

# *Super Asderek World*



*He will do what is necessary, to protect his people*

**A game by**

**Lucas P Nepomuceno**

## **Mecânica Do jogo**

O jogo é programado na linguagem C#

### **Plataforma**

PC

### **Código externo**

Super Asderek World utiliza-se de códigos externos na geração de colliders custom.

### **Código e dados de objetos**

Super Asderek World possui diversos elementos, estruturados da forma: Gerenciadores de Conteúdo (GameManager, UIManager, WeaponsManager etc...), Personagens (Jogador, Inimigos), Objetos Interativos (Baús, itens, NPCs) etc.

### **Controle de loop**

O jogo mantém-se em loop automático conforme a programação padrão da plataforma Unity. Os personagens têm seu comportamento gerido por seus scripts que indicam o que devem fazer a cada interação (update).

### **Dataflow**

Toda informação a ser exibida passa pelo gerenciador de UI, qualquer informação do jogo é enviada para o GameManager, informações de progresso são tratadas pelo ProgressManager. Personagens interagem entre si para tratamento de colisão e agressão.

O armazenamento das informações do jogo são feitas em um arquivo JSON armazenado e lido automaticamente pelo jogo

### **Física e estatísticas**

O jogo possui o conceito de gravidade e decaimento de movimento para uma movimentação mais fluida. O jogador acelera uniformemente ao pressionar o botão de movimento e desacelera quando solta.

Vida Magia e Poder são representados em porcentagem por uma barra que enche ou decai conforme a utilização. Pontos de Habilidade são representados por números no menu de habilidades, assim como quantidade de itens.

### **AI/AS**

Super Asderek World não possui uma AI propriamente dita. Os inimigos reagem de acordo com sua programação pré-processada.

## Interface jogador

### Telas principais e secundárias

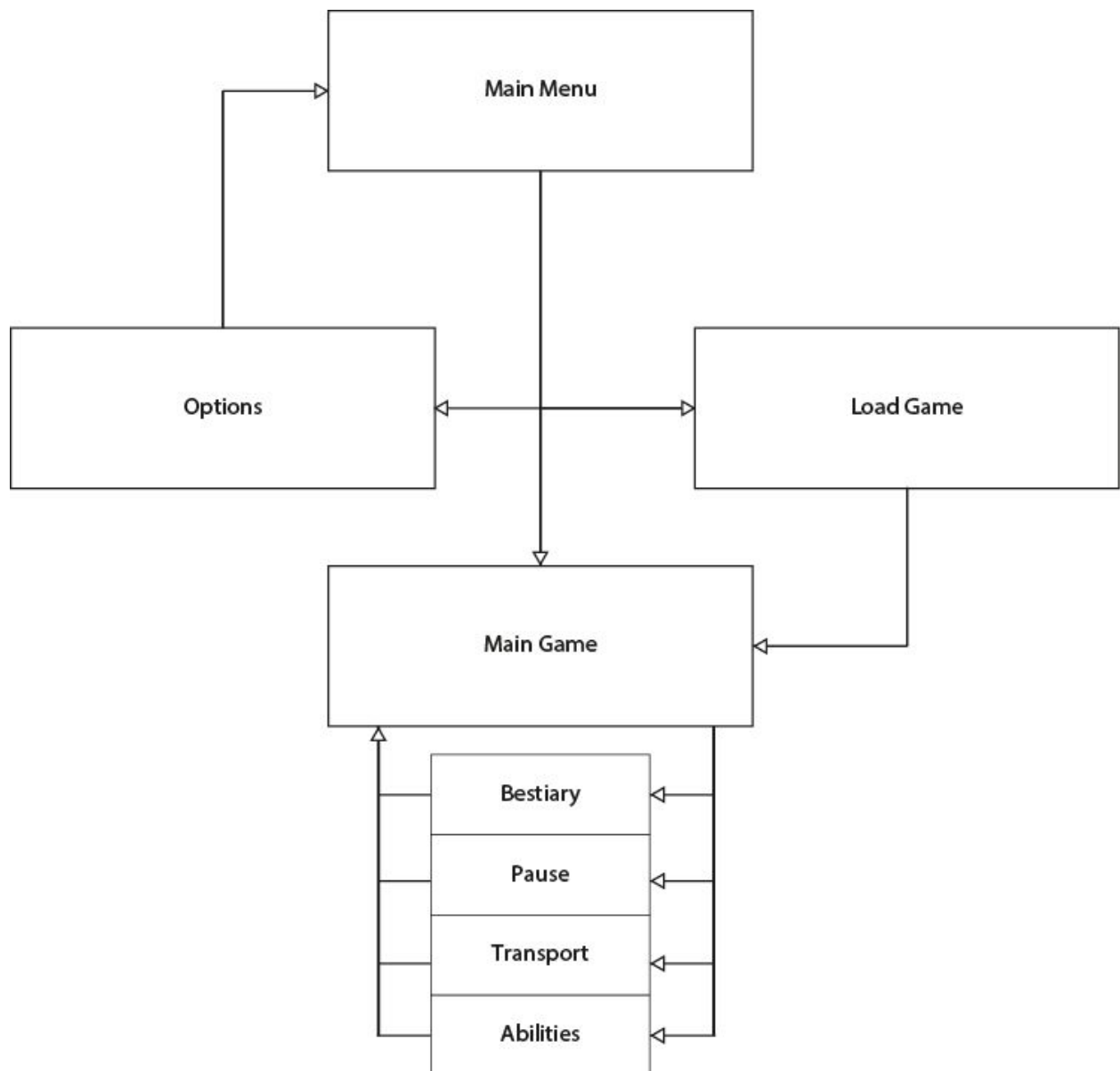
O jogo possui 5 telas iniciais:

Main Menu, Load Game, Options.

E 6 telas dentro do jogo:

Main Game, Pause, Abilities, Bestiary, Transport

### Nós Decisórios



## AUDIOVISUAL:

### Audio

**ENGENHARIA DO SOM:** O som será feito com arquivos de mídia digital .wav, .midi e .mp3

**GATILHO:** Efeitos dentro do jogo reproduzirão o arquivo desejado no momento apropriado.

### Visual

#### ENGINE GRÁFICA E EFEITOS

A engine utilizada é o Unity. Todos os sprites foram criados pela equipe de produção. A animação se dá pela troca sequencial de sprites, gerenciada pela função animator da engine.

#### GATILHOS

Os gatilhos são tratados pelo código de cada artefato, e gerenciados pela própria engine. Os scripts cuidam da resolução e preparo da próxima transição entre animações.

## **CÓDIGO ESPECÍFICO DE NÍVEL/FASE:**

### **Refinamento**

Retoques no balanceamento e game feel do jogo foram deixadas para o final quando um mínimo produto viável for alcançado

### **Teste**

Após o desenvolvimento da parte funcional de cada objeto, e a parte gráfica implementada, ela é testada de forma que o game feel esteja coerente com um jogo moderno. Retoques na arte e no código são implementados conforme a necessidade observada

## **ERROS COMUNS:**

### **Falhas de Comunicação**

Desacordos entre o que deve ser implementado ocorrem com certa frequência, mas encontra-se uma resolução para que o desenvolvimento não cesse

### **Pausas Frequentes no Desenvolvimento**

Por ser um projeto pessoal de fim de semana, ocorrem ocasiões em que os desenvolvedores passam meses sem trabalhar no jogo, o que causa problemas para a continuação do desenvolvimento, tendo em vista que os desenvolvedores necessitam se habituar novamente com o estágio atual de desenvolvimento.