GameState

```
+MENU
+PLAYING
```

sdlManager

```
-SDL_Renderer* gRenderer;
```

```
+TextureManager(SDL_Renderer* renderer) ;
+SDL_Texture* loadTexture(const std::string& path) ;
```

GAME

```
-const int SCREEN_WIDTH = 900;
-const int SCREEN_HEIGHT = 650;
-vector<SDL_Rect> P;
-std::array <Vegetables*, 9> v{};
-vector <Animals*> a;
-Startscreen* startScreen;
-GameState game;
-SDL_Window* gWindow = NULL;
-SDL_Renderer* gRenderer = NULL;
-SDL_Window* marketWindow = NULL;
-SDL_Window* storageWindow= NULL;
-bool game_over = false;
```

```
+Game():game(GameState::MENU){};
+bool Game_logic(SDL_Event &e, +bool quit);
+bool init();
+void close();
+void run();
+bool isClickedOnBarn(int clickX, int clickY);
+bool isClickedOnMarket(int clickX, int clickY);
+SDL_Texture* loadTexture(const std::string& filePath, SDL_Renderer* renderer);
+void renderTexture(SDL_Texture* texture, SDL_Renderer* renderer);
+void createStorageWindow(SDL_Window* storageWindow, Storage *storage);
+void createMarketWindow(SDL_Window* marketWindow, SDL_Renderer* Renderer, LandPatches
*land, Storage *storage, Marketplace *market);
};
```

Startscreen

```
-SDL_Renderer* mRenderer;
-SDL_Texture* mTexture;
```

```
+Startscreen(SDL_Renderer* renderer);
+~Startscreen();
+bool loadMedia();
+void handleEvent(SDL_Event& e, GameState& gamestate);
+bool loadStartscreenImage();
+void render();
```

GameOver

```
-SDL_Texture* GTexture = NULL;
```

```
+GameOver(SDL_Renderer* renderer);
+~GameOver();
+bool loadMedia();
+void render(SDL_Renderer* gRenderer);
```

Wheat

```
-int current_stage;
```

```
+Wheat(SDL_Renderer* gRenderer, int x);
+~Wheat();
+uint32_t timeElapsed();
+void Grow();
+char Type();
+int Harvest();
+bool isHarvestable();
```

Vegetables

```
-uint32_t creation_time;
-uint32_t current_time;
-static int carrot_price, radish_price, wheat_price;
-int growth_time;
-int totalHarvest;
-bool harvestStatus;
-SDL_Renderer* Renderer = NULL;
-SDL_Texture* Texture = NULL;
-SDL_Rect Rect;
```

```
+Vegetables();
+virtual ~Vegetables();
+bool loadMedia();
+void VegetableRenderer();
+int Sell(char type);
+virtual void Grow(){}
+virtual char Type() = 0;
+virtual int Harvest() = 0;
+virtual bool isHarvestable() = 0;
+void setVegetables(int time, int harvestItems, bool
harvest);
+SDL_Rect getRect();
+virtual uint32_t timeElapsed()=0;
```

Radish

```
-int current_stage;
```

```
+Radish(SDL_Renderer* gRenderer, int x);
+~Radish();
+uint32_t timeElapsed();
+void Grow();
+char Type();
+int Harvest();
+bool isHarvestable();
```

Carrot

```
-int current_stage;
```

```
+Carrot(SDL_Renderer* gRenderer, int x);
+~Carrot();
+uint32_t timeElapsed();
+void Grow();
+char Type();
+int Harvest();
+bool isHarvestable();
```

Farm

```
-SDL_Texture* farmTexture = NULL;
-SDL_Texture* fenceTexture = NULL;
-SDL_Rect fenceRect = {470, 300, 310, 320};
-SDL_Rect fenceRect2 = {137, 69, 250, 150};
```

```
+Farm(SDL_Renderer* gRenderer);
+~Farm();
+void farmRender(SDL_Renderer* gRenderer);
+bool loadMedia();
+void getAnimals(SDL_Renderer* gRenderer, vector <Animals*> &A);
+SDL_Rect getFence(char type);
```

Farmer

```
-const int farmer_ht = 40;
-const int farmer_wdt = 35;
-int farmer_x = 40;
-int farmer_y = 0;
-SDL_Texture* farmerTexture = NULL;
-SDL_Rect farmerRect = { 40, 0, 35, 40 };
```

```
+Farmer(SDL_Renderer* gRenderer);
+~Farmer();
+void farmerRender(SDL_Renderer* gRenderer);
+bool loadMedia();
+void movements(char x);
+void out_movement(int sw, int sh);
+SDL_Rect Get_SDlRECT();
+bool checkCollision( SDL_Rect b );
```

Marketplace

```
-SDL_Texture* marketTexture = NULL;
-const int market_ht = 128;
-const int market_wdt = 100;
-const int market_x = 562;
-const int market_y = 86;
-SDL_Rect marketRect = {market_x, market_y, market_wdt, market_ht};
```

```
+Marketplace(SDL_Renderer* gRenderer);
+~Marketplace();
+void marketRender(SDL_Renderer* gRenderer);
+bool loadMedia();
+SDL_Rect getSDLRect();
+ sellItems<T>(store: Storage*, item: T*, type: char): void
+ Purchase<T>(store: Storage*, item: T*, type: char): bool
```

Land_Patches

```
-int locked_patches;
-int open_patches;
-static const int num_patches;
-static const int item_price;
-vector<SDL_Rect> AllPatches;
-const int land_ht = 200;
-const int land_wdt = 200;
-const int land_x = 166;
-const int land_y = 413;
-SDL_Rect patch1 = {176,423,56,56};
-int patch1_x = 176;
-SDL_Rect patch2 = {236,423,56,56};
-int patch2_x = 236;
-SDL_Rect patch3 = {296,423,56,56};
-int patch3_x = 296;
-SDL_Rect patch4 = {176,483,56,56};
-int patch4_x = 176;
-SDL_Rect patch5 = {236,483,56,56};
-int patch5_x = 236;
-SDL_Rect patch6 = {296,483,56,56};
-int patch6_x = 296;
-SDL_Rect patch7 = {176,543,56,56};
-int patch7_x = 176;
-SDL_Rect patch8 = {236,543,56,56};
-int patch8_x = 236;
-SDL_Rect patch9 = {296,543,56,56};
-int patch9_x = 296;
-bool p1 = false;
-bool p2 = false;
-bool p3 = false;
-bool p4 = false;
-bool p5 = false;
-bool p6 = false;
-bool p7 = false;
-bool p8 = false;
-bool p9 = false;
-std::array<bool, 9> plantStatus;
-SDL_Texture* landTexture = NULL;
-SDL_Texture* patchTexture = NULL;
```

```
+LandPatches();
+LandPatches(SDL_Renderer* gRenderer); //make default values 0
+~LandPatches();
+void landRender(SDL_Renderer* gRenderer);
+void patchRender(SDL_Renderer* gRenderer);
+void createVeg(SDL_Renderer* gRenderer, std::array <Vegetables*, 9> &v, SDL_Rect rect );
+bool loadMedia();
+void getPatches(vector<SDL_Rect> &p);
+bool isPlanted(SDL_Rect rect);
+Vegetables* returnVeg(std::array <Vegetables*, 9> &v, SDL_Rect rect);
+void setEmpty(std::array <Vegetables*, 9> &v, SDL_Rect rect);
+void UnlockPatch();
+int itemPrice(char type);
```

Storage

```
-const int barn_ht = 180;
-const int barn_wdt = 150;
-const int barn_x = 390;
-const int barn_y = 32;
-int WIDTH = 600;
-int HEIGHT = 450;
-SDL_Window* StorageWindow = NULL;
-SDL_Renderer* StorageRenderer = NULL;
-SDL_Texture* StorageWindowTexture = NULL;
-SDL_Rect storageRect = { barn_x, barn_y, barn_wdt, barn_ht };
-SDL_Rect StoreRect = {330, 45, 500, 400};
-SDL_Texture* storageTexture = NULL;
-SDL_Texture* trees1Texture = NULL;
-SDL_Texture* trees2Texture = NULL;
-SDL_Rect leftTrees = {-60, 0, 200, 680};
-SDL_Rect rightTrees = {760, -25, 200, 680};
-static int GoldCoins;
-static int milk;
-static int eggs;
-static int carrots;
-static int wheat;
-static int radish;
-static int seeds;
```

```
+Storage(SDL_Renderer* gRenderer);
+Storage();
+~Storage();
+void storageRender(SDL_Renderer* gRenderer);
+bool loadMedia();
+SDL_Rect getSDLRect();
+void addMaterial(char Material, int qty);
+void useMaterial(char Material);
+int getAmount(char Material);
```

Cow

```
-bool current_produce;
-bool hunger;
-int produce_amount;
-int y_speed = 2;
-SDL_Rect cowDown = {150, 362, 57, 81};
-SDL_Rect cowUp = {150, 100, 57, 86};
```

```
+Cow(SDL_Renderer* gRenderer);
+~Cow();
+void walk();
+char Type();
+char feedType();
+bool feed();
+bool produce();
+uint32_t timeElapsed();
+int collectProduce();
+void setProduce();
+void setHunger();
+bool animalSick();
+void setAlert();
```

Animals

```
#uint32_t time_to_feed;
#uint32_t feeding_time;
#uint32_t produce_time;
#uint32_t current_time;
#static int egg_price;
#static int milk_price;
#static int cow_price;
#static int chicken_price;
#bool alert;
#SDL_Renderer* Renderer = NULL;
#SDL_Texