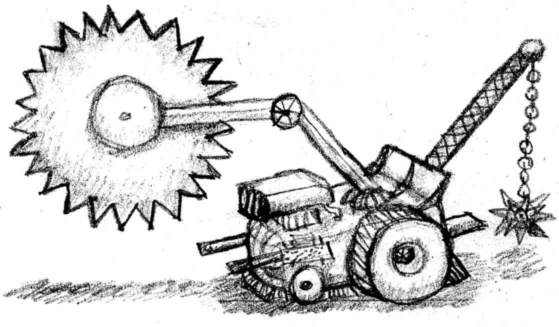
INCREDIBLE BATTLE MACHINES

Game Programming Laboratory

Formal Game Proposal

Huw Bowles
Samuel Muff
Filip Wieladek

Revision: 1



Contents

1 Introduction

This document describes the game *Incredible Battle Machines*, which will be developed throughout the Game Programming Laboratory course. It explains the main game idea along with the proposed time schedule. The document also contains an assessment of the game idea. Finally, this document is only a detailed starting point for the game. Any feature described here in this document can change at any time and new features can also be added during the development cycle.

The rest of the document is structured in the following way. Section 2 describes the general game idea and graphical style of the game. The following section, section 3 handles the development plan and the detailed list of tasks. Lastly, the section 4 evaluates the game idea.

2 Game Description

This section describes the general idea behind the game. The main focus of the game is controlling a robot (also referred as Machine) in order to defeat other players who are also controlling a robot. The Game is made out of 3 phases, Machine Construction, Logic Construction and Battle which are described in the following subsections 2.1, 2.2, 2.3 respectively. Section 2.4 touches on the targeted visual style of the game

2.1 Machine Construction

This is the very first phase of the game. It is optional to players who have already gone through this phase, and may skip it with a machine they have already built. That means, that one can build the robot anytime and then use it in a match without having to create a new machine. As the name suggests, this phase of the game requires the player to build their own machine (robot).

The player will start from scratch and should be able to build their own robot. Every player is given a budget. Players are forced to design their robots within this budget. Players start from building the basic hull of the robot. During this stage players are able to choose from different materials and thickness they want the robots to be built from. These materials include:

- Steel
- Aluminum
- Wood
- Carbon Fiber
- Foam

The materials the player picks will have different effects during the battle phase (see section 2.3). For Example, Carbon Fiber offers a very light and reasonably sturdy material, which however is very expensive. Steel, on the other hand offers the same physical properties, but weighs a lot more.

The player starts the next phase of the construction once he finalizes the design. This requires the player to place additional equipment on the robots. The equipment includes:

- Batteries
- Weapons
- Motors
- Additional optional equipment added during the development of the game

The player can mount any of these onto the robot. It is important to note, that every additional equipment requires energy to function correctly. Thus, the player needs to choose the correct battery for their add-on otherwise the robot will not be able to function correctly. Figure 1 shows a basic sketch of the machine construction once the shape is done. It also displays a mockup of the user interface.

It was decided that there robots will not have wheels and that all robots will be floating in the air. This makes the control structure uniform over all the robots and makes the construction of the robots a lot easier without loosing too much of flexibility. Finally, it was also decided to skip the shape construction at all and provide players with already present shapes. This might be changed in the future depending on the user feedback during the development cycle. The appendix contains sample sketches of 3 basic hulls which will be provided. Note, these shapes are not the only ones which will be available, but provide a starting point. Section 2.2 describes the next phase.

