of playing video games. This is not surprising at all, as the people who were interested in testing the game were students of TUM and/or friends of the team members.

### 5.3.2 Gameplay

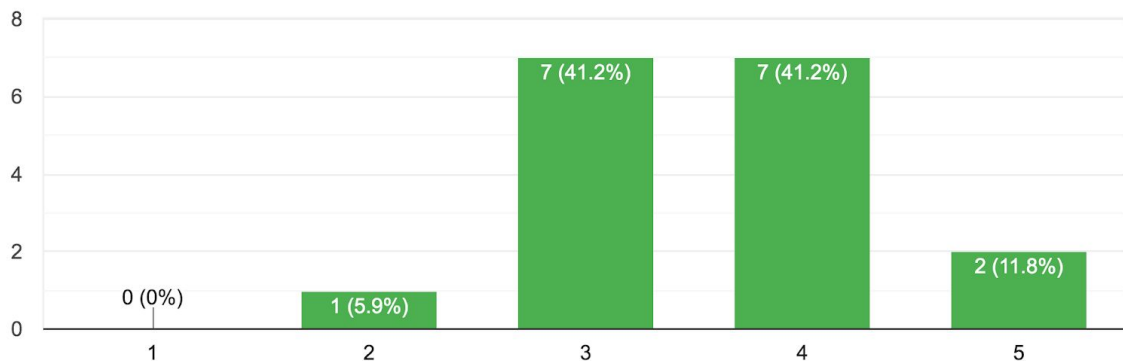
#### Were you able to defeat the L.A.M.P.?

17 responses



## How difficult was the game?

17 responses



Only 41% of the participants reported to have completed the game. The time to kill the boss varied a lot with 3:21 being the fastest result and 15:38 being the slowest one. The average completion time is 7:18.

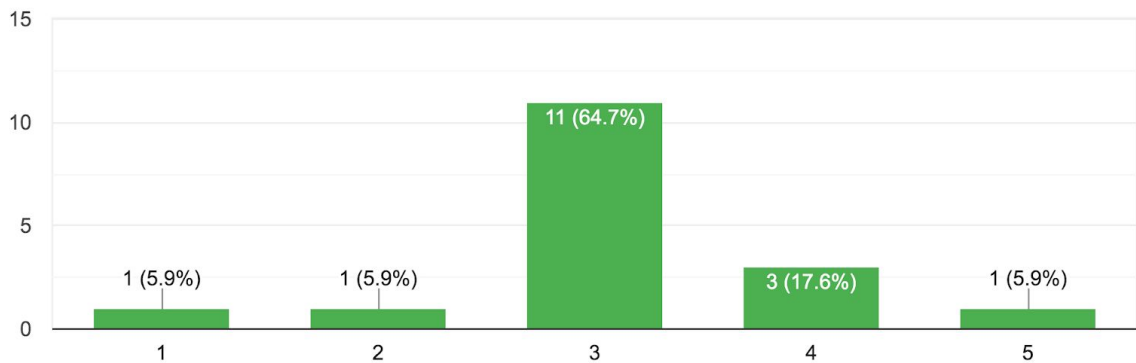
With the fail rate it's not surprising that people considered our game to be challenging, however only 2 players thought that the game is too much difficult. We consider this to be a good result, as we were aiming towards challenging and fun boss fight, that might take a couple of tries before successful completion.

Some of the players complained about unresponsive controls, which was something that we expected, as we went into alpha with a little flawed control system. Others said that the learning curve is too steep, but also multiple people said that the boss itself was challenging.

Most people were confused by the controls: button mappings and color-coding were too much to remember for some players during the tutorial. We believe that this is inevitable when you try to have deep player mechanics packed into a short game, as the usual way to handle the complex gameplay is to introduce the player to new elements over a longer period of time. Another major issue for the players was the unfortunate level design mistake that we did. The tutorial introduced the "weak spot" mechanics, showing that the player has to interact with buttons in order to open the shielded parts of the boss. However, in the actual game the shields appear only during the seconds phase. This has led to players trying to hit inactive buttons on the wall during the boss fights without making any effect on the fight.

## How did you like the pace of the game?

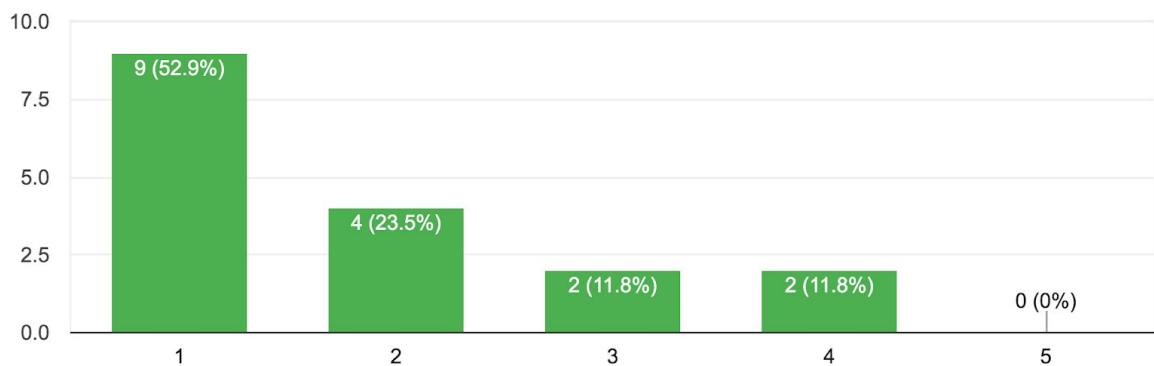
17 responses



We are pleased to see that the majority of the players have found the pace of the game to be optimal, with some people finding it too fast and some people find it too slow. The people who found it slow complained that the bots are not moving fast enough and that the controls are somewhat unresponsive, however the pace of the fight is “fine”.

