Design Document

Version 1.3

Sam Pospischil
pospi@spadgos.com
http://pospi.spadgos.com

Contents

Overview

Philosophy statements

- A game should be simple and easy to play.
- A game should be fun and not take itself too seriously.
- A game should give the player as many options as possible on how they want it to be played.

What is the game?

Furious Steals is an arcade racing game, with an emphasis on open environments and checkpoint-driven scoring. It will consist of unique game modes that cannot be played in other racing games, and will be built as a mod on the UT2004 engine.

Why create this game?

There is a lack of simple and lighthearted racing games that anyone can pick up and play easily. The development team also shares an interest in creating such a game – this is a project of passion.

Similar Games

The game most comparable to Furious Steals is Midtown Madness 2. Indeed, much of our gameplay aesthetics are mutations of ideas displayed in this game. Other influences are Carmageddon and Flat Out.

Game Locations

There will be a wide variation of settings for the game as we are leaving the environmental design up to the imagination of the map authors. Generally, the game will take place in large urban environments, full of shortcut possibilities, large jumps and z-axis gameplay.

What do I control?

Wheeled vehicles.

Number of Vehicles

The number of selectable vehicles will be up to the community in general. We are aiming for around 5 initially, but are creating a system where others can be added easily by modders to expand the collection. Each vehicle will have a selection of skins that players can choose to make themselves appear unique next to vehicles of the same type [see general features].

Main focus

The main focus will be on speed, control and evasion. Players must race to particular areas on the map, finely handling their vehicle to evade the attempts of other players to crash into them.

What's different?

Unique gamemodes, simple gameplay and a unique visual style.

Gameplay

Overview

The most common theme to the gameplay choices in Furious Styles is that they all will involve some amount of vehicle/vehicle collision interaction. Players must crash into each other to obtain game-critical objects and then avoid others once that object is obtained.

In each game mode vehicles may or may not be damageable, as set by the server [see server configurable features]. If a vehicle is destroyed, it will be reset at the spawn point nearest to where it was destroyed.

The game modes planned have been designed with the object of being different than game modes you would see in other racing games. The aim is for players to have a unique racing experience that they cannot get anywhere else.

Below are the planned game modes, with a more indepth explanation of each.

Steal the Loot

The flagship gametype, Steal the Loot is similar to a game like cops and robbers except that all players are playing for their own gain.

A pile of gold will be spawned on the map at a random predefined position [see game actors]. These locations will be specified by the mapper, and one will be randomly chosen at runtime to place the gold on.

Players must then race to the gold and touch it with their vehicle (a-la CTF) to pick it up. The gold can be transferred between vehicles through collisions, so the player holding the gold should be avoiding other players whilst others should be attempting to crash into the player with the gold.

The goal for a player holding the gold is to take it to a hideout. Hideouts spawn along with the gold, and there will be one hideout present in the game at a time.

Once the player holding the gold has driven through a hideout, they score a point and both the gold and the hideout are spawned anew.

The game is over when a player reaches the score limit or the time limit expires.

Cops and Robbers

This will be an extension of Steal the Loot where the game is played in teams. There will be two hideouts now, one for each team. These hideouts will try to position themselves with the gold and players in the middle, to create a tug of war between both teams.

Player teams will be easily identifiable by the vehicle skins. The vehicle skin will be locked on a team basis so that no matter what vehicle is chosen, all players on the same team will be similar enough to know that they are working together. In this game type, players do not have access to the standard vehicle skins, instead they have a choice of one for each team.

Note: This gametype is considered optional and may not be implemented if time is a consideration.

Hold the Bomb

This gametype is played in 'rounds', however there will be no hard restart between rounds as in CounterStrike or other likeminded games. At the start of a round, a random player has a bomb attached to their car [see game actors].

All players will then have a timer on their screen which counts down from a random time (between some minimum and maximum time) until the time when the bomb will explode.

Players will have two distinct sets of points, which I will label 'game points' and 'temporary points'. Game points count towards the player's overall score, whilst temporary points are current for that round and are reset after each round. Temporary points are earned by holding the bomb.

When the timer reaches 0 and the bomb explodes, the unlucky car with the bomb attached is destroyed (and subsequently reset). This player obtains no game points for the round, whilst all other players have their temporary points for the round added to their game points. After this, all temporary points are reset to 0 for the next round. If the bomb is on the ground when the round ends, no players will score any game points for that round.

This process is repeated until the match time limit is reached or a player reaches the match score limit.

The twist to this mode, which makes the game play more like a 'greed' style game, is that the points obtained by holding the bomb increase exponentially with the time the bomb has been active. This means that players must weigh the risk with the reward, as holding the bomb will be worth more points as it is about to explode.

Game Actors

Overview

There are essentially four types of actors in a Furious Steals game - the player-controlled vehicles themselves, a target object (i.e. the gold and bomb), dropoff points (i.e. hideouts) and clutter that provides interactive properties.

Vehicles

These are fairly straightforward. Each vehicle corresponds to one player (or AI opponent). They handle in a very arcade manner, with tightly dampened movement on the pitch axis to prevent flipping midair, and loosely dampened roll rotation to allow for exciting sideways flips if a jump is not met accurately. Vehicles may be damaged [see server configurable features], can be reset [see controls], and may use a boost feature [see server configurable features]. They also have a handbrake, the slip of which is dependent on that particular vehicle.

A player's only method of attack is to ram opposing vehicles to transfer game critical objects. These collisions will also transfer damage if allowed [see server configurable features] and can of course be used to inhibit the movement of opponents.

The Gold

The gold is what players compete to obtain in the Steal the Loot gametype [see gameplay]. It is passive in nature and will not move unless attached to a vehicle. The only exception is that it will continue along its trajectory until it lands if the vehicle that was holding it is destroyed. The only way it can become attached to a vehicle is if touched by one – once attached, it is transferred between vehicles when they collide.

If left alone for a certain timeout period, the gold will be moved to a new spawn point to prevent unreachable locations from halting a match.

There is some simple logic with regard to the spawning of the gold. It will look for a position that is a compromise between furthest from the player that just scored and closest to the losing player. In this way, the game remains balanced so that the losing player has the best chance at getting the gold and it is difficult for players to go straight from a checkpoint to the next gold drop.

The Bomb

The bomb is the main element in the Hold the Bomb gametype [see gameplay]. Unlike the gold, the bomb starts off attached to a random vehicle. The bomb can be transferred between vehicles via collisions.

The bomb has a timer, which counts down until the time when it will explode. All players can see this timer. Upon exploding, the vehicle holding it goes along with it.

If the player holding the bomb manages to kill themselves through collisions with the environment, the bomb will be left on the ground at the place where they died. Vehicles can then pick it up by running over it.

Hideouts

Hideouts take the form of a giant garage that players must drive through. They are walled in on most sides so that players are required to prepare their approach, rather than just driving through in any direction.

Hideouts will come in a few variations – double doored (with a door at each end that can be driven straight through), four doored (to be placed in intersections), and midair hideouts which may just be a ring that must be jumped through.

The mapper is able to specify predefined locations for hideouts, which the game will randomly select from at runtime. Which variation each should be is also specified, along with the hideout's rotation. This can either be fixed, for when the hideout is placed along a street or at an intersection, or random for hideouts that should be rotated differently each time.

Hideouts will try to spawn so that the body of players lie between it and the gold. The maximum possibility for evasion and collisions is thus maintained by the position of the hideout.

Hideouts exist in three distinct states once active – a leading in state, waiting state, and leading out state. The two leading states are transitions between the hideout not existing and existing, and are basically complicated animations that will play. On lead-in, the hideout will burst out of the ground, throwing anything above it (including players) out of the way and spewing dirt and particles into the air. It will then wait until the gold is driven through it, and explode/dissolve into the air as if it was never there. The basic idea with these leading states is to have as much visual noise going on as possible.

Hideouts also have indicator beams coming out the top of them to show players where they are from long distances away.

Clutter

Clutter is the domain of physics objects and breakable items that populate the game's environments. These come in two flavours – clientside and serverside.

Clientside clutter has no effect on player's vehicles and is there purely to add visual noise. Breakable objects that simply shatter on impact without slowing vehicles and physics actors such as witches' hats that fly into the air without any impact on the game are clientside clutter. Clientside clutter has no mass to impart upon other objects.

Serverside clutter is stuff that has an impact on the movement of player vehicles. Serverside clutter may reset itself after being disturbed or stay active constantly [see server configurable features]. Breakable serverside clutter simply explodes when hit by a vehicle and slows the vehicle down by a certain number of units. Serverside physics clutter is heavy moveable objects such as dumpsters and bins that will be thrown around the level and impact with vehicles. Serverside clutter will throw vehicles when it hits them and will also impede their progress.