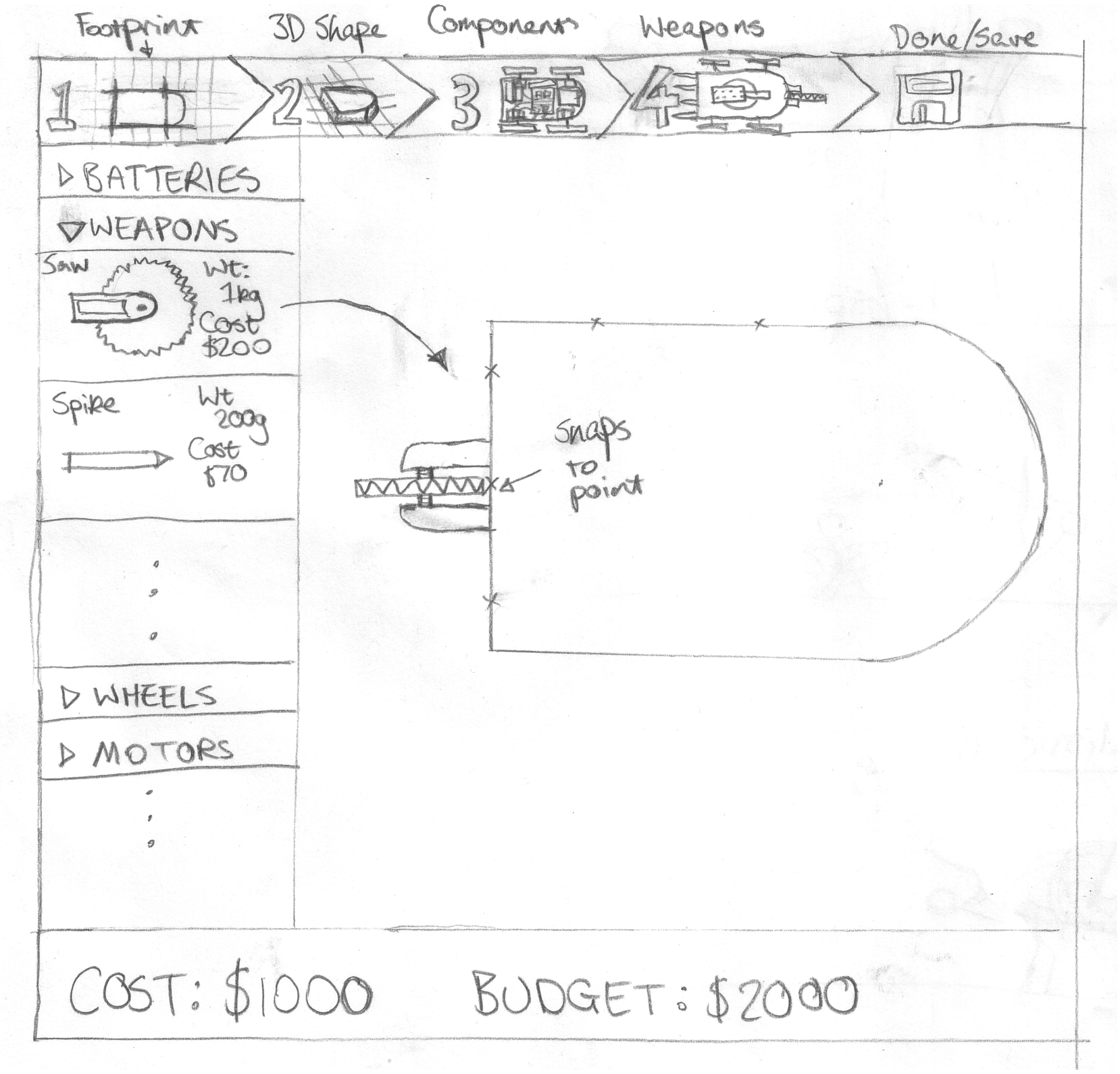


Figure 1: A basic sketch of the machine construction

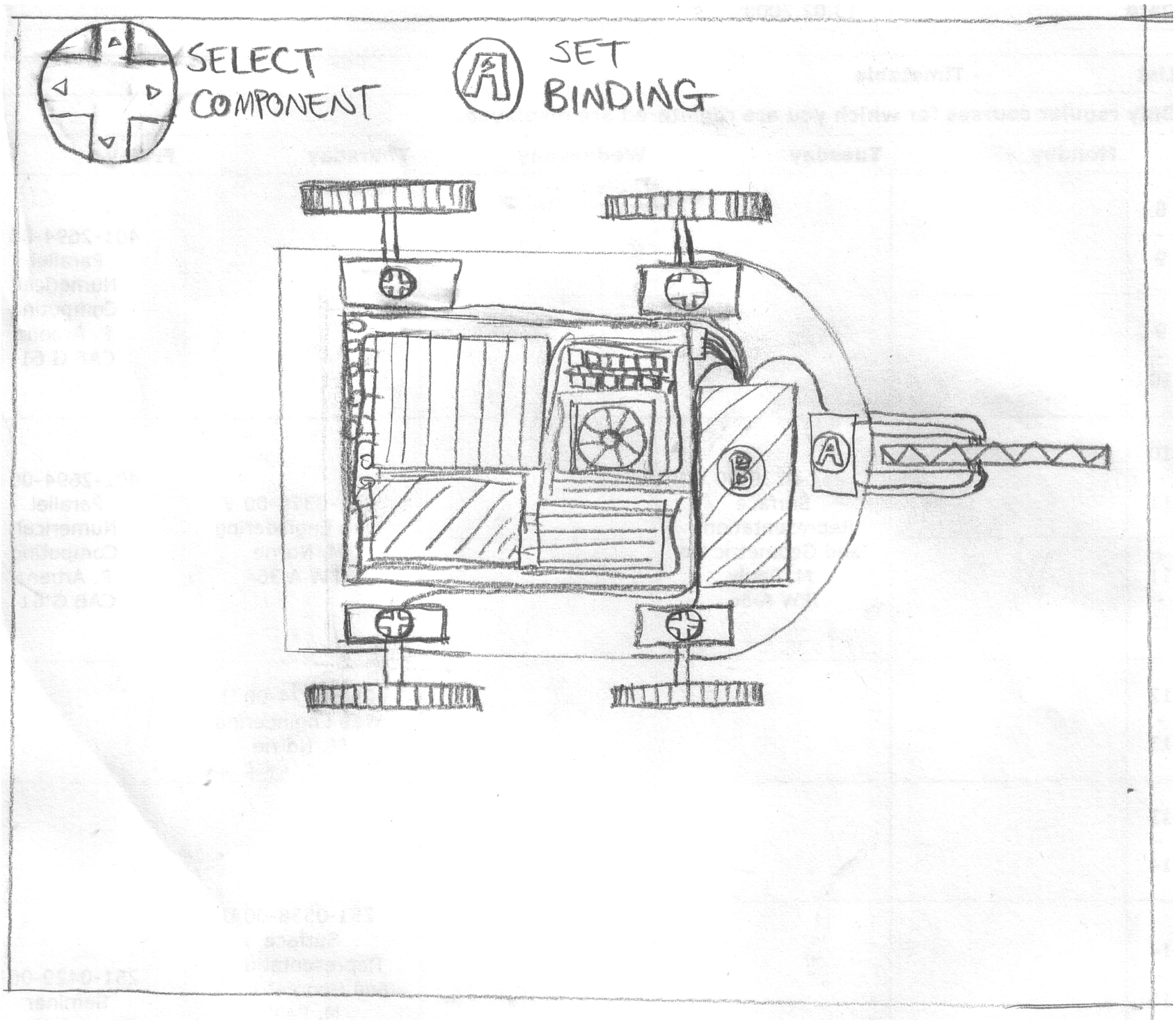


Figure 2: A basic sketch of the logic construction

2.2 Logic Construction

This section describes the phase after the Machine Construction phase (see Section 2.1). In this phase the user defines the robot logic. As every robot is unique, it is impossible to define a common control scheme for all the robots. Thus it was decided that the user will actually define its own control scheme. The general idea is that the player will create a logic circuit for its robot and use the xbox controller as a remote control.

The logic circuit will be an abstraction of a real circuit, as most players will probably not be familiar with electrical engineering. The general idea is that the players place event sources and connects them to the components. Moreover, the player is given a choice to place additional simplified logic blocks to add some additional functionality. The most important parts are the events sources which will be made of the controllers button.Figure 2.2 shows a basic sketch of the logic construction phase.

