

4 Resultados

Nesta seção são apresentados alguns resultados obtidos em forma de funcionalidades implementadas dentro do contexto do jogo.

4.1 Interação

As figuras a seguir ilustram as mudanças ocorridas na interface do Microbacias a fim de adaptar ao novo módulo *game*. Lembrando que o grande objetivo foi não impactar significativamente na interface original.

A figura 4.1 mostra a interface do Microbacias sem que o modo *game* esteja habilitado, a única diferença a ser notada para a interface original foi um botão adicionado no canto inferior direito com figura de joystick que serve pra habilitar o modo.

Já a figura 4.2 mostra a interface do Microbacias com o modo *game* habilitado. Podemos ver agora vários perfis juntos numa mesma imagem e interagir com todos eles. A barra de progresso da simulação foi retirada e os botões de simulação e de exibição 3D foram desabilitados, por não serem compatíveis com o *game* ainda. O botão de Play foi reutilizado para passar rodadas do *game*.

A figura 4.3 mostra como ficou o formulário de microrregião enquanto o *game* não está rodando. Foi apenas adicionada uma aba extra chamada Estoque que fica desabilitada.

Esta mesma aba é habilitada quando o modo *game* está ativo (figura 4.4) e serve para executar as vendas dos produtos em estoque de cada fazendeiro. Escolhemos a quantidade a ser vendida, o produto em estoque a ser vendido e clicamos em vender. Vendas de estoques pertencentes a fazendeiros que não são players não são executadas e é a última venda que conta; ou seja, se selecionarmos 500 unidades do produto milho e clicarmos em vender e depois mudarmos o valor para 1000 e clicarmos em vender, a última venda de 1000 será executada. Os produtos só são listados se estiverem em estoque.

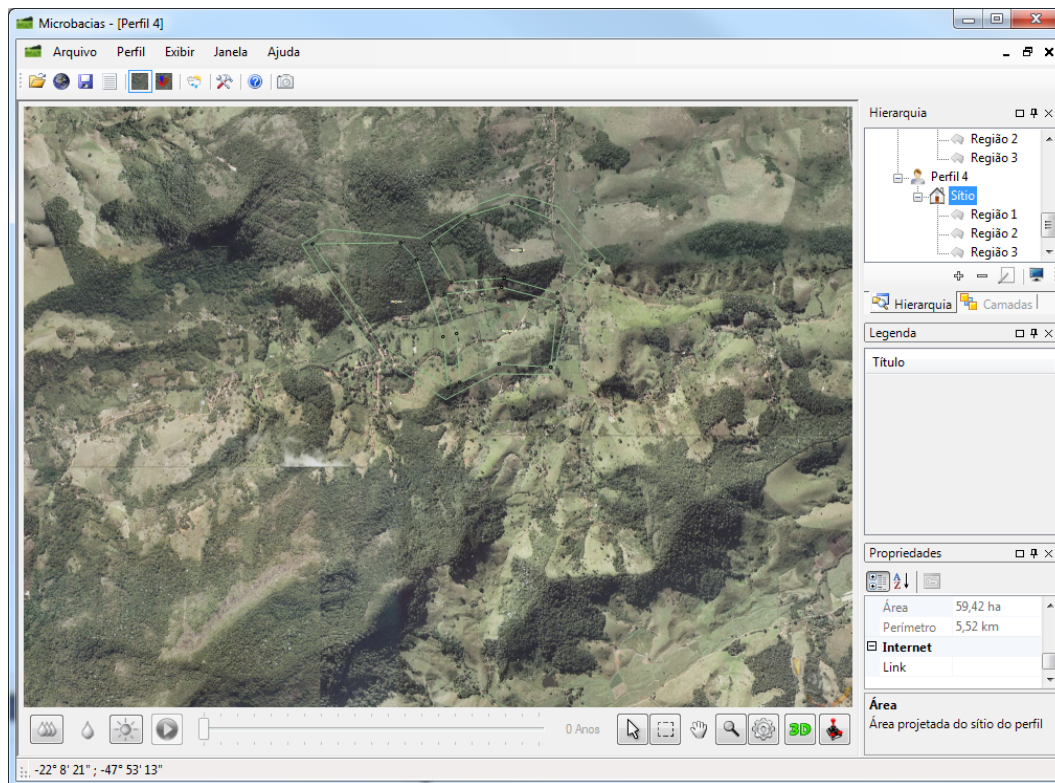
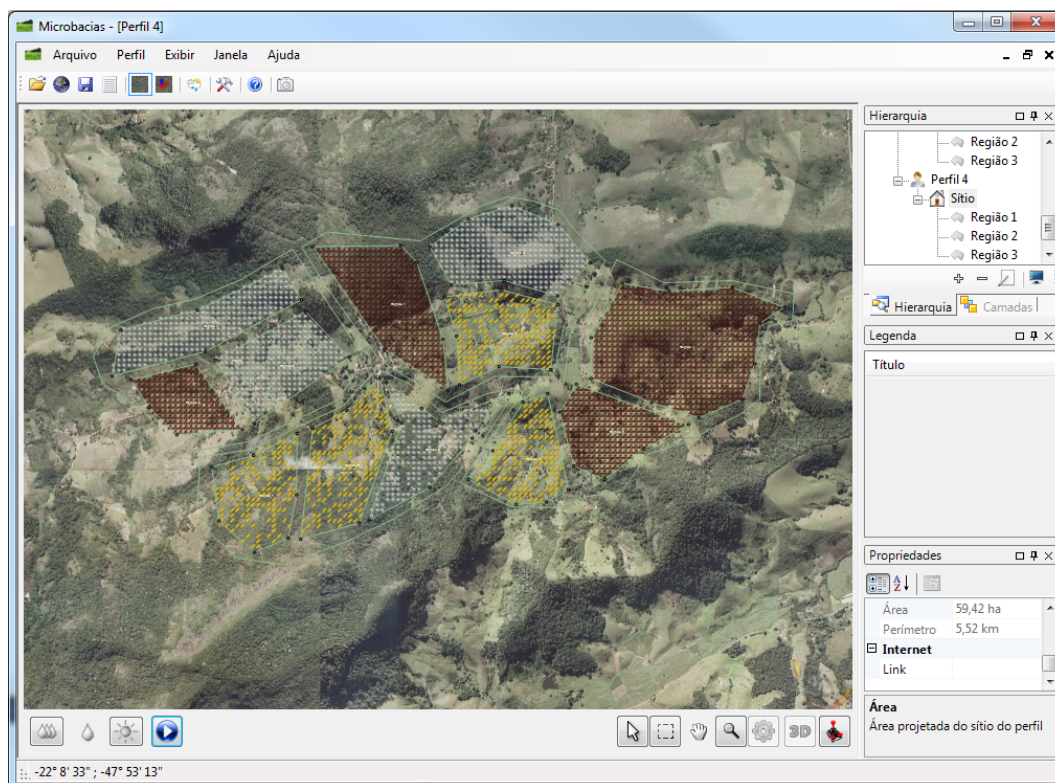