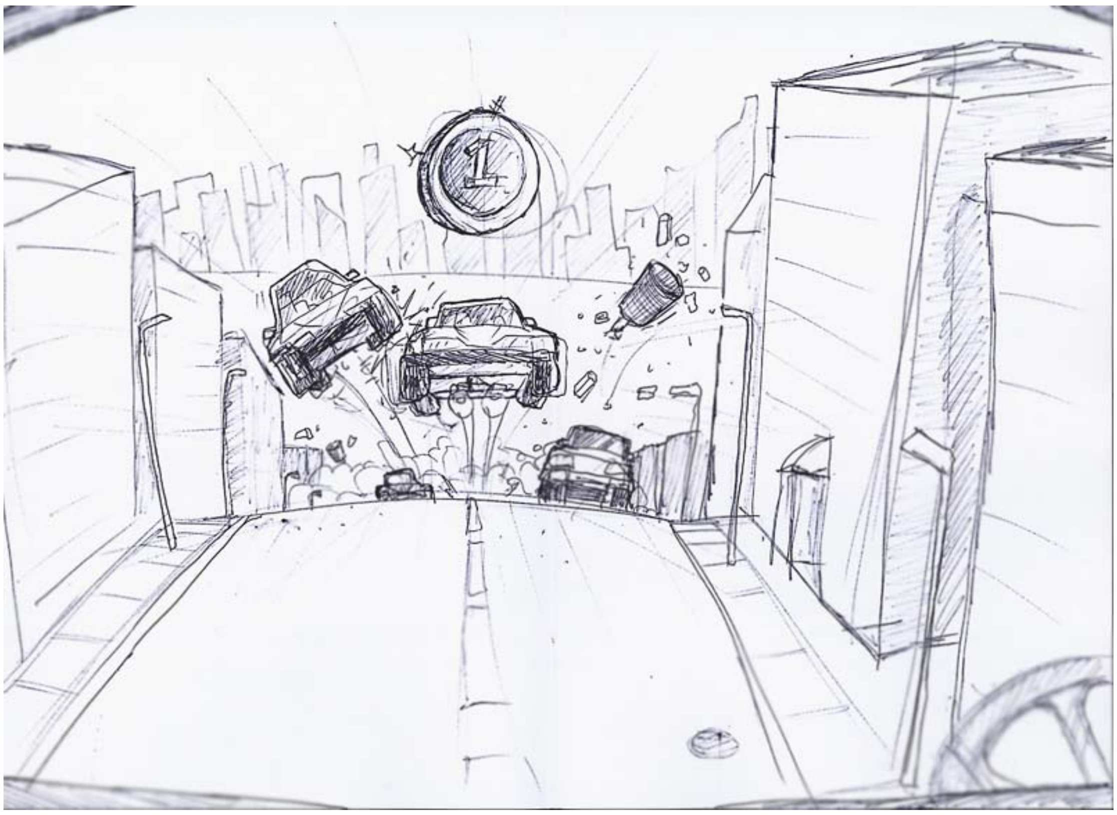


Fig. A – Ingame action concept

Options and Customisation

General Features

- Vehicle selection. Players can choose a vehicle to drive from a list. The list gives them various useful statistics and a preview of what that vehicle looks like.
- Vehicle skins. Players can choose a skin for their vehicle to make it appear different than other vehicles of the same type.

Server Configurable Features

- Vehicle boost. This is used to enable or disable a nitrous-like feature that allows players to add an extra kick of speed to their vehicle.
- Damage. This can be disabled to have vehicles that are invulnerable to collision damage. When damage is enabled, vehicles can be destroyed. When destroyed, they will spawn at the closest spawn point to their point of destruction.
- Vehicle flipping. When enabled, vehicles will flip back onto their wheels when they come to rest if they are toppled over.
- Allowed vehicles. We plan to include many different vehicles of various types. As such, players may not always feel the vehicles are balanced and may opt to include a subset of the entire selection.
- Clutter resetting. Used to reset physics items and breakable props after a certain time to save on processor resources and to clean up the game environment.

Editing

Any editing for Furious Steals can be done within the standard UT2004 engine toolset. This includes the use of unrealEd for mapping and 3D packages for modeling game objects and new vehicles. Tutorial documents will be created to explain the specifics of modding for the game vs modding for UT2004 in general.

Camera

The camera will be in either 3rd person or 1st person mode, centered on the player's vehicle.

In third person, it will be positioned with regard to the vehicle's velocity so that it always rotates facing the direction the vehicle is traveling in. Its movement will be sufficiently dampened that no sudden movement will occur, and when the vehicle reverses it will not invert its view but rather rotate around on the yaw axis.

The view position in third person will be quite analogue so that players can position the camera at the exact location that they prefer. There will be two camera controls, one to zoom the view in and one to zoom out. Minimum and maximum zoom will be clamped between two values.

When the view is zoomed all the way in, the view will change to a first person perspective. This view will be taken from the vehicle's nose to provide a bumpercam view of the action. The camera in this case will never rotate, and remains fixed to the rotation of the vehicle.

The camera will also have a few effects applied to it at different times. Motion blur accompanies the boost of the player's car, but can be disabled if not desired. The camera will also shake violently when the car impacts with objects.