2.3 Battle

This phase of the game is the most important phase. Right now there is only one battle mode planned which will be regular deathmatch. In this mode, all the players (up to 4) will use their constructed robots to battle each other. This will be done in a last man standing mode, where a round is won when only one player is left in the map.

The game starts with all players spawning in a corner of the map. The robots use their weapons to do damage to other machines. The machines take damage in two forms. One form, is the chassis, which will gradually take damage until it reaches a critical state and the robot is taken out round. The other form is component damage. The outer components attached to the hull will take gradually damage until they fail and are unusable. The robot can continue until the chassis is not critically damaged. Thus a player may still win although the driving unit (motors) have been taken out.

Every match consists of a variable number of rounds (depending on the settings). For each won round, a player is awarded a number of points. Thus at the end of a match, the player with the most number of points is the winner of the match. Figure 3 shows a basic sketch of the battle. Note, the camera position will be a bit skewed so that it will not be a direct top down view.

2.4 Graphical Style

The game aims at being more cartoon-like and exaggerated. Thus it is most likely that the game will use a cel shaded renderer. Moreover, the physics will be aimed to be exaggerated so that the game has a less serious “feel”.

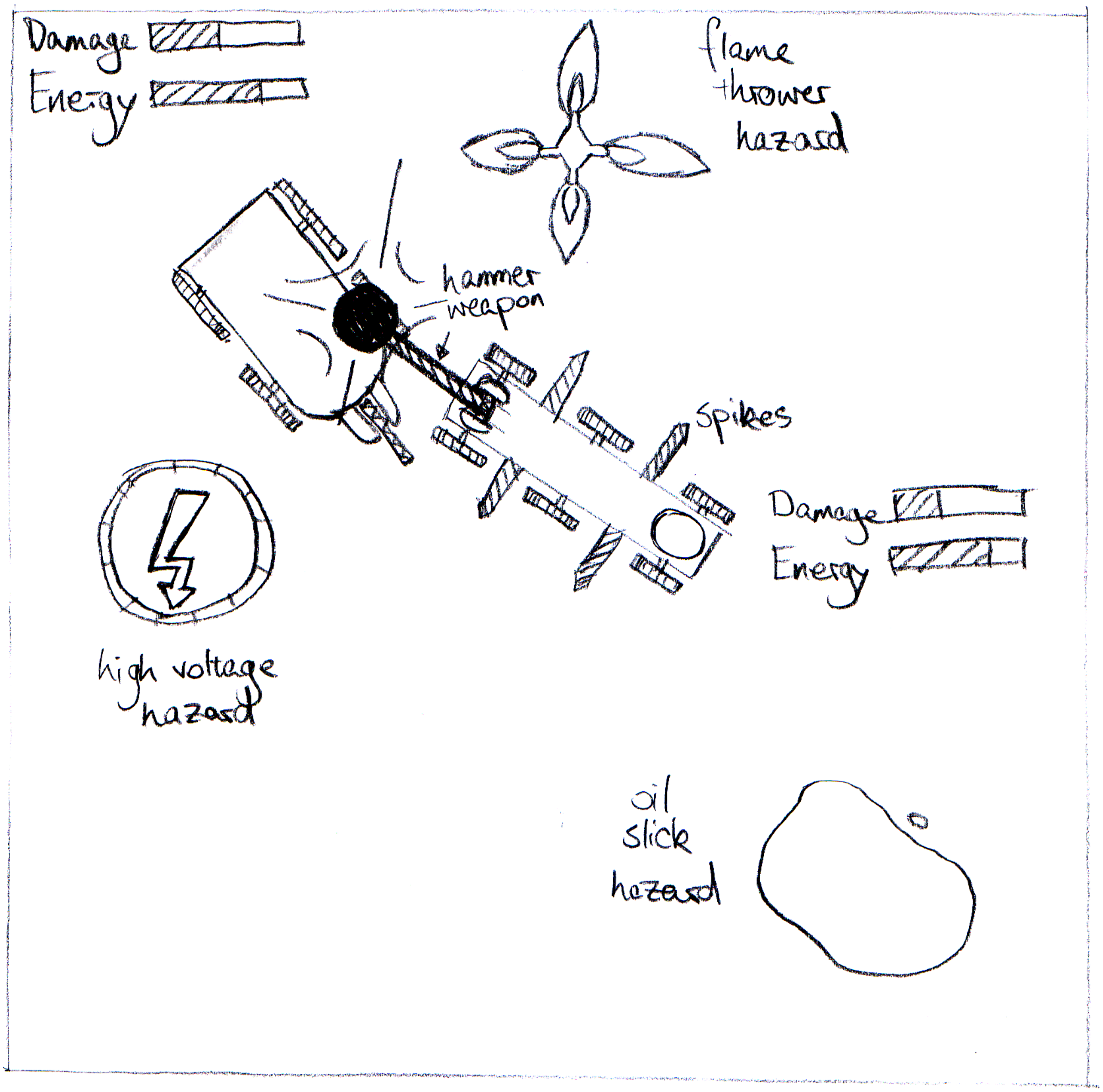


Figure 3: A basic sketch of a Battle

3 Development Schedule

This section describes the development schedule for the project.

3.1 Development targets

What follows is a list of targets for the development schedule.

3.1.1 Functional minimum

- No editor, predefined robot(s)
- Basic physics / basic calculation of weight / accel / max speed
- Infinite plane arena

3.1.2 Low target

- Robot structure/exterior editor
- Robot control logic editor
- 1 map simple arena
- 1 or 2 weapon types
- Deathmatch (scoring)
- Local multiple player support
- Health bar style damage

3.1.3 Desired target

- Robot structure/exterior editor
- Robot control logic editor
- 1 map, with traps, obstacles etc
- Multiple plug on weapons
- Advanced physics - simulation of components with weights etc
- Robot structural damage / component breakdown
- Choice of batteries - balance weight / energy ratio
- Team battles

3.1.4 High target

- Stylized rendering
- More maps (2 or 3)
- Painting robots / sprite tattoos
- Robots have ability to fly
- Team battles