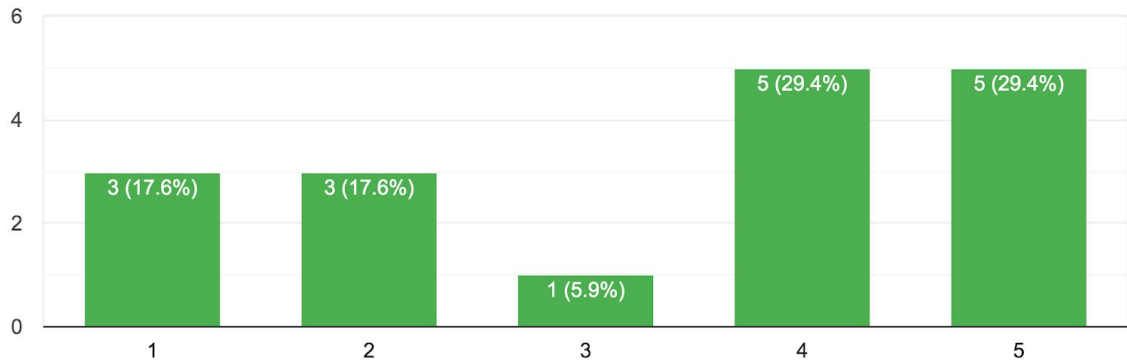
## How skill or chance based do you think the game is?(According to matrix above)

17 responses



## In which way does the game challenge the player?(According to matrix above)

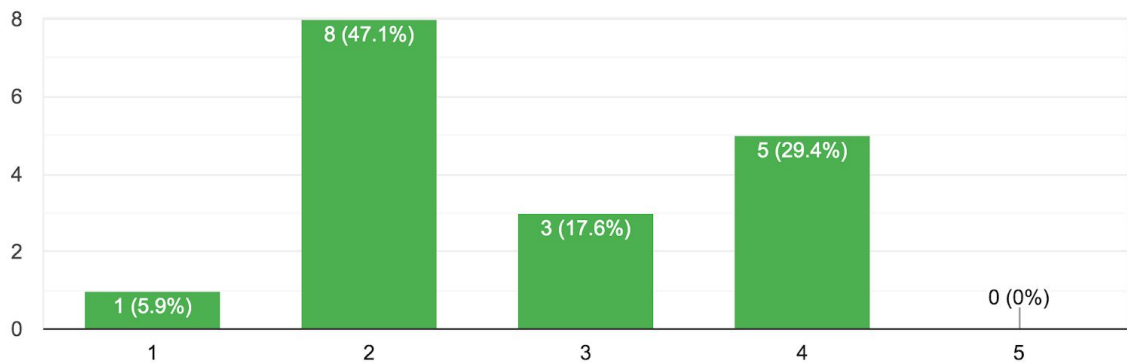
17 responses



The majority of the players have reported that the game is more skill-based than luck-based. This is an expected outcome. However, answers to the second question might seem surprising: the players were split while identifying the main challenge of the game as a mentally challenging (Chess = 1 on the horizontal scale) or physically challenging (5 on the horizontal scale). This shows that players took different approaches in going through the game: some players tried to put the bots on “autopilot” in order to complete the game, while the others were controlling mostly the only one bot like they would in a shooter game. While we would like the game to remain high-paced and requiring the player to make many action in a short period of time, we can see that the players should be incentivized more towards a strategic thinking.

## Rate how good the controls felt.

17 responses

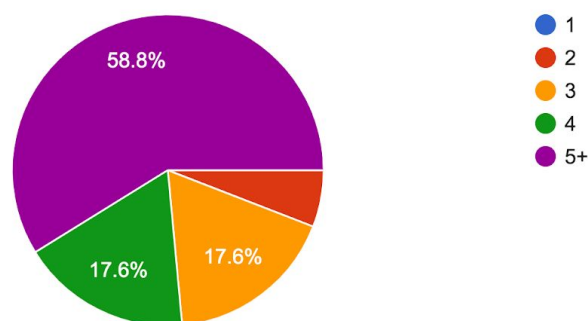


Most of the players had an issue with the controls of the bot. To some it felt stiff, some had an issue with aiming and running. Again, we went into alpha-testing phase knowing that the controls have some issues to them but didn't have enough time to implement them in a nicer and more responsive way, so the complaints from the testers are expected and were welcomed.