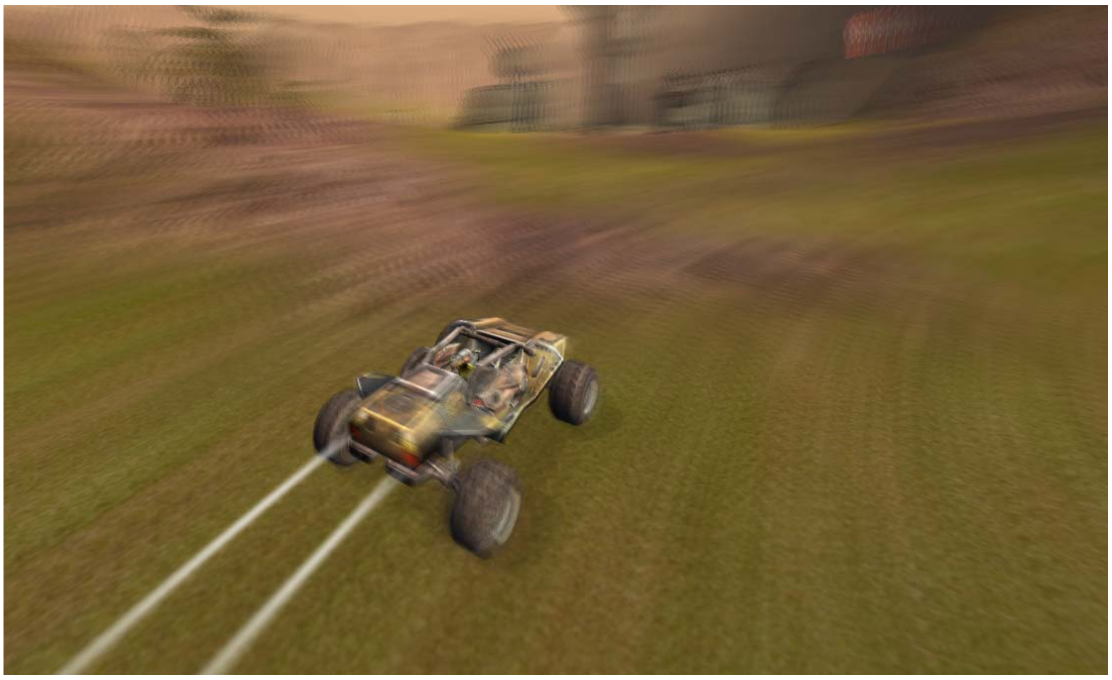


Fig. B – Motion blur and camera turning slowly to meet vehicle's new orientation

Controls

Furious Steals will be controlled entirely with the keyboard, with the exception of menus and dialogs. The aim is for a tight control scheme to be maintained with a minimum number of required inputs. All controls may be bound to other keys, but defaults have been chosen for obviousness and simplicity.

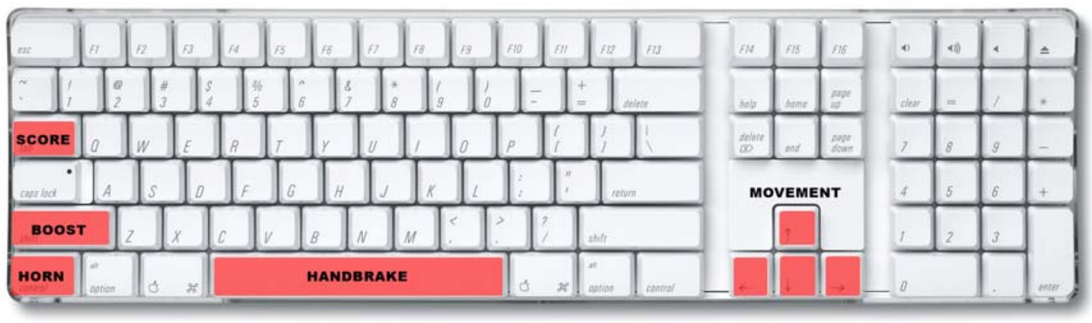


Fig. C – Default control scheme

Vehicle movement is controlled via standard arrow keys. Left and right turn the wheels, up accelerates and down shifts into reverse.

Vehicle handbrake is controlled by depressing the space bar. They are also able to beep their horn by pressing control.

A boost [see server configurable features] is initiated with the shift key. Depressing this key adds a velocity impulse to the vehicle and drains boost fuel reserves. If the boost fuel is exhausted, the player must wait until it recharges to a certain amount to boost again. Otherwise, they may boost again immediately.

The World Layout

The layout of each environment is particular to the tastes of the mapper. Following is a basic layout plan for the Brisbane map to be finished upon first release of the mod. Note the structured, block-like layout of streets and urban style of the environments.