Figura 4.1: Interface do Microbacias durante sua execução normal

Figura 4.2: Interface do Microbacias durante um *game*

Na figura 4.5 o formulário de perfil enquanto não está sendo executado

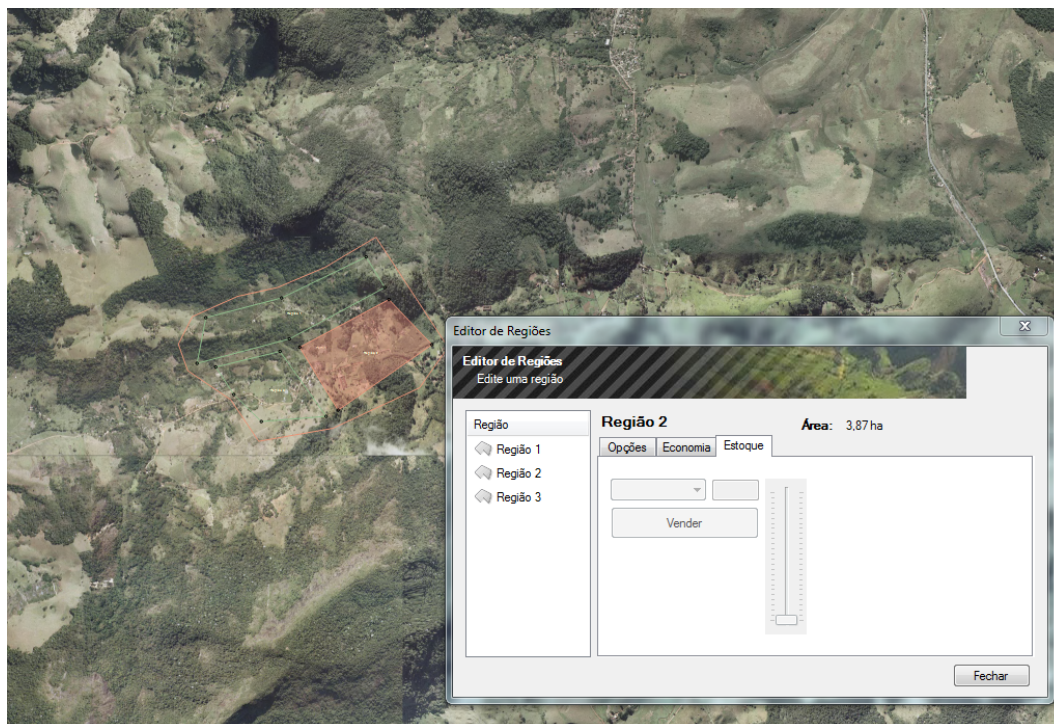


Figura 4.3: Formulário de vendas durante a execução normal do Microbacias

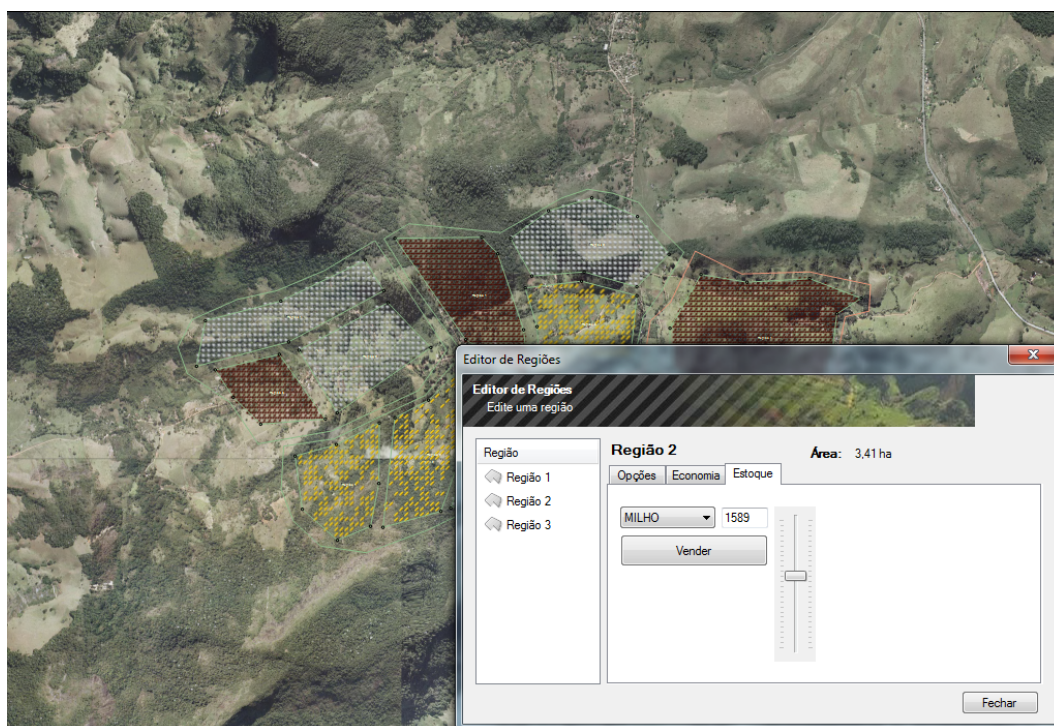


Figura 4.4: Formulário de vendas durante um game

o jogo. Foram apenas adicionados os campos Player, Risco, Exploração e Inteligência que ficam desabilitados.