### 5.3.3 A.I.M. Bots

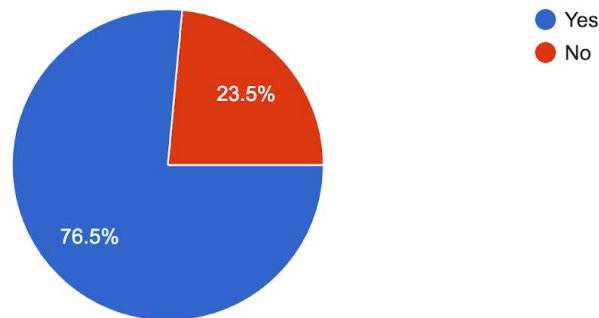
#### How many different A.I.M. behaviors did you see?

17 responses



## Were you able to understand the factors for some of the behavior transitions?

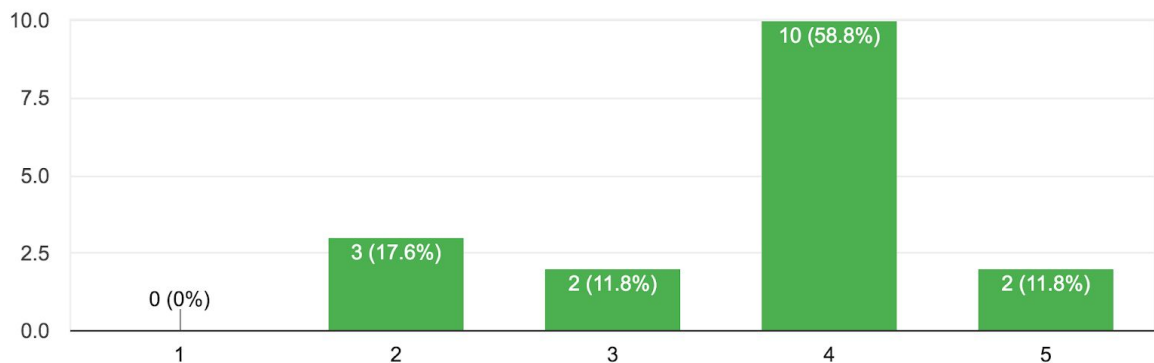
17 responses



Most of the players were able to notice all the bots' behaviour modes. Also even more people were able to pick up on the most important game mechanics: the behaviour changes. We believe that improved tutorial would help to increase the numbers.

## Rate the Almost Intelligent AI of the bots.

17 responses

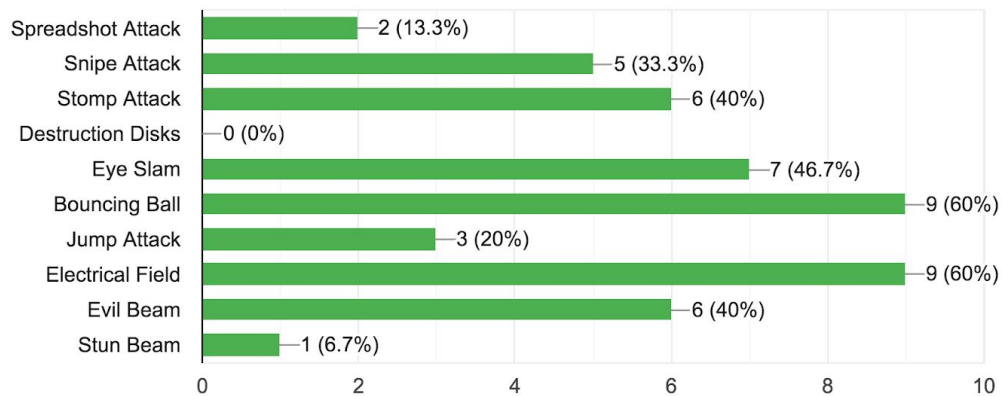


The majority of the players was satisfied with the AI of the allied bots. Some players suggested to add more behaviours to make the game more interesting, and one player marked that the "coward" bots get stuck in the walls sometimes. Two players suggested that the bots should run away from the bomb, however we designed it specifically in this way and believe that this is a nice gameplay element.

### 5.3.4 The Boss

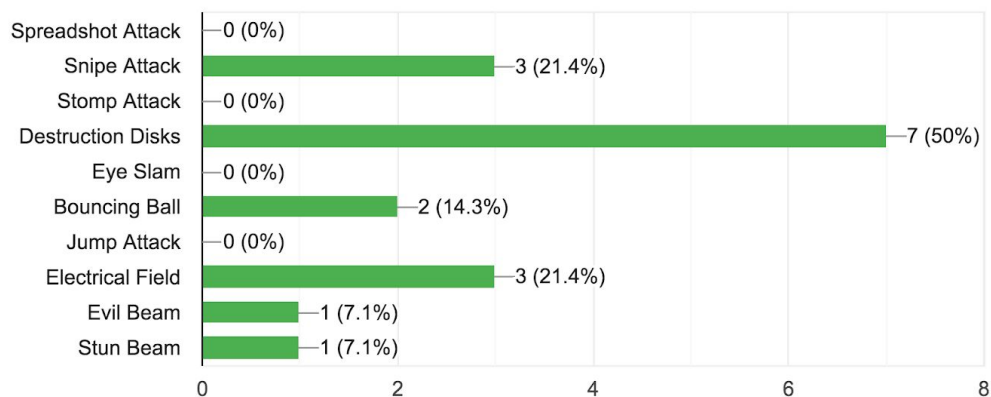
#### Mark the attacks you liked.