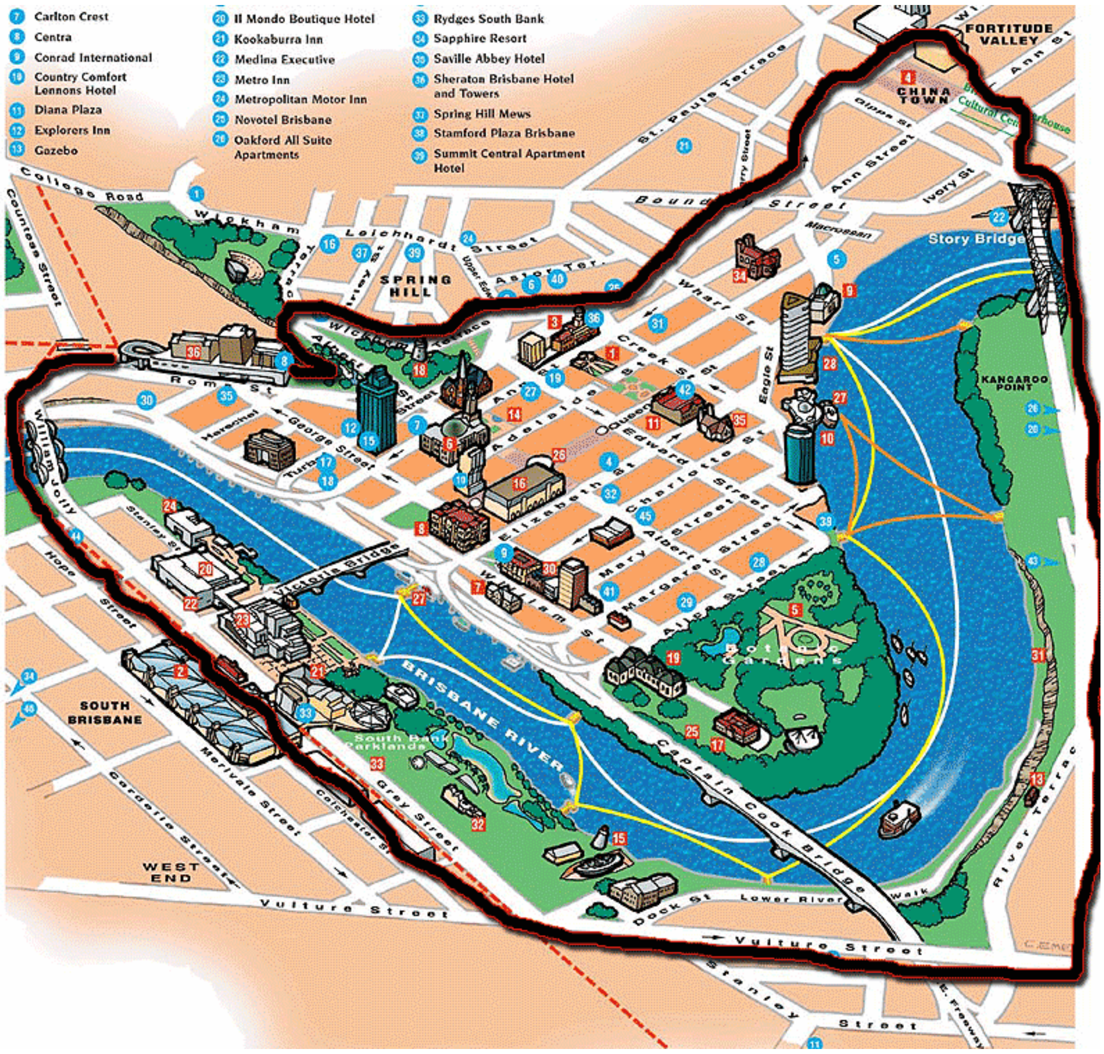
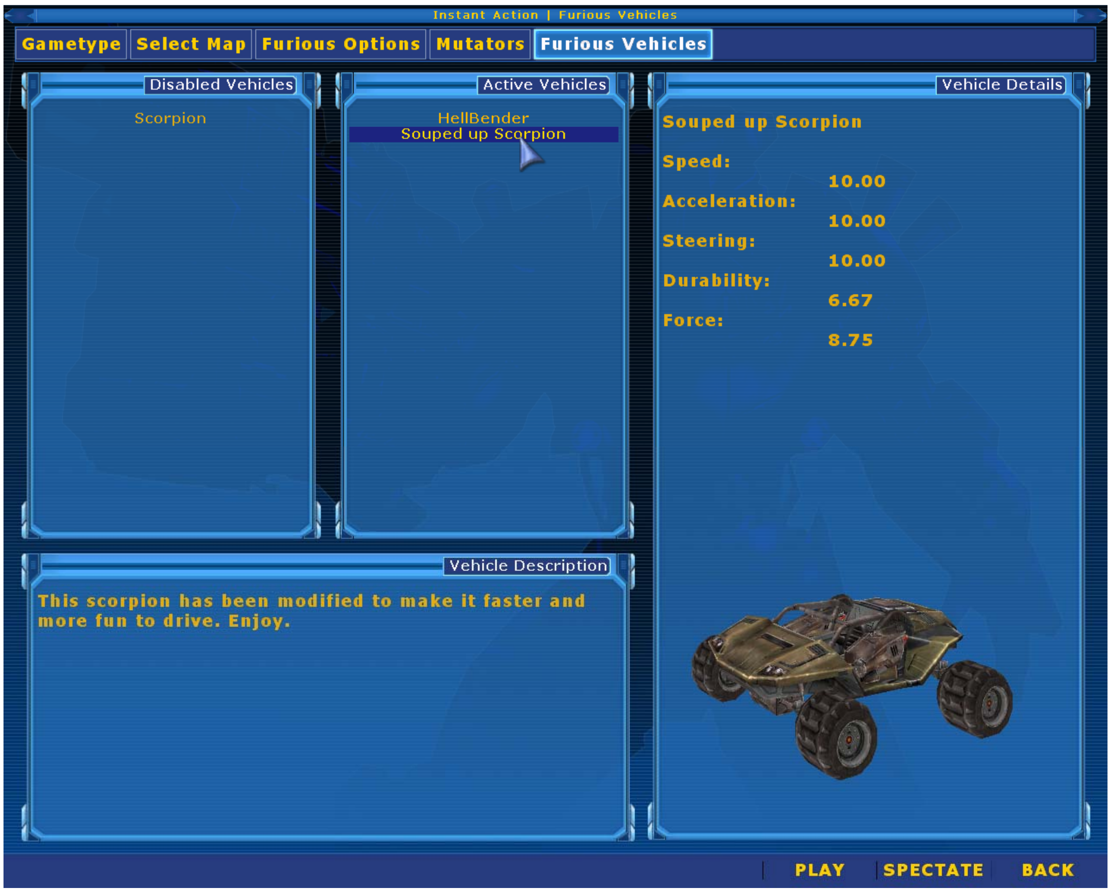