- Different game types (soccer, etc)

3.1.5 Extra

- Online multiplayer, tournaments
- Map editor
- Points Scoreboard / Career play

3.2 Development Tasks and Estimated Schedule

This section provides a detailed summary of the task needed to accomplish the targets defined in subsection 3.1. The following subsections include these specific tasks as well as the assigned team members. The estimated length of the tasks will be measured in days and the following symbols, "Huw", "Sam", "Fil" will be used to indicate the team members Huw Bowles, Samuel Muff and Filip Wieladek respectively.

3.2.1 Functional minimum

The following table displays the tasks and an estimation on how long they will take to be completed.

#	**Task**	**Length**	**Member**	**Description**
1	**Defining Rules**	2 days	Sam	This task involves refining the rules of the game
2	**Weapons**	2 days	Huw	This task involves defining the weapons which will be in the game
3	**Robot**	3 days	Fil	This task involves modelling the first robot
4	**Basic Physics**	7 days	Huw	This task involves implementing / linking the basic physics engine into the code
5	**Robot Engine**	7 days	Sam	This task involves implementing the basic robot engine, so that attachments can be attached
6	**Map**	4 days	Fil	This task involves the creation of a basic map implementation and the corresponding example map

3.2.2 Low target

#	Task	Length	Member	Description
7	**Logic**	4 days	Fil	Implementing the general framework for the robot logic. This includes adding basic components (all buttons, + triggers + sticks) of the controller and also for one component (motor)
8	**Logic Editor**	8 days	Fil	Creating the graphical user interface for the logic framework so that logic for robots can be created by users
9	**Structure**	5 days	Huw	Implementing a framework for creating custom robots shapes, including all the required classes . . .
10	**Structure Editor**	10 days	Huw	Implementing a Graphical user interface for the structural framework.
11	**Basic components**	4 days	Sam	Implementing the basic components according to the framework specified
12	**Basic Modelling**	4 days	Sam	Modelling the basic components which are to be implemented. This includes, a motor, a wheel and 1 weapon
13	**Game mode**	3 days	Fil	Implementing a the first game mode "DeathMatch" where player score according to the rules
14	**Multiplayer**	2 days	Huw	Implement the support for local multiplayer. This includes players building robots one after another and then fighting on one map
15	**Game Physics**	4 days	Sam	Implementing the game physics so that robots take damage
16	**Simple Map**	4 days	Fil	Making a simple map for the players to play in