15 responses



#### Mark the attacks you disliked.

14 responses

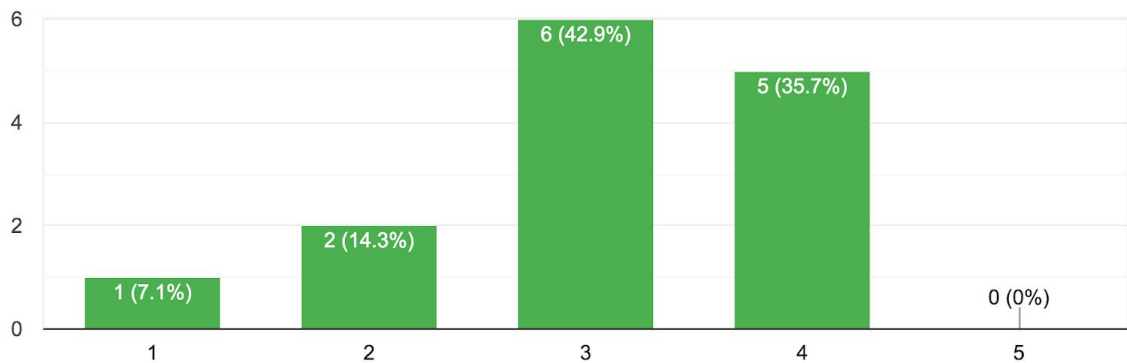


While the distribution of the liked attacks is somewhat even, the most disliked attack was the destruction disks attack. The interesting thing is that it was disliked for different reasons by different players: the players that used mostly ranged weapons thought that the attack is boring and useless, while the more melee-focused players thought that the attack is unfair and is impossible to dodge.

Also the players thought that the snipe attack is too easy to dodge and that the electric field attack can be overwhelming to the player, especially if the bots are low on hit points and are prone to panicking.

## Rate how much you liked the LALA attack.

14 responses



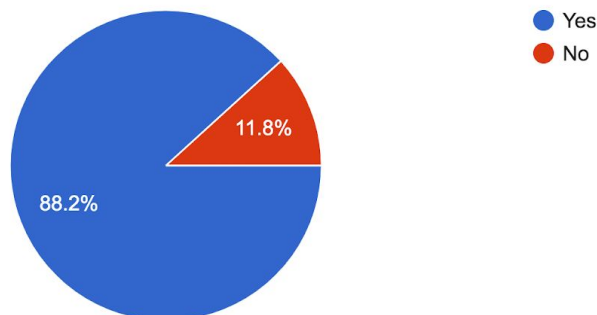
The LALA attack was received more positive than negative, however most of the players didn't get what the effects of the LALA attack are. The general feeling of the players was that the blindness lasted for too long. Also players complained about dropping frames during the LALA attack, which means that some optimisation work is due.

Most of the players suggested to telegraph the incoming attack more and to have more attacks that require micromanagement from the player.

### 5.3.5 Items

## Did you dig up any items during the bossfight?

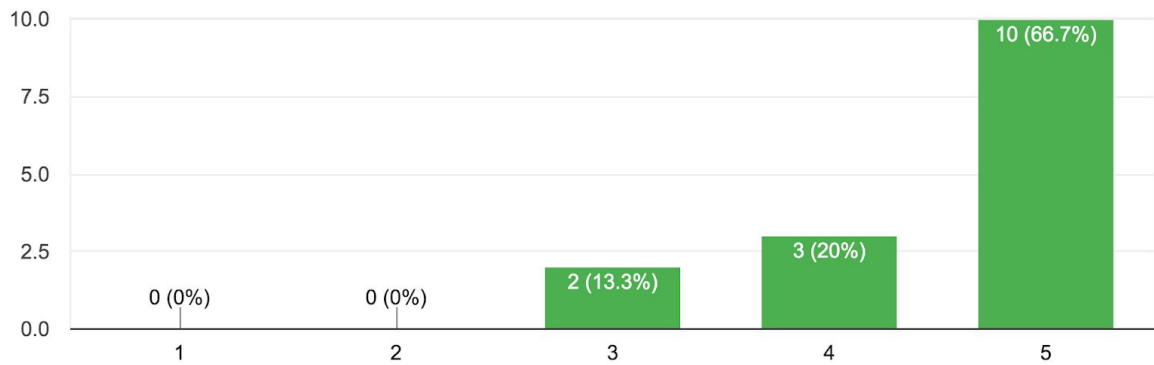
17 responses





### Rate how much you liked the digging feature.

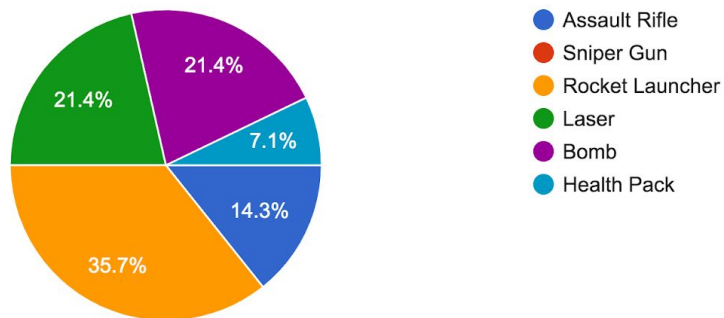
15 responses



Most of the players used the mechanic of digging up useful items during the fight and the majority liked it.

