

Club Sandwich

# DESIGN DOCUMENT

Desktruction Derby

Michael Brown - [mjbrown@live.com.au](mailto:mjbrown@live.com.au)  
Paul Watkins - [watkinzez@hotmail.com](mailto:watkinzez@hotmail.com)  
Kitman Yiu - [kitman200220022002@gmail.com](mailto:kitman200220022002@gmail.com)  
[10/21/2013](#)

Desktruction Derby is a car combat game where the flow of play is defined by randomly chosen rules, except the player is encouraged to break them in order to receive special bonuses. Breaking these rules changes the behavior of the enemies in the arena, depending on the type of rule that is broken, the amount of rules that are broken in a designated amount of time, and the type of enemy that is in play.

## Gameplay cycle

At the start of the game the player is placed in an arena with various obstacles and cover with a small number of enemy cars. The enemies will be randomly picked from a pool of different classes.

As the player destroys enemies by ramming into them at speed, rules chosen randomly from a pool are put into play. As long as the player adheres to these rules and continues to destroy enemies, new rules will continue to be put into play and the enemy capacity of the arena will increase.

Inevitably it will become harder and harder to adhere to all of the rules in play; indeed, some of the rules may contradict each other. Once a rule is broken, the enemies will go into a berserk mode defined by their enemy class for a limited time. If only one rule is broken, only some of the enemies will enter this berserk mode. If multiple rules are broken in succession, the enemy response will be greater and more fierce.

The player tracks their progress by a point system. A base point reward is given if the player destroys an enemy when no rules are broken. As rules are broken in succession, a multiplier effect is put into play which can earn the player big points if they survive the deluge. One broken rule will multiply the point reward by 2, two broken rules will multiply it by 4 and so on.

The game ends after a specified time (3 minutes by default) or if the player is destroyed.

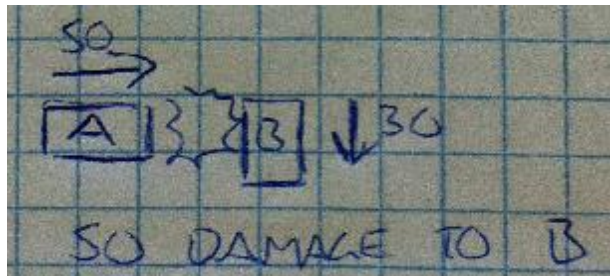
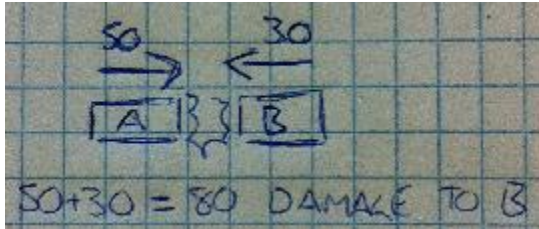
The optimal result will be a game that encourages a mix of offensive and defensive play in order to build up rules, then break them and survive the consequences.

## Core combat mechanics

In order to destroy enemies, the player rams their car into other cars at speed. In a collision between vehicles, the faster vehicle deals damage to the slower vehicle. The amount of damage dealt is calculated by taking into account the difference in speed between the two vehicles, as well as their relative directions.

## 2

## CORE FUN/MECHANIC



## 3

## GAMEPLAY

### Base Enemy AI

Each enemy will drive around the arena randomly. If they come within a certain distance to another vehicle (player or other enemy) they will attempt to ram into the vehicle. This is their line of sight.

To encourage the enemies to go for the player more than each other, their line of sight for the player will be larger than their line of sight for each other.

### Rules

These are the rules that can be put into play at various intervals.

#### Destroy only X

- The player must destroy enemies of a certain class.

#### Don't destroy X

- The player must not destroy enemies of a certain class.
- The player may damage enemies of the given class without destroying them.

#### Destroy alternate enemies

- The player must destroy enemies by alternating between classes.
- Don't destroy enemies of the class one immediately after another.

Destroy X enemies in Y time

- This rule counts down from Y time, and the player must destroy X enemies before the timer reaches zero.

- This rule can be 'finished', so it may be removed from the rule list once completed.

- Alternatively, X and Y can be reset once completed, with Y being slightly less time than before.

Stay above X speed

- The player must keep their velocity above a given speed.

Stay below X speed

- The player must keep their velocity below a given speed.

Drive in reverse

- The player must drive in reverse.