3.2.3 Desired target

#	Task	Length	Member	Description
17	**Map Objects**	4 days	Sam	Implementing the general framework for map objects, such as traps and power ups
18	**Advanced Physics**	15 days	Huw	Implementing/linking a better physics engine so that the robots can be simulated correctly
19	**Component breakdown**	5 days	Fil	extending the existing robot damage to component damage based on locations of component
20	**Team Battles**	3 days	Fil	Extending the exisiting deathmatch gamemode to include team deathmatch
21	**Map 2**	5 days	Sam	Creating another map
22	**More components**	4 days	Huw	Adding new components to the game (to be decided which)

3.2.4 High Target

#	Task	Length	Member	Description
23	**Maps 3**	5 days	Huw	Creating another map
24	**Painting**	3 days	Sam	implementing robot paint, which allows the user to paint robots
25	**Fly...**	4 days	Fil	Adding the possibility of flying robots (adding a motor and a propeller)
26	**Soccer**	5 days	Sam	Adding a soccer game mode to the game, where robots need to push a ball into a target

3.2.5 Extra

This section will be omitted as it is highly unlikely that the team will reach this phase of the development during the course.

4 Assessment

This section describes the evaluation of the game.

4.1 Strong points

The game strongest points will be in the custom content design. Users will have the ability to create their own robots and fight each other. This will be challenging as depending on the enemy, a totally different approach may be necessary.

One of the most innovative parts of the game is creating the robot logic. The players are no more constrained to what the developer of a game supports in term of control. The creation of custom robots will be most likely entertaining enough to keep people creating robots just for fun. Moreover, in future version, a history of robot matches might be stored online which will give the creators of robots "bragging rights".

Lastly, the player is also faced with the choice of trade offs. One cannot have a very fast machine, with huge guns and sturdy armor. Players will have to chose depending on their preference to create their robot. Thus it might be very hard to create the ultimate machine.