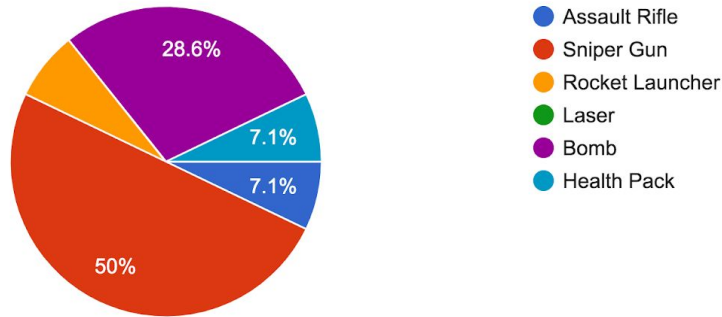
### Which item did you like the most?

14 responses



## Which item did you like the least?

14 responses



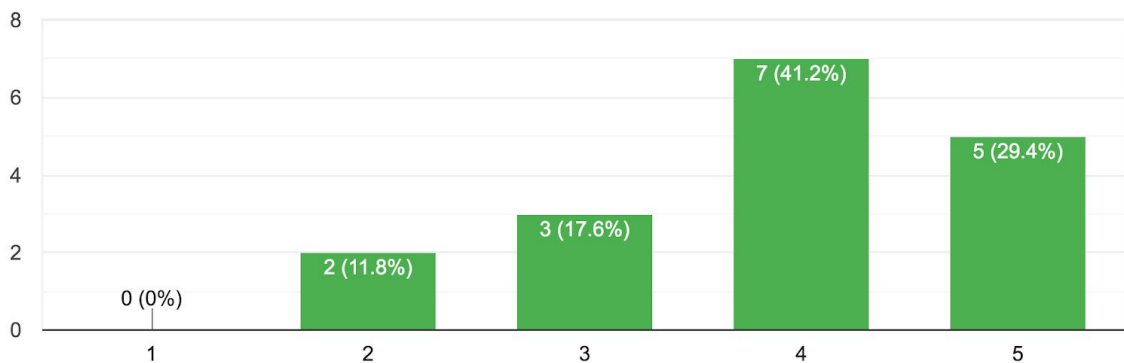
The most favored item was the rocket launcher, because “it was fun” and “DAMAGE”. Also quite a lot of players enjoyed the ticking bomb idea.

On the other hand, the most disappointing weapon was found to be the sniper gun. People described it as boring and not dynamic enough. Also some players expected health packs to heal other bots on usage. At the moment the healthpack health the holder of the healthpack, however it can be an interesting gameplay element if the healthpack can be used only to heal other bots.

### 5.3.6 Overall experience

#### Rate your overall game experience.

17 responses



Overall, majority of the players have found the game good and fun to play. People highlighted the visuals, AI mechanics, fast and chaotic gameplay as the best aspects of the game. Some people thought that voiceovers for sound effects were funny,

some people didn't enjoy the idea that much. The most criticised part of the game were unresponsive controls and broken aiming.

## **5.4 Issues and planned changes**

While the overall feedback was rather positive, there are a lot of issues to be resolved within the next two weeks. During testing we discovered quite a few bugs and the feedback made us rethink a lot of mechanics.

### **5.4.1 Balancing issues**

The testing showed us, that people find certain attacks/weapons/behaviours etc. too hard/annoying/not well enough explained, that we didn't expect. The following bullets were especially present.

#### **1. Hard Mode**

While our game, because of the sheer amount of different things the player needs to look out for, was quite difficult in the beginning for most players, it turned out to be rather easy for most after the second or third try.

Also, it turned out that the survivability of a bot with use of health packs is quite high.

In order to make the game more difficult, also when players are already familiar with the mechanics, we plan to add a 'hard mode' where the boss has more health and the bots take more damage.

#### **2. Sounds**

Quite some people got confused over the amount of different sounds playing at the same time, so some tuning in volume to further stress the important one's may be necessary here.

#### **3. L.A.L.A.**

Most people, when interviewed, did not know what the L.A.L.A is and how it worked, so we may need to add tutorial scene vii, that we originally scrapped, back in. Also the extra damage, that can be dealt to the eye-weakpoint is at this point not worth the risk of getting into the L.A.L.A.s danger zone. We need to increase the damage multiplier here.

## **4. Electric Field**

While the electric field was the second most liked attack, there is a big problem with it, that several testers ran into. Once it gets rather big (so big, that it reaches one or more of the towers that need to be hit simultaneously to destroy it), bots stationed by the player at a tower within the field tend to turn cowardly, since they take damage all the time.

That leads to the electric field being nearly indestructible once it swallowed all the towers.