## Enemy Class Types

Each enemy class has their own model and color for easy identification, and are defined by the nature of their berserk mode.

Catherine wheel

- This enemy begins to spin on its axis and bounces around the arena.

- The more rules that are broken, the faster the enemy bounces around the arena.

Swarm

- The enemy goes straight for the player no matter how far away they are.

- The more rules that are broken, the faster the swarm becomes.

Invisibility

- The enemy's goes transparent by a certain value.

- The more rules that are broken, the more transparent it becomes until it is completely invisible.

Multiplier

- The enemy splits into two when a rule is broken.

- The newer enemies are smaller and faster, keeping the same damage value as their parent.

## Arena

The game takes place in one level, designed to look like an school table top. Walls and obstacles will fit in with this theme (pencils, erasers, rulers etc).

# 5

## TARGET AUDIENCE

---

The length of a round being relatively short allows for casual play. Ideally we'd want to port this to a handheld device to allow for short bursts of play, but for the moment we're concentrating on PC and console. Maybe Vita? Maybe Vita.

# 6

## TARGET PLATFORMS

---

In using XNA, we intend to design an implement for PC, then convert to 360. Controls should be easy to transfer to a controller, though it won't offer complete analogue movement using the control stick.

# 8

## CONTROLS

---

Standard WASD directional controls and a mouse controlled menu. Accelerating and decelerating is done directly through the directional controls (forward and backward). All of these will be controlled with the control stick on the 360, using the buttons for menu select.

# 9

## KEY PROGRAMMING CHALLENGES

---

Our original intent was to work entirely in 3D, with sloped terrain and a third-person camera. It wasn't long before we realized that implementing that in the time we have would not be feasible. That said, we still want to use 3D model designed by fellow first year art students. In order to do this, the logic of all the game's systems will remain in two dimensions, with the camera being a top down view that will remain static in regards to its height above the level. This way we can easily design and implement the game with the 2D skills learnt this year, while using cool looking 3D models that scale easily.

The core mechanic relies on lots of physics and collisions. Naturally, this means that the implementation of this section has to be solid. Extensive testing with multiple vehicles at different speeds will have to be done to make sure that the underlying logic is sound.

We intend to define level data in an XML format and then load them into the game to

build the level. This is an area none in the team are overly familiar with, which makes it a slightly risky proposition. Research into how this is done will be required, as well as attempting to implement it in a separate application first to make sure it works as intended.

## 10

## KEY ART CHALLENGES

Funnily enough, we can't art! So we have to find people who can art, namely fellow first year art students. The main hurdle here will be our ability to effectively communicate the asset requirements we need for the game. Taking into account the limited amount of time they will have to create the assets, keeping the scope in check while also making sure what is produced is to spec will be paramount.