Fig. D – Layout plan for city map

User Interface

The body of the GUI at the frontend of the game will be unchanged from that of UT2004 with regards to functionality. However there will be an extra screen similar to UT2004's mutator configuration menu where the game creator can specify which vehicles to allow in the game.



Disabled Vehicles: Scorpion

Active Vehicles: HellBender, Souped up Scorpion

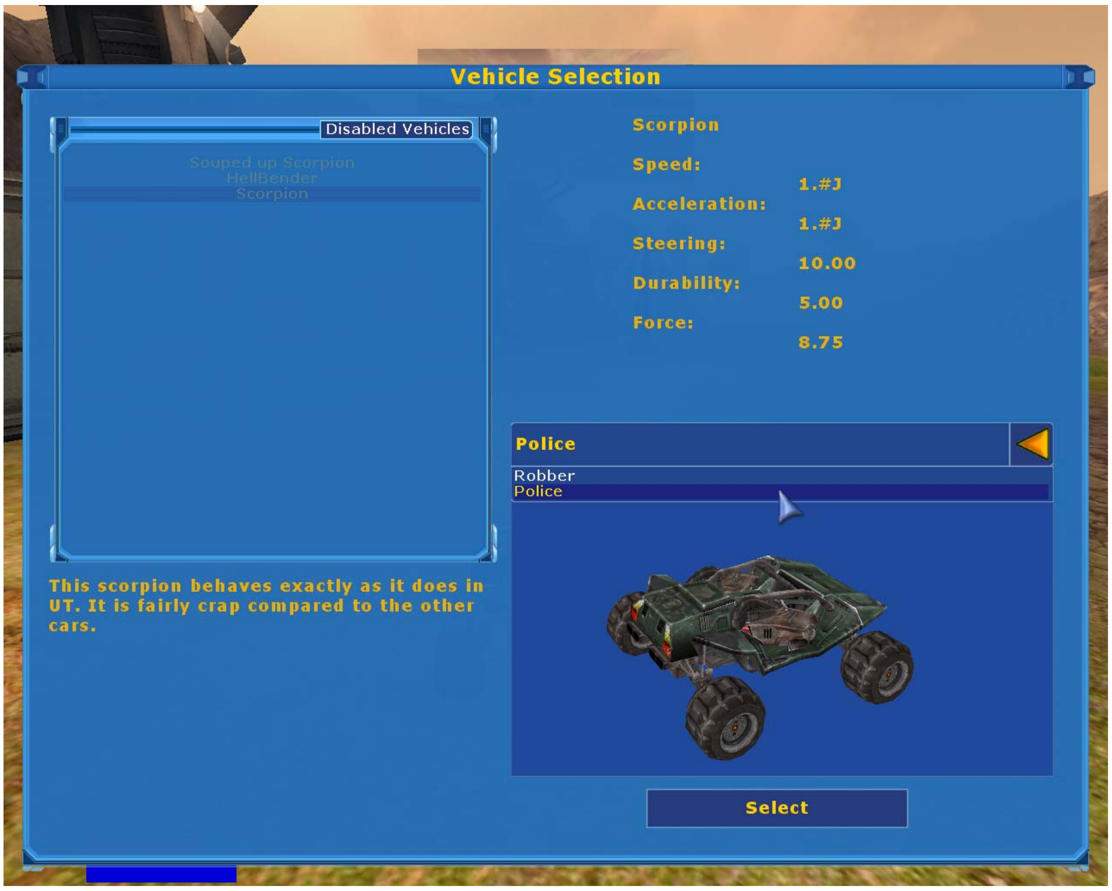
Souped up Scorpion

Speed:	10.00
Acceleration:	10.00
Steering:	10.00
Durability:	6.67
Force:	8.75

This scorpion has been modified to make it faster and more fun to drive. Enjoy.

Fig. E – Vehicle configuration menu layout

Once inside the game, all players are presented with a menu to select their vehicle. This menu is very similar in style to the configuration menu, but adds the vehicle skin selector.



Vehicle Selection

Disabled Vehicles

Souped up Scorpion
HellBender
Scorpion

Scorpion	
Speed:	1.#J
Acceleration:	1.#J
Steering:	10.00
Durability:	5.00
Force:	8.75

This scorpion behaves exactly as it does in UT. It is fairly crap compared to the other cars.

Fig. F – Vehicle selection menu layout

Finally, the player HUD will display all relevant game information to the player at all times. It will include an optional rear view mirror and minimap, as well as damage and boost fuel indicators if these features are enabled in the current game – thus the elements visible on the HUD will indicate to the player the way in which the current game is configured.

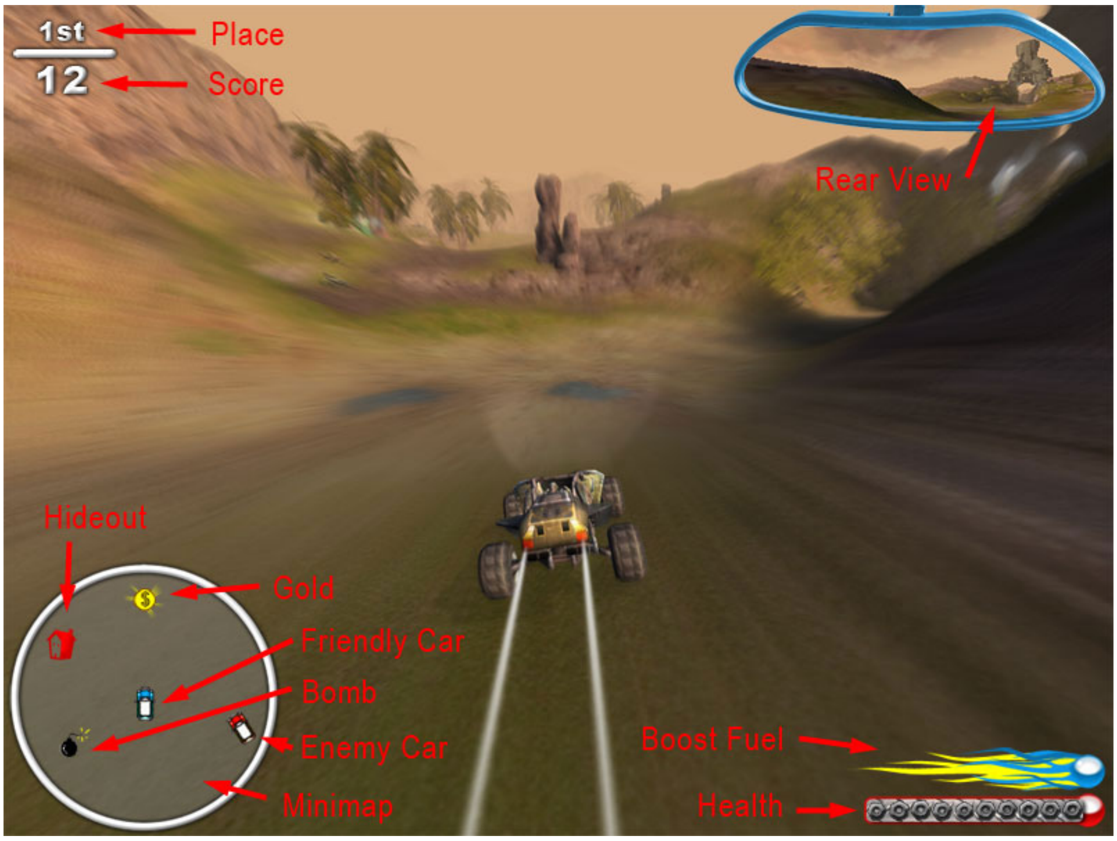


Fig. G – HUD layout mockup

The minimap will display different icons for enemy vehicles, friendly vehicles, the target object (gold or bomb) and hideouts. The position of these icons relative to the center of the minimap circle will relay their location in the world to the player.

Visual Design

Furious Steals will be styled in a primarily cartoon way, reminiscent of much earlier titles such as Micro Machines.

Vehicles will be designed ‘off the factory floor’, with shiny, chromed appearance and no dust or scratches visible. They will be of unrealistic proportions, set to a cutesy and bubbly style.

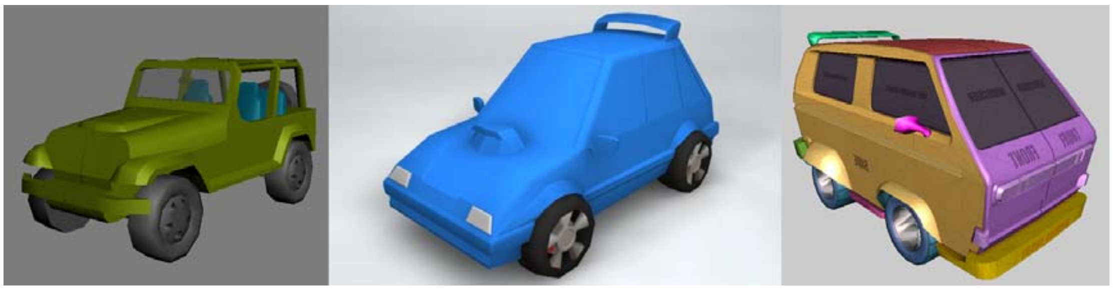


Fig. H – Vehicle geometry style renders

Environmental design will utilize large areas of flat colour with minimal detail. Realistic city environments will be created using source photography and quantization of those images to provide a cleaner and less cluttered look.

By contrast, a great amount of visual noise will be present with the game’s particle effects. Impacts between vehicles, dust kicking up from tires etc will fill the areas of action with lively and exciting explosions of colour and motion to draw the player in. These effects will vary depending on the surface of contact, and with 11 distinct surface types planned (rock, dirt, metal, wood, plant, flesh, ice, snow, water, glass and sand) there will be more than enough variation to the effects generated.

The upshot of this design principle is to draw the player in to any action that may be happening. The environments the game takes place in are very static, whilst a great amount of motion and noise is generated in areas heavily populated by player controlled vehicles via particle effects and interactive objects.

Music Scores and Sound Effects

Any music the game features will be designed to both add tension to the experience and be a lighthearted and 'fun' sounding track. Music will be done on a per-map basis, with different music for each environment. A further idea is to have two tracks for each map – one for regular play, and another track to build tension (similar to the music dynamic in Super Mario, perhaps) that will play when a player is close to scoring a point.

Beyond environmental sounds, most of the game's aural nature will stem from the behaviour of the vehicles themselves. Of the 11 distinct surface types defined in the game, each will have different sounds for different interactions with the vehicle. Wheel slip sounds, chassis scrape sounds and impact sounds for each surface provide a diverse sound bank for the game.

Additionally, there will be sounds for game events (such as gold captures) and of course ambient engine noise for the vehicles which will vary in pitch as they change speed.

Note: At present there is no sound designer on the development team, which may make fulfilling these requirements difficult. In this case, stand-in sounds will be used from UT2004 and other games.

Miscellaneous

Other Current Projects

'Rock Blunderbus' – Torque Engine 3D platformer title for consoles

Past Projects

'Escher Physics' – UT2004
'Marshmallow Duel : Mowbray's Revenge' – UT2004
'Instagib : Source' – Half Life 2
'Soldat UT' – UT2004
'MarBALLS engine' – Flash
'Somos Player Model' – UT2004
'Excessive TV' – Tribes Vengeance

Other Contributors

Various assets and diagrams within this document were contributed by other team members, as follows:

Fig. A – Ingame action concept © Michael Lane
Fig. D – Layout Plan © Travis Draper
Fig. H – Vehicle Geometry:
car © Michael Lane
jeep © Kieran O'Sullivan
van © Daniel Cotter

And finally, all ideas expressed in this document are to a great extent the product of many hours of team discussion, debate and brainstorming. The Furious Steals team is:

Daniel Cotter
Iain Danvers
Travis Draper
Michael Lane
Kieran O'Sullivan
Sam Pospischil

...and will probably be expanded as we go along.