### **5.4.2 Tutorial Changes**

As there is quite a lot going on in the scene and many testers told us, they had a hard time focussing on the boss attacks while needing to control four bots, we decided, that we need to explain more to the player. All in all the tutorial needs to be updated quite a bit still.

#### **1. Buttons for exposing weak points**

Many people did not understand, that the exposing of weak-points via the buttons in the scene only works during the second phase, when the protection of the boss is actually up. We need to make that clearer in the corresponding tutorial.

#### **2. Boss attacks**

As mentioned above a lot of people were still confused about the boss attacks, that weren't explained in the tutorial. We may add a scene for every attack or incorporate several attacks into one scene in some clever way.

#### **3. Walls of text**

The tutorial was intended to teach the player game mechanics without too much reading in a fun and interactive way. While that did work out nicely to some extent, there still ended up being quite some text in some of the scenes. During testing, a lot of players ended up not reading the texts carefully and therefore not understanding the supposed lectures one hundred percent.

We plan on updating the existing scenes and shape the new ones in a way, that needs even less text than we have now and instead incorporates the mechanics we want to teach in a way that requires to understand them in order to solve the scene.

### 5.4.3 Bugs and incomplete features

Obviously, we discovered quite a few bugs during testing, that we intend to fix. For example the 'Exit to main menu'-button in the tutorial only works in scene one.

Also we plan on adding a model for the sawblades and destroy the buttons, that reveal the weak-points, when they no longer work (during the 3rd phase, that is), since this, as stated in 5.4.2.1, caused quite some confusion.

### 5.4.4 Controls

A lot of people also complained about the controls being unintuitive.

#### **1. Selection**

Bot selection has a few issues:

- The way, the selection changes, when using WASD is kind of weird at some points
- There is no way to tell, that a bot is uncontrollable, other than by the sound it makes, when trying to take control of it, as well as the fact, that it just doesn't work

We intend to improve in both points, by giving a visual clue, that a bot is currently not selectable and making the controls in selection mode more intuitive.

#### **2. Movement and A.I.M.ing**

Because of the different speeds of the animations of moving forward, diagonally, sideways etc., the movement speeds also differ, depending on which way the bot is facing vs. where it is moving. We intend to fix this, as well as make it clearer, where a bot is currently aiming its weapon.

## 6. Conclusion

### 6.1 The Final Product

Our game is a boss fight against a powerful lamp-shaped robot, the 'Lethally Armed Management Program', shortly L.A.M.P. The player has four vessels at their disposal to put up against the L.A.M.P., the 'Almost Intelligent Machines' (A.I.M.s), also referred to as bots.

The catch here is that one cannot simply order the bots to do something particular, but has to use their in-built mood-routines, that switch mood, depending on their surroundings (they get angry if they attack, afraid, if they take damage etc.), to one's advantage. The player can control only one bot at a time and if they want the bot to do something special, they need to set it up to do so by making it adapt a certain mood and positioning it accordingly.

There are several items to be found to aid the bots in their quest to defeat the L.A.M.P., as well as a tutorial for new players to be able to learn the core mechanics quickly.







### Most significant changes since Alpha Release

The most noteworthy addition to A.I.M. after the alpha release four weeks ago is arguably the tutorial with its seven levels introducing new players to the core mechanics they need to know before being able to play.

While it has been discussed extensively in the last two chapters, there was an extra scene added as well as one scene changed in response to the playtesting:

- Scene II no longer has the player kill a bot, instead they need to dig up a bomb from the single digging spot in the room and throw it over a wall to damage the boss.

Therefore, it is now necessary to learn the digging and throwing mechanics in order to complete the level and the player is no longer wrongly hinted at being supposed to kill corrupt bots.

- There was an extra scene added between the original scenes III and IV, that has the player bring four different bots into four different moods. This makes the player understand the moods better, which was not yet properly explained and not understood by quite a few testers (almost a quarter claimed, they didn't understand any of the transitions)

Another important change was the addition of a hard mode, that was proposed in chapter 5.4.1.1. In order to be fun for a larger variety of players for a longer time, we added this mode, where everything is more difficult: the boss has more protection and deals more damage, the health packs give less HP back etc.

During the open alpha testing the mostly criticized facets of the game were the “stiff” controls, bad aiming and unintuitive jumps during the bot selection. All of these issues were properly addressed: now the bot running speed is independent of the



direction it's moving in, the bots aim at the mouse cursor and the selection logic was changed to make the transition between the bots feel more natural to the players.

Further changes are mainly tweaks, rebalancing and bug fixes, some conducted before, some after the tests.

## 6.2 Experience

Overall, we are happy with the final state of the game. We managed to implement the initial idea of providing an epic but fun boss fight experience with original game mechanics. With 2 members of the team having experience of participating in this course before we narrowed our vision of the game down to its core elements and kept the development inside the scope of the original idea. This allowed us to have a complete production-ready game in the extremely tight time constraints. Some of the optimistically set "High target" goals were not achieved (such as extra boss or local multiplayer), but this sacrifice gave us enough time to polish what we have.