Na figura 4.6 é apresentado o mesmo formulário de perfil mostrado

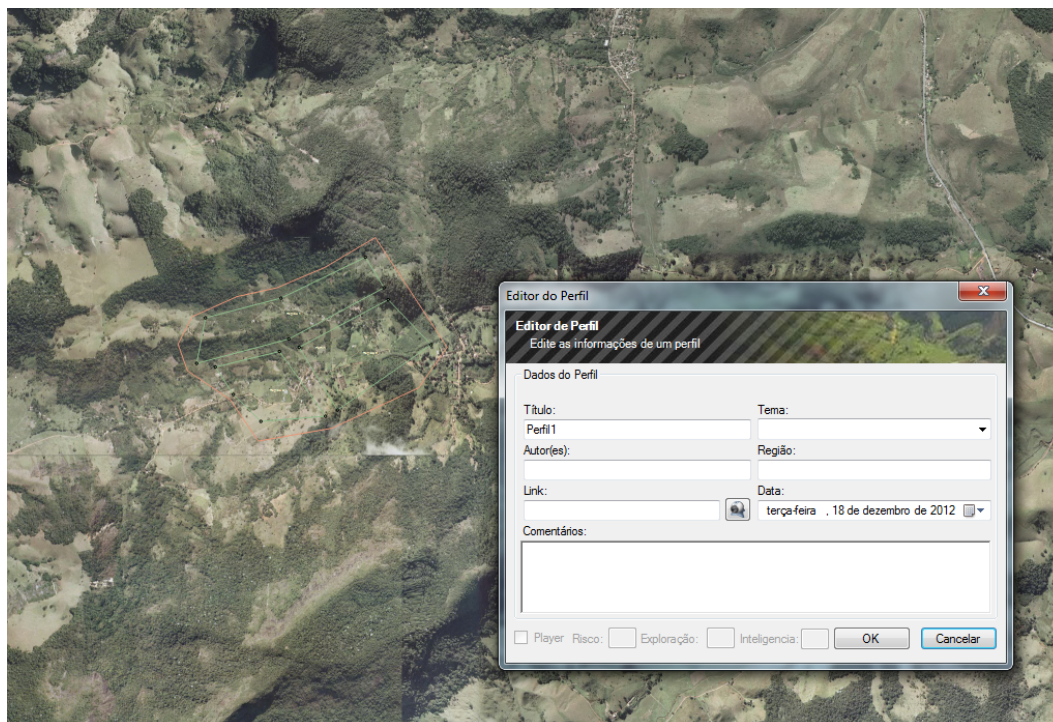


Figura 4.5: Formulário de um perfil durante a execução normal do Microbacias

anteriormente, porém agora com os campos habilitados devido ao modo *game* estar ativo. É possível transformar qualquer fazendeiro em player ou não, em qualquer momento do jogo, implicando em suas ações serem controladas por inteligência artificial ou por um jogador humano. Os campos Risco, Exploração e Inteligência servem para criar variações nas características dos fazendeiros controlados por inteligência artificial. É exibido ainda, no topo do formulário, a quantidade de dinheiro obtida até o momento pelo jogador.

4.2 Instanciação

Por se tratar de um framework, tentamos validá-lo por meio de algumas instanciações, para ver se houve alguma falha na modelagem do framework. Esta instanciação não faz parte de um conjunto de instanciações de uma mesma aplicação, mas sim de um novo jogo criado, são produzidas aplicações distintas quando se altera as configurações do framework. Foram desenvolvidas algumas estratégias para os agentes, e foram preenchidos valores para alguns produtos, tanto para sua produção quanto para o mercado. É obvio que estes testes não são suficientes para validar o framework, mas são um bom indicativo de que o mesmo funciona corretamente, no sentido de que todos os processos têm coerência uns com os outros.

As estratégias, apesar de diferirem no resultado, seguiram um modelo

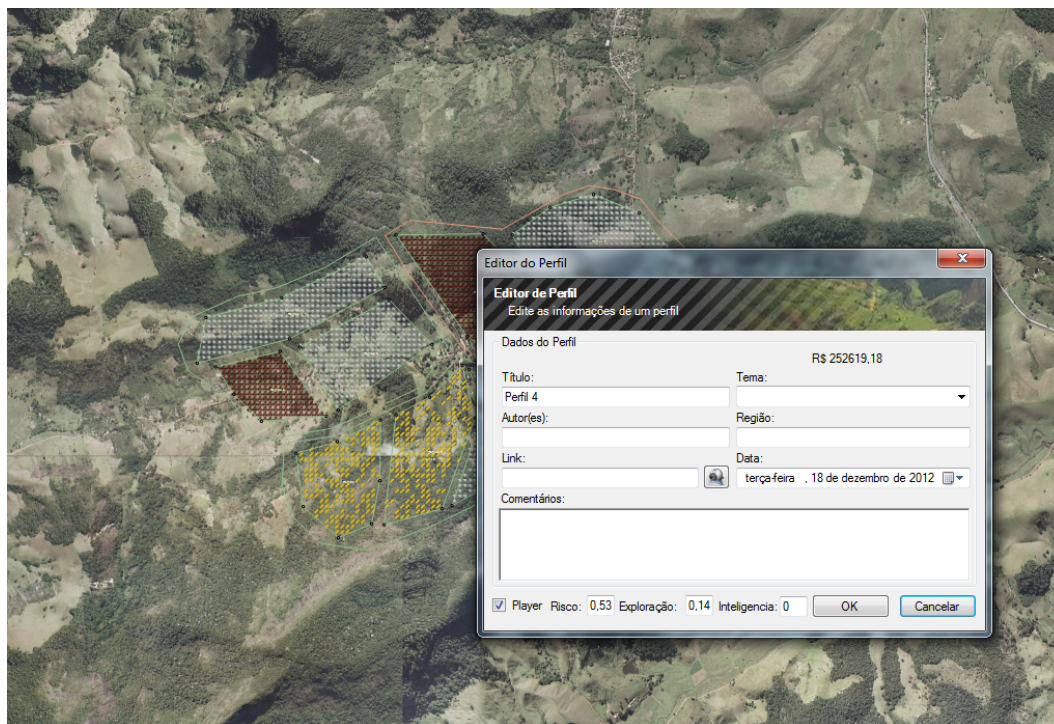


Figura 4.6: Formulário de um perfil durante um *game*

semelhante entre si como árvores de decisão previamente criadas. Árvores de decisão após criadas podem ser representadas por regras *if then* de maneira a simplificar a leitura. Árvores de decisão representam uma disjunção de conjuntos de restrições sobre os valores dos atributos de instâncias. Cada caminho da raiz da árvore a uma folha corresponde a um conjunto de testes de atributos, e a árvore é uma disjunção desses conjuntos, Mitchell (9).

Cada estratégia é dividida em duas decisões principais: venda e produção. Para a venda, o agente decide o que vender e em que quantidade. Para a produção, o agente decide o que produzir em cada região de cultivo e se mantém ou não o que está produzindo. Os melhores e piores produtos são definidos por uma tabela que é atualizada pelos agentes pela observação do ambiente, como mostrado anteriormente. As cinco estratégias inicialmente definidas pela inteligência dos agentes, criadas sem qualquer compromisso com a realidade, são as seguintes:

1. Para venda, a primeira estratégia usa uma previsão fornecida pelo mercado e o último preço de cada produto em estoque, e faz um cálculo para ver se a previsão é suficientemente boa. Com esse valor somado ao risco do agente, é feito um mapeamento simples para saber que porcentagem (do que se tem de cada produto) irá vender. Para a produção, o fazendeiro só troca de cultivo se o ciclo produtivo do produto

atual tiver se completado, ou seja, ele não interrompe nenhuma produção no meio. Para cada microrregião, se ele estiver plantando o produto que degrada menos e que rende mais dinheiro, ele mantém; caso contrário, baseado na taxa de exploração, ele seleciona um produto aleatório (isto para cada microrregião).