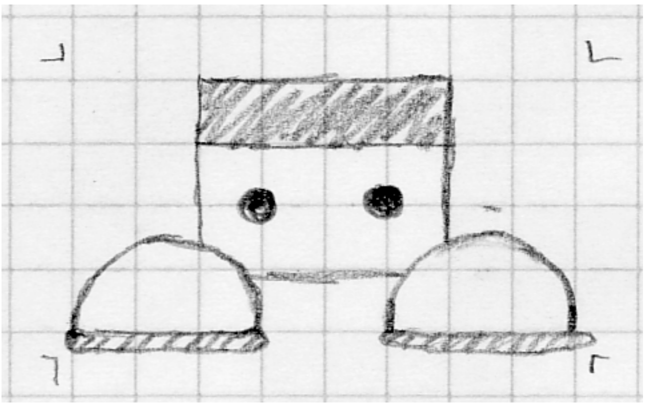
4.2 User Group

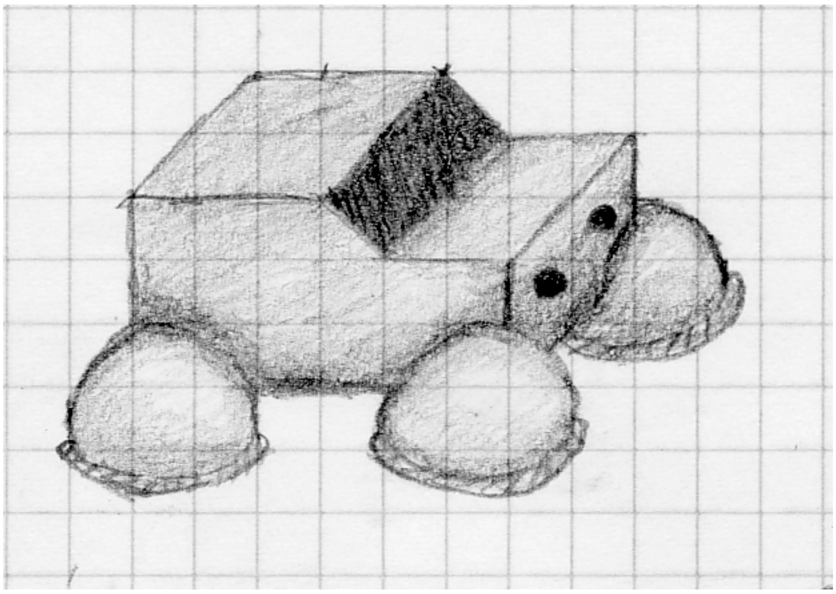
This game should be for a broad audience. The game is aimed to be such that it allows advanced players for a lot of customization, but at the same time, allow new players to create exciting robots without any previous know how. On the other hand, this game will be most appealing to players who prefer, puzzle/construction games as this will be the main focus of the game. The battle phase will always depend on how well a robot is designed.

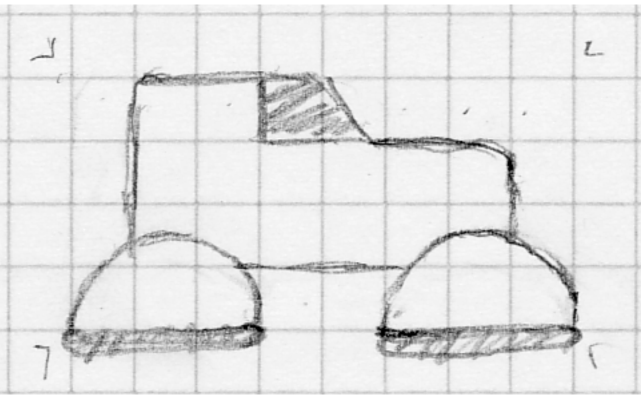
4.3 Evaluation Criteria

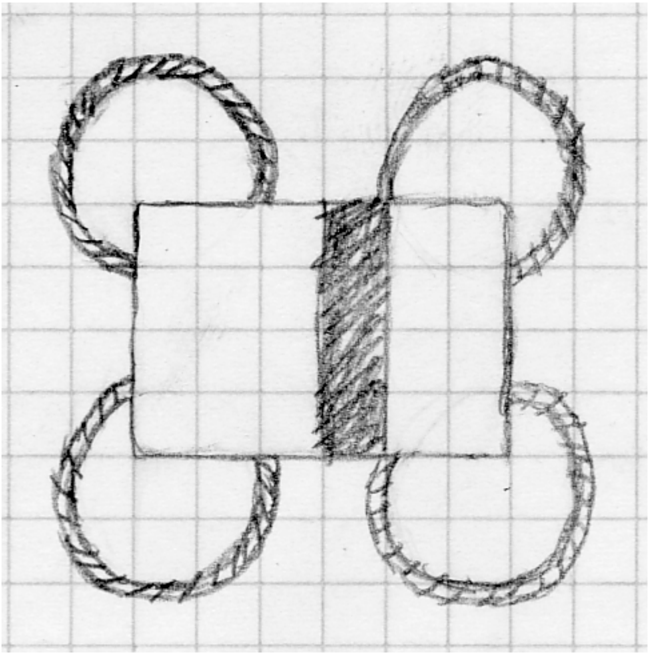
The game will be successful if players can create their own robot, the logic and fight with each other.