The development schedule that was created in the first stage was used more as a guideline and not as an actual day-by-day schedule. It helped us to assign milestones to the actual points in time so that we were able to see how good (or bad) we're doing on a global scale. Instead trying to stick tight to a schedule, we chose more agile approach with weekly (more frequent at the times of very intense development) calls, during which we were deciding what features should be implemented in the upcoming days, and then distributing tasks based on our personal preferences and areas of expertise. Our process resembled scrum methodology in some sense, however, it seems like it's virtually impossible to follow a "proper" scrum in a student project, because the number of hours people could spend on the project varied from week to week. This deliberate flexibility facilitated our development greatly, as we were able to focus on the task with the highest priority and swap the tasks between the layers when we realised that it would make more sense.

The prototype stage helped us to put our finger on the core gameplay idea, to figure out what makes our game unique and interesting, its strengths and weaknesses. It is hard to estimate how much it has helped us, and whether or not we could've reached the same conclusion without investing our time into making the "real world" prototype. However, it might very well be that we could've spent time more efficiently without building the prototype and focusing on eliciting the design decisions instead because most of the time during the prototyping stage was spent thinking about how to make it work with real-world objects and establishing rules for movement and attacks and the results were not transferable to the final product.

The playtesting stage went very smooth and was a good indication that we've done a good job of keeping the coherent vision of the game during the implementation. The most criticized element were the clunky controls, which were fixed shortly thereafter, but the game itself was interesting for the players and most of them said in the private discussions that they'd like to play a game like this more with different bosses and maybe other gameplay elements.

From an organisational point of view inside the team, everything went smooth. We didn't have breaking interpersonal conflicts from the very beginning of the project to the end. The personal interests of the team members didn't overlap, the tasks distribution always happened without arguments, and everyone was content with the work they had to do.

## 6.3 Course personal impression

### **Q: What was the biggest technical difficulty during the project?**

**Liou:** For me, the biggest technical difficulty was implementing the various algorithms which the L.A.M.P. boss uses to gather information on their surroundings. The different attacks need distinct information of the bots and use it to determine a logical probability of the attack. Defining the correct search areas and combining different turned out to be a real challenge.

**Nikita:** For me the biggest technical difficulty was the AI implementation for the boss. It was the first time I programmed a game AI, so I had to go through a bit of a relevant literature and internet resources in order to choose a fitting approach. In the end the AI was implemented in a classical state-machine approach, with different machines for each of the bot's behaviours. The bots AI also needed to be able to walk around the level without being stuck on the walls, being able to aim properly with the guns, etc.

**Jan:** Getting the tutorials to behave as I wanted them to behave. I usually had quite clear ideas, what I wanted the tutorials to be like, but since we didn't plan to have an extensive tutorial, as we do now, from the beginning, a lot of the code, that was already there, wasn't designed to allow for behaviour, that didn't conform with the main game. For example seemingly simple things like getting a bot into a fixed mood and keep it there or have the boss perform a fixed attack once and only once, then idle, was harder than anticipated, since bot moods change per default and the boss wasn't programmed to ever stop attacking.

**Konstantin:** For me it definitely was the animation system. Although the animations for the bots were taken from a mocap library, it was a lot of work to set up the animator with all its transitions and the code backend for linking the animations to

player input and events in the game. This also applies to the Boss, where additionally the animations had to be created.

**Q: What was your impression of working with the theme?**

**Liou:** The theme was very interesting as artificial intelligence is on the mind of any gamer or programmer. The theme is not restrictive at all because you can implement it in any type of game genre, which also shows in the different types of games the teams presented. Of course, it leads to the common aspect of having artificial intelligence in the game but this does not make the game similar.

**Nikita:** Having a theme helped our new team to start a conversation about the game ideas. Without a common theme from the beginning, it would've been harder to decide on the game idea to start actual work.

**Jan:** The theme was not very restrictive and offered for a lot of different interpretations so I thought it was nice to work with.

**Konstantin:** Considering that most of my previous games either didn't have complex AI or got around the problem by having human competitors, I found it really interesting to work with the theme [Artificial Intelligence]. I also liked that it left much room for interpretation, which shows in the quite different games every team came up with in the end.

**Q: Do you think the theme enhanced your game, or would you have been happier with total freedom?**

**Liou:** The theme led to many different ideas during the brainstorming phase, that's why I think it helped us a lot in defining our core game mechanics. As mentioned before, the theme was not restricting which meant we had a lot of freedom while being led into a certain direction.

**Nikita:** The game idea was completely inspired by the offered theme (it's even in the name of the game!), so there is no doubt that it's better to work with the theme.

**Jan:** I usually find it easier to be creative, if I'm not restricted by a theme. When trying to come up with own ideas in the beginning, I often realized, that I just had some game concept I liked and then mapped it to the theme somehow.

**Konstantin:** I think the theme actually enhanced our game. It laid the focus on having a convincing bot AI, that intentionally acts "not so smart" at times if left alone, but that can be useful to the player if they interfere and take care of their bots.

**Q: What would you do differently in your next game project?**

**Liou:** There is not really anything I would change in terms of project management. We had weekly meetings and clear goals and tasks, which we defined with an issue tracker, for each week. Also, we were always up-to-date on the progress of the other members, because we reviewed the code before we merged. With this, we could detect game breaking bugs early and correct them before the game is merged.