2. Para a venda, a segunda estratégia é semelhante à primeira, porém usa uma previsão e a média de preço fornecida pelo mercado e os valores no mapeamento também diferem. Para a produção, ele também mantém caso o produto seja o melhor e troca baseado na exploração. Mas também troca, mesmo que o ciclo produtivo não esteja completo, caso haja um produto que degrade muito menos o solo ou um produto que renda muito mais dinheiro.
3. Para a venda, ao invés de usar o último preço, como na primeira estratégia, é feita uma transformação linear usando as variáveis de, previsão, risco e último preço, para saber que quantidade de produto vender. Para a produção, esta inteligência se preocupa com a perda de solo e sempre troca o cultivo caso esteja degradando muito. Também há uma pequena chance de trocar caso esteja com o produto menos degradante, baseado na taxa de exploração.
4. Para a venda, a estratégia é a mesma da terceira; porém, agora usando a média como na segunda estratégia. Para a produção, usamos a taxa de risco para saber se trocamos para o melhor produto caso esse seja muito melhor, tentemos escolher o produto que renda mais dinheiro ou escolhamos um produto aleatório.
5. Para a venda, são usados o último preço, a média e uma previsão para fazermos uma análise se o mercado está tendendo a melhorar ou piorar; e isto é usado para vender tudo, vender metade ou não vender cada produto. Para a produção, é apenas usada a taxa de exploração para saber se muda ou não de cultivo caso a produção atual não seja jovem demais, ou seja, caso o ciclo produtivo não esteja no início.

Foram adicionadas cinco configurações de testes para alguns produtos, de forma que para um melhor funcionamento do programa deveria haver um estudo e com a ajuda de profissionais da área definir quais seriam as melhores curvas de produção e configuração de mercado para cada produto em particular, afim de gerar uma simulação mais próxima da vida dos agricultores que usarão o programa. Todos os outros produtos que estão na base do

Microbacias podem ser utilizados no *game*, porém sempre usarão uma curva de produção constante e valores de mercado default.

A tabela 4.1 mostra as configurações de cada produto para o mercado, onde #Farmers representa o número de agentes no *game*:

	Preço Inicial	Alfa	Beta	Ômega
LEITE	20	2600 * #Farmers	10000 * #Farmers	0.1
CAFE	100	45 * #Farmers	100 * #Farmers	0.1
MILHO	10	5000 * #Farmers	100000 * #Farmers	0.1
SAF	0	0	0	0
MATA	0	0	0	0
Default	50	#Farmers	#Farmers	0.1

Tabela 4.1: Variáveis configuráveis de mercado

As curvas de produção mostradas nas figuras 4.7, 4.8, 4.9, 4.10, 4.11 e 4.12 representam a razão da produção calculada a ser produzida em uma dada rodada.