A Stomp

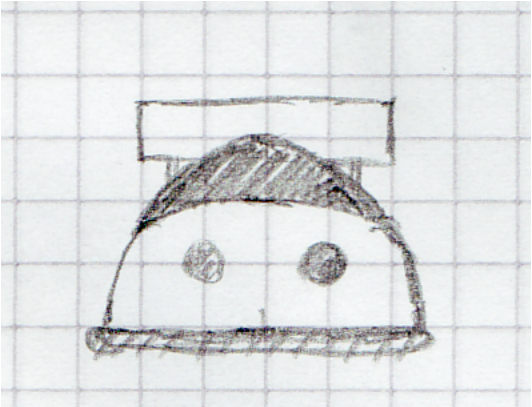


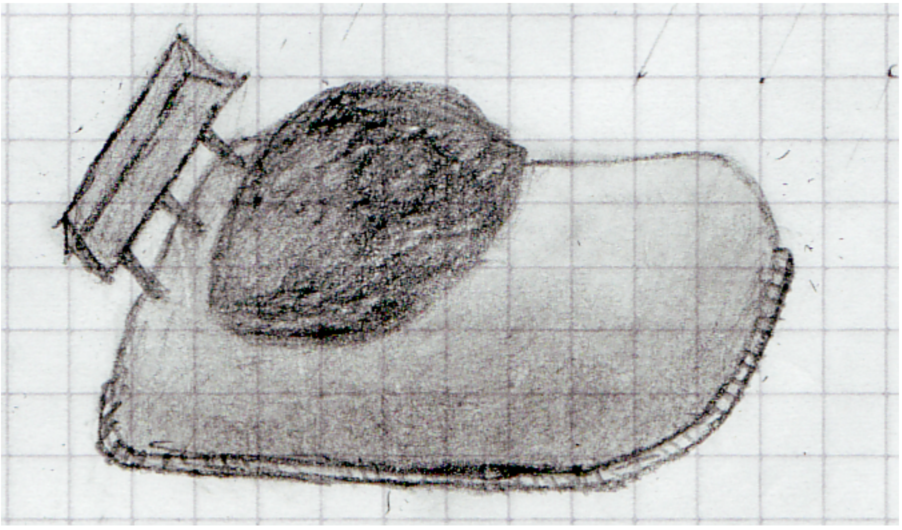


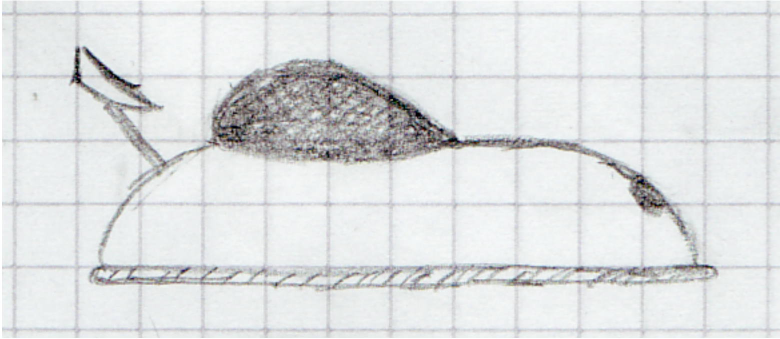


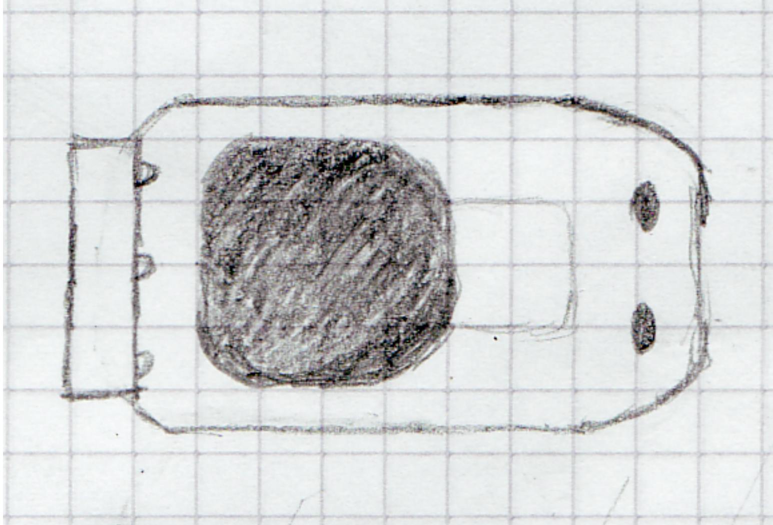


B Galaxy









C Chavelle