**Nikita:** The project went very smooth from start to finish, I will use this experience as a standard for group projects I take part in. The only thing I would've done differently is not using the Github LFS solution, as it "blackmails" you to subscribe after expanding 1GB free storage limit, so we had to spend 10\$ on the LFS subscription :)

**Jan:** Put way, way more effort into the beginning phases (planning and such). Especially in the beginning, but partly also after starting to program, I often felt like lacking a clear idea of how the game was supposed to look, play, feel like in the end, what exactly it is, that I want to implement programmatically.

**Konstantin:** When making the trailer I discovered an awesome Unity plugin for camera management (Cinemachine). I would use this earlier next time and integrate it in the game, making custom camera scripts obsolete.

**Q: What was your greatest success during the project?**

**Liou:** The greatest success for me was when the first version of the boss was fully functional. It would intelligently target bots and correctly perform its attacks. This was at the same moment the basic bot A.I. was functional, so this was the moment where we could finally play the game.

**Nikita:** Making Unity's NavMeshAgent work without it being able to drive the bot's movement directly was definitely a challenge and when I finally made it work was definitely a big success-feeling. Also the development process that we established at the early stages proved to be very successful.

**Jan:** The Tutorial.

**Konstantin:** That all models & textures in the game self-made and original.

**Q: Are you happy with the final result of your project?**

**Liou:** I am really happy with how the game turned out. It is strategical in a way that the player has to think which actions benefit him the most, but also action-packed because a lot of things are happening and they have to be fast. The models and effects look gorgeous and the sound effects and voice lines underline the humorous aspect of the game.

**Nikita:** Yes.

**Jan:** Somewhat. I think it turned out quite nicely in the end.

**Konstantin:** Yes, I am! Thumbs up to my teammates, I think we did a great job together in the given amount of time.

**Q: Do you consider the project a success?**

**Liou:** Although there are still some little issues we could not fix due to time constraints, all in all I would consider our game a huge success. We were able to realize the vision we had at the design phase and most of the playtesters really liked the main mechanic of the game.

**Nikita:** Yes, big time. With the time pressure we had, we had to push ourselves and spend a lot of time working on the project during the semester, and the result feels very satisfying.

**Jan:** Not necessarily a success, but I think I learned a lot during the course and am quite content with it in the end. Though, in future projects, I hope I'll be able to organize myself better.

**Konstantin:** Yes, not only am I happy with the final result, but also the way we got there.

**Q: To what extent did you meet your project plan and milestones (not at all, partly, mostly, always)?**

**Liou:** We mostly met the milestones during the project phase. We certainly had to reorganize and change some goals, for example, we discarded the command ability as we realized during development that it did not fit with the game flow. All in all, the general plan for the game stayed the same and we were able to reach our main goals.

**Nikita:** Except for small adjustments of the milestone goals we did during the development of the game, we mostly kept with the initial plan for the game.

**Jan:** Mostly.

**Konstantin:** I can't remember a time where we really got behind our plan. We met regularly to discuss our progress and what task each of us tackles next. To keep track of everything we assigned Github issues for every task to one team member, that should be resolved until the next meeting. A feature branch/pull-request

workflow with code reviews kept everyone up to date about the current state of the game. These things helped quite well to organize ourselves.

**Q: What improvements would you suggest for the course organization?**

**Liou:** The course organization was fine with the milestone presentations and brief introductions to the next milestone. Unfortunately, I think the development time was too short for the scope of the given project. Here I would suggest either let the course cover 2 semesters or put part of the course into the semester breaks, either earlier start or later final deadline.

**Nikita:** It would be nice to have more time after the playtesting stage to be able to go through 2 iterations and see how the players feedback changes. Besides, I'm not entirely convinced by the necessity of the real-world prototype assignment: with the time issue there is, it feels like an impediment. I think it doesn't make sense to ask students to prepare a schedule with tasks week by week by person for 3 months in advance. It would never work, especially in students projects. As for our situation, we had this assignment before we even went into the "Prototype" stage, and we didn't have the gameplay completely figured out yet. This forces us to stick to our original idea during the prototyping stage more, making us less open to changes (which defeats the whole purpose of prototyping). Considering that we also wanted to try things (technology-wise) we didn't use before, our time estimates could not have been precise at all. I believe it should be enough to ask the students to determine layers and milestones with the areas of the application each team member would like to be responsible for, but the schedule tables with tasks assigned to people week-by-week is unreasonable to ask for. Also, application architecture and application design were not touched in the course. I think it would be better if for the development schedule it was enough to set up milestones with describing functional minimum, desirable, etc. requirements and drop the real-world prototype task, and instead focus more on the game design and application design. It would've been interesting to see how other teams designed their game from the software engineering perspective, not only game design decisions.

**Konstantin:** THIS!



**Jan:** The schedule is a bit tight, but can be worked with, I think. It would maybe be cool (since you can do the practical course twice, anyways), if it were possible for a team to do a two semester version of the practical. I know, it is possible to extend the game you made the first time you did the practical during the second time, but it might be cool to be able to plan the project for two semesters from the beginning on.

Having 4 - 6 weeks instead of 2 - 3 between presentations etc. However, I'd probably not do this myself.