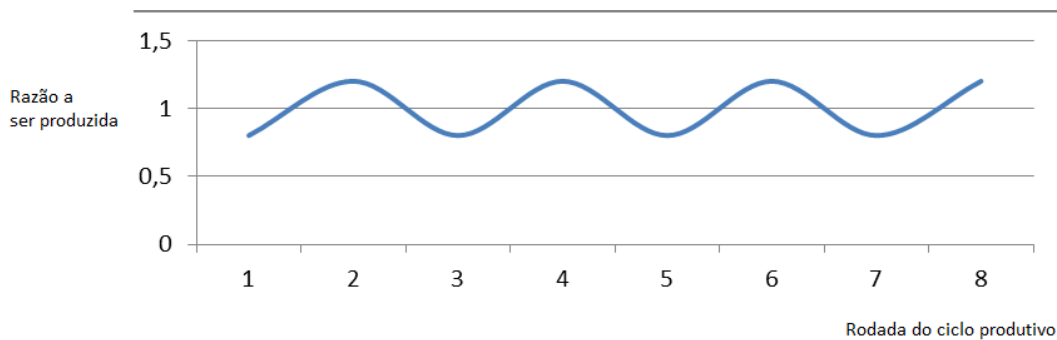


Figura 4.7: Curva do Produto leite

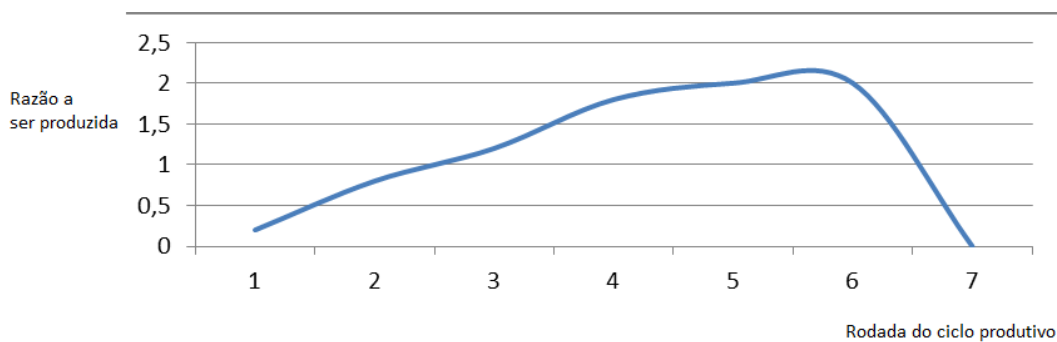


Figura 4.8: Curva do Produto café

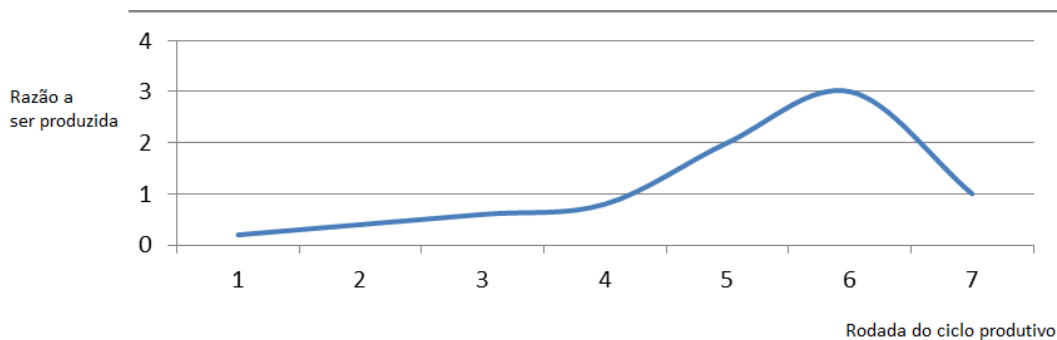


Figura 4.9: Curva do Produto milho

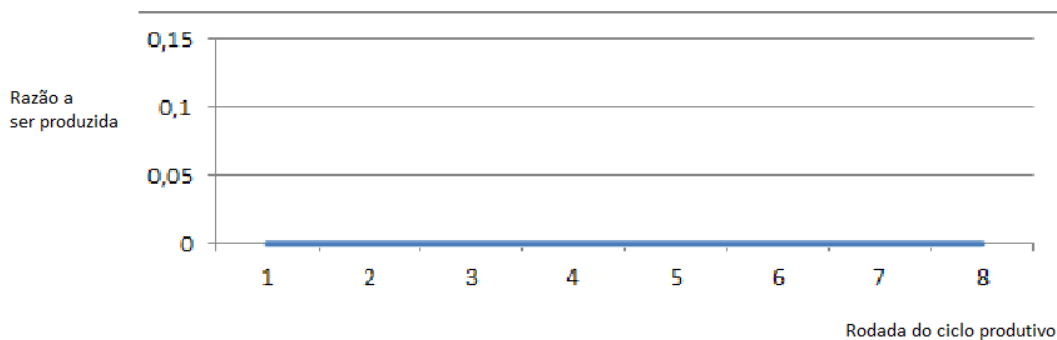


Figura 4.10: Curva do Produto SAF (Sistemas de Agro-reflorestamento)

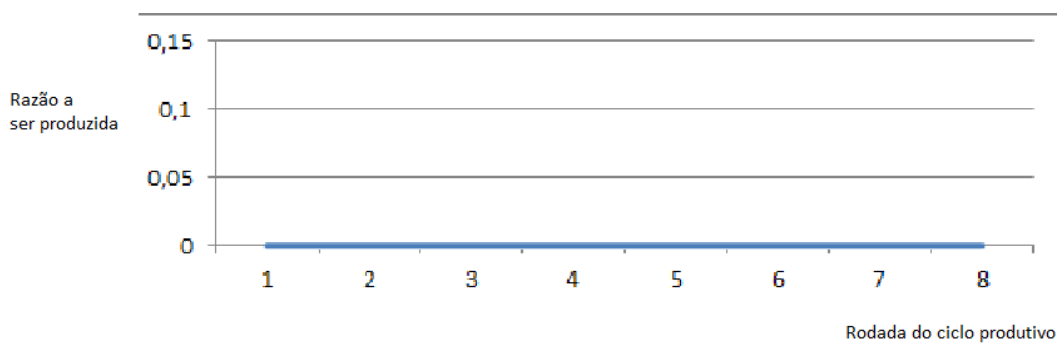


Figura 4.11: Curva do Produto mata nativa

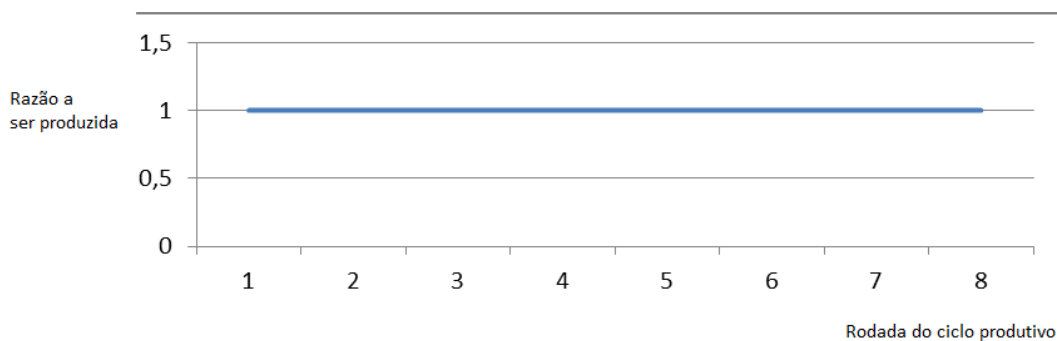


Figura 4.12: Curva default para qualquer outro produto não citado

Os fatores alfa, beta e ômega são usados para o cálculo do novo preço

por parte do mercado.

Além destas configurações, ainda existem as configurações de cada produto dentro do simulador que calcula os valores de produção e erosão. Mas devido a transparência do programa, estas tabelas e configurações já estão consolidadas, sendo relevante apenas os resultados que elas produzem para o nosso framework.

4.3

Log de Eventos

Produzimos um gerador de Log de eventos para ser utilizado futuramente com algum tipo de análise pelos jogadores, humanos ou artificiais. O log também serviu para validar os testes executados. Foi gerado um log de exemplo que representa um jogo com três rodadas com um jogador controlado pelo usuário que começa com os cultivos, café, leite e milho. Depois muda o cultivo de café pra leite na segunda rodada, e, em seguida, muda o cultivo de milho pra café na terceira rodada e vende cem unidades de leite na terceira rodada.

```
Perfil1 Int-\>0.3867100483676 Risk-\>0.277863141045842 Explor-\>0.716232400255386
```

ROUND :1

fazendeiro Perfil1 vendeu 0 de LEITE

fazendeiro Perfil1 vendeu 0 de CAFE

fazendeiro Perfil1 vendeu 0 de MILHO

0 total de vendas de LEITE foi de 0

o preco do produto LEITE foi alterado de 20 para 20.307148903132

0 total de vendas de CAFE foi de 0

o preco do produto CAFE foi alterado de 100 para 100.495129596021

0 total de vendas de MILHO foi de 0

o preco do produto MILHO foi alterado de 10 para 10.04205427833

ROUND :2

fazendeiro Perfil1 vendeu 0 de LEITE

fazendeiro Perfil1 vendeu 0 de CAFE

fazendeiro Perfil1 vendeu 0 de MILHO

fazendeiro Perfil1 mudou o cultivo de CAFE para LEITE
0 total de vendas de LEITE foi de 0
o preco do produto LEITE foi alterado de 20.307 para 20.5291650029962
0 total de vendas de CAFE foi de 0
o preco do produto CAFE foi alterado de 100.495 para 100.954351971727
0 total de vendas de MILHO foi de 0
o preco do produto MILHO foi alterado de 10.042 para 10.0785544581956