## 11

## GAME PLAYER STATISTICS (HEALTH, DAMAGE, ETC)

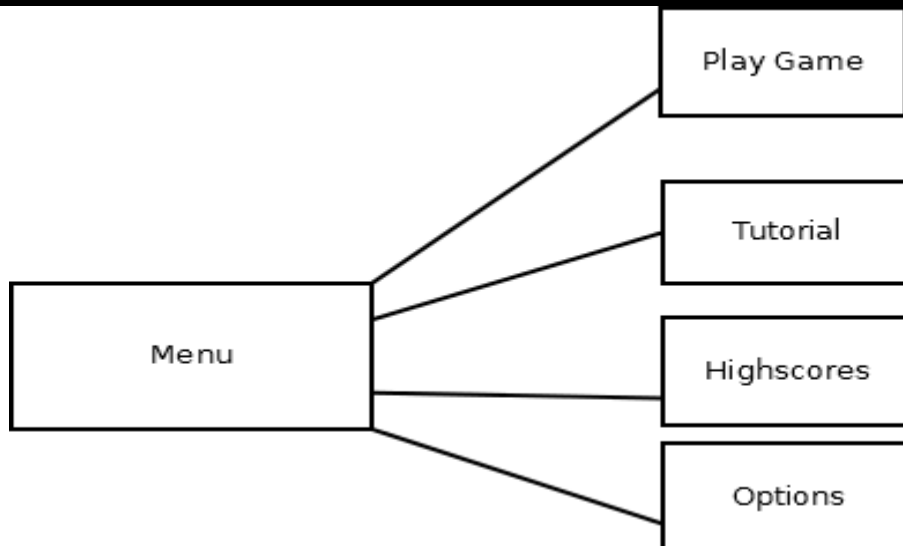
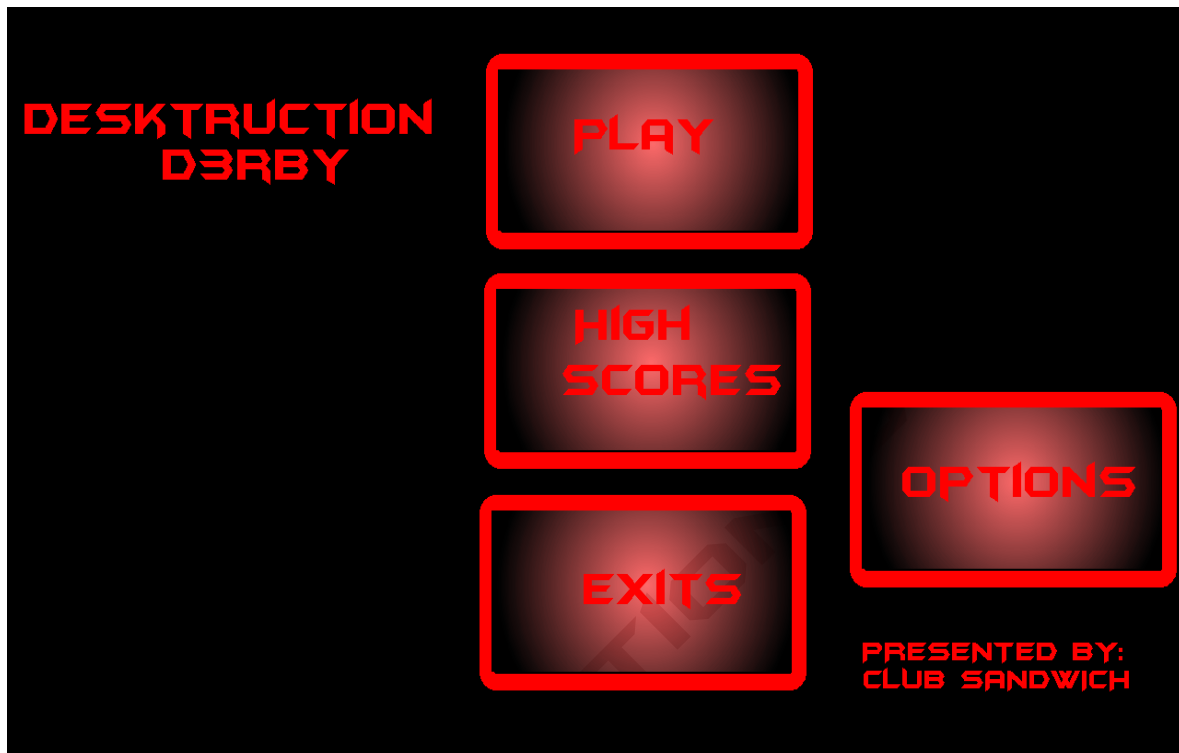
The players progress is measured by their score and health. The player scores points by damaging and destroying other vehicles. One hundred points are issued for destroying, and ten points for damaging another vehicle. The score gains a multiplier for each rule that is broken, which is doubled for each extra rule broken.

The health is displayed as six points around the car, each having 5 points of health, displayed as Green, Yellow, Orange, Red and Black. If a damage box is black, it will affect your driving. The front boxes will make the car always steer to the side and the back two will decrease the maximum speed of your car. If three of the damage boxes are black, your car breaks and you lose the game.





# MENU SYSTEM



The Settings will have turn on/off shader , on/off music, on/off sfx, on/off background music ,set resolution , and of course back button.

# 14

## ART STYLE

The art style is very similar to Micro Machines V4, in that you are toy cars in a real scale world. The textures have vibrant colors and the objects in the game look like toys. You play on a table-top style environment. The enemies are each different colors, Red, Green, Blue and Yellow. They are a distinguishable shape from the players car.





# 17

## WEAPONS, POWER-UPS, ETC

---

Your car is the weapon, being that you have to crash into other vehicles to damage them. There are no power ups, however, there is a wrench that will fix some of the damage on your car when you collect it.

# 18

## ENVIRONMENT

---

The environment will feature a real scale objects, so that you feel small in comparison. The levels will feature environments such as a desk-top, which will have objects such as books, keyboard, mice, pencils etc. and a living room which will have toys scattered around, lounge, television etc.  
The levels will be designed as arenas.

DESTRUCTION DERBY