ROUND :3

fazendeiro Perfil1 vendeu 100 de LEITE
fazendeiro Perfil1 vendeu 0 de LEITE
fazendeiro Perfil1 vendeu 0 de MILHO
fazendeiro Perfil1 mudou o cultivo de MILHO para CAFE
0 total de vendas de LEITE foi de 100
o preco do produto LEITE foi alterado de 20.529 para 20.8125328412166
0 total de vendas de CAFE foi de 0
o preco do produto CAFE foi alterado de 100.954 para 101.426956468385
0 total de vendas de MILHO foi de 0
o preco do produto MILHO foi alterado de 10.079 para 10.1448569453125

É importante notar que os valores Int (Inteligência), Risk (Risco) e Explor (Exploração) representam as configurações iniciais do fazendeiro, que no caso de um jogador humano não tem nenhuma função. O log é bem claro no seu objetivo e mostra claramente as ações tomadas. A alteração no preço diz respeito a cálculos do mercado que envolve valores aleatórios; portanto, podem não ser os mesmos de execução para execução.

O log foi modelado de maneira ortogonal, de tal forma que qualquer informação presente no jogo possa ser cadastrada.

4.4

Comportamento de mercado

A avaliação do mercado consiste em avaliar se o sistema de vendas do mercado funciona realmente e se é possível ter controle através de configurações programadas previamente. São estas configurações que provam que qualquer produto pode ser adicionado ao framework sem que haja uma desestabilização

do mercado, como uma grande subida de preços de um produto deixando os outros não competitivos.

Neste caso devemos testar alguns valores de alfa, beta e ômega para algum produto e ver se eles produzem a saída esperada. O número de fazendeiros foi fixado em 20 e foram gerados jogos com apenas jogadores controlados por IA com 200 rodadas.

Os gráficos a seguir ilustram o efeito da variação do fator alfa na curva de preços do produto milho (Figuras 4.13, 4.14, 4.15, 4.16 e 4.17).

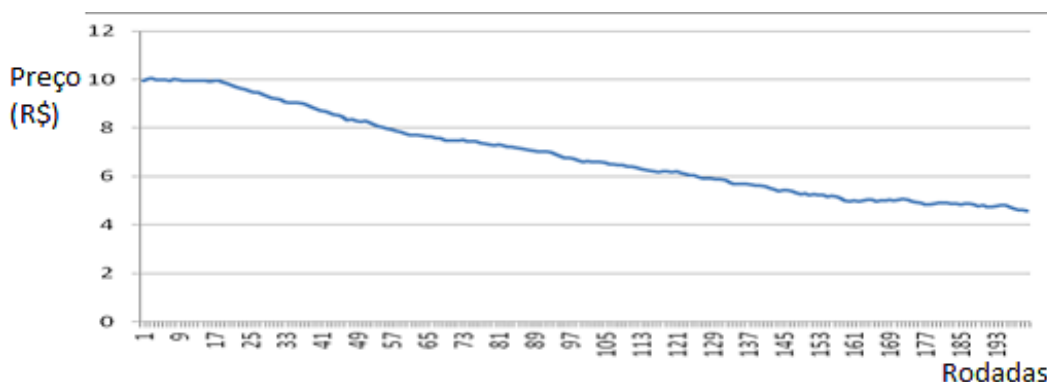


Figura 4.13: Alfa igual a $500 \cdot \text{NúmeroDeFazendeiros}$

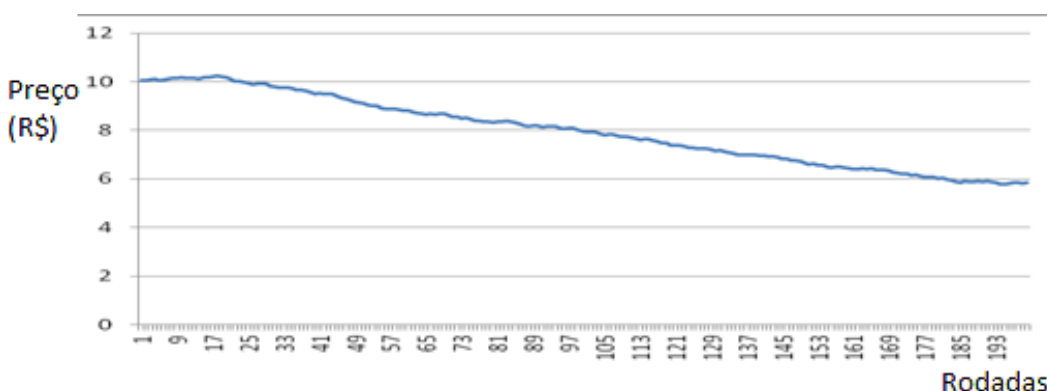


Figura 4.14: Alfa igual a $2000 \cdot \text{NúmeroDeFazendeiros}$

Podemos ver claramente que quanto maior a demanda a ser suprida mais crescente se torna a curva de preço, chegando ao ponto em torno de 5000, que equivale a um ponto de equilíbrio.

Podemos agora observar a variação do valor de beta, que é o valor que transforma a unidade de vendas do produto em unidade de preços. Os gráficos do produto café a seguir ilustram isso (Figuras 4.18, 4.19 e 4.20).

Podemos ver, pela escala dos gráficos, a variação obtida mudando o valor de beta; e vale notar que a estrutura da curva não muda. Quanto menor o



Figura 4.15: Alfa igual a $5000 \times \text{NúmeroDeFazendeiros}$

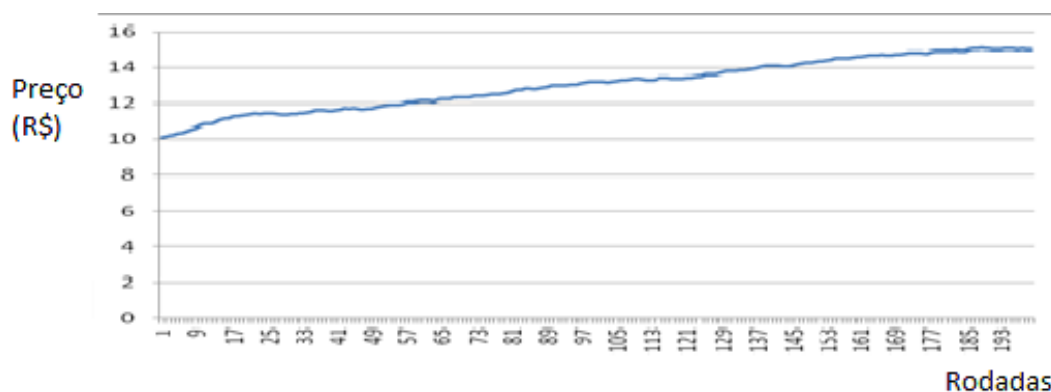


Figura 4.16: Alfa igual a $7500 \times \text{NúmeroDeFazendeiros}$

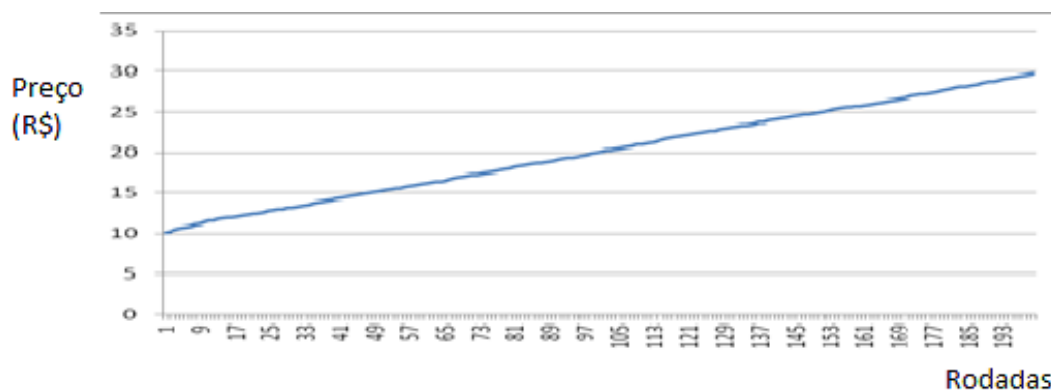


Figura 4.17: Alfa igual a $15000 \times \text{NúmeroDeFazendeiros}$

valor do beta maior a variação obtida, já que as vendas estão em uma escala muito maior que o preço.

Por último, foi avaliado o ômega que pondera o quanto o ruído influenciará no resultado, isto é, quanto maior ômega mais aleatória será a curva, que podemos observar nos gráficos (Figuras 4.21, 4.22 e 4.23).

Tendo o controle destes três fatores, podemos estipular uma configuração para os produtos que produza o resultado desejado. Vale lembrar que a

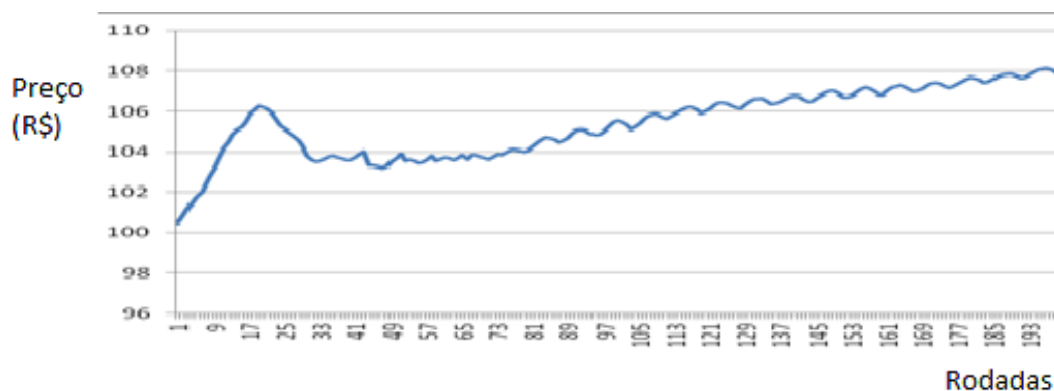


Figura 4.18: Beta igual a 5000*NúmeroDeFazendeiros

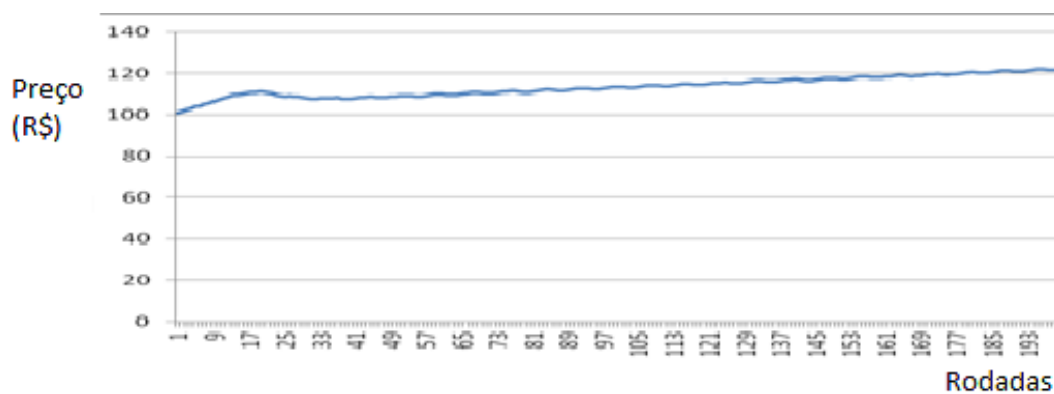


Figura 4.19: Beta igual a 7500*NúmeroDeFazendeiros

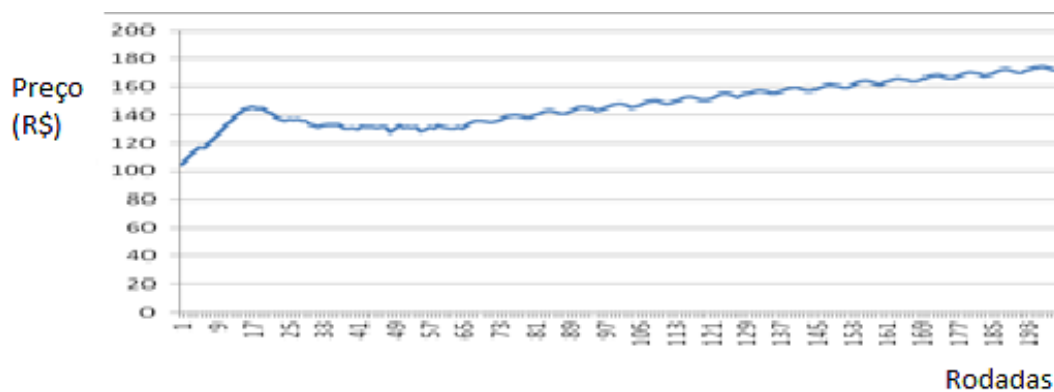
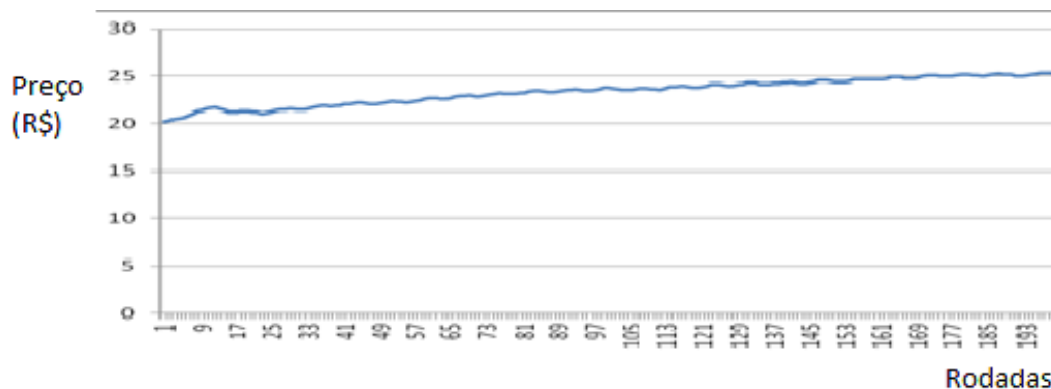
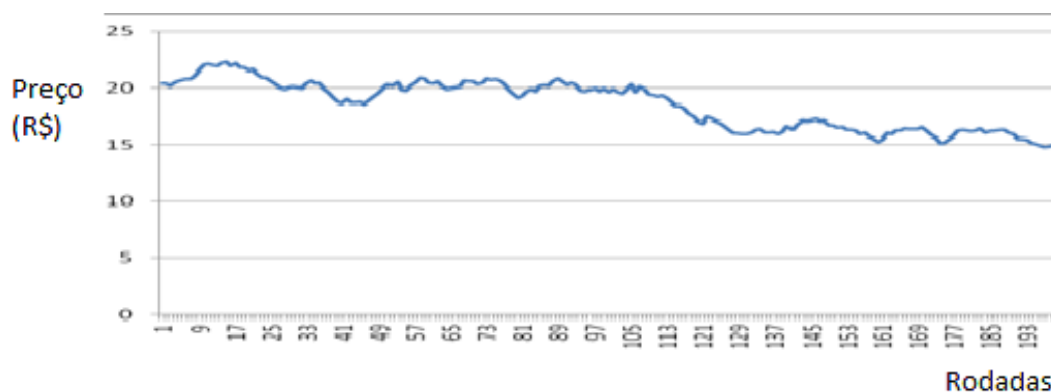
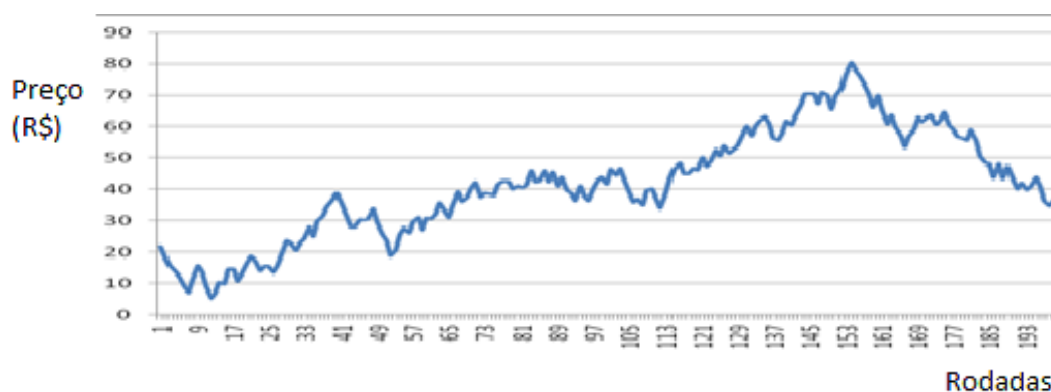


Figura 4.20: Beta igual a 15000*NúmeroDeFazendeiros

estratégia dos fazendeiros também influencia no mercado, mas estas já não fazem mais parte do framework; e, inclusive, podemos ajustar as configurações do mercado se baseando nelas.

Figura 4.21: $\hat{\Omega}$ igual a 0.1Figura 4.22: $\hat{\Omega}$ igual a 1Figura 4.23: $\hat{\Omega}$ igual a 10

4.5

Avaliação geral

Por fim, uma avaliação geral foi feita para verificar se os principais requisitos do projeto foram satisfeitos.

- Integração com o Microbacias - O bom funcionamento do framework depende diretamente deste requisito. Como foi mostrado, o framework consegue ter acesso aos dados lidos e calculados pelo simulador que

é o programa principal. Logo este requisito foi satisfeito; a própria modelagem mostrou isso.

- Flexibilidade - Apesar de não ter sido feita uma análise de quão flexível é o framework para diversas situações, os resultados obtidos revelaram facilidades na definição do jogo. Tivemos, por exemplo, um controle razoável do mercado, o que torna possível inferir que todos, senão boa parte de produtos com as mais diversas configurações podem ser adicionados ao simulador e que podemos obter um jogo balanceado. Por outro lado, a parte de estratégias dos fazendeiros provê diversos mecanismos para implementar as mais diversas estratégias possíveis, mostrando assim um framework flexível ao seu propósito.
- Independência parcial - Poucas alterações no Microbacias foram necessárias, basicamente adicionais na parte da interface para mostrar alguns dados do *game*. Tendo em vista que apenas foram adicionados dados que no estágio de exibição da interface não tem correlação uns com os outros e não se atrapalham. Nenhuma mudança significativa foi feita no Microbacias de forma que alterasse sua estrutura ou seu funcionamento comum.
- Desempenho - O desempenho do Microbacias foi afetado em dois pontos apenas. Na inicialização de cada jogo e entre cada rodada de um jogo. Em um jogo com uma média de 6 jogadores gastamos aproximadamente 40 segundos para inicializar e entre cada rodada aproximadamente 2 segundos. este é um tempo razoável visto que o grande gargalo são os cálculos feitos pelo simulador para calcular a perda de solo que consomem bastante tempo. Os cálculos entre cada rodada estão otimizados para recalculer a perda de solo apenas para as fazendas onde houver modificações.
- Facilidade de uso - O uso do framework se dá por meio de alterações e inclusões no próprio código. O balanceamento do *game* é relativamente difícil de se controlar quando colocamos vários dados ao mesmo tempo. Fazer alterações é fácil para uma pessoa com nível básico de programação, mas tornar essas alterações um *game* balanceado é a parte realmente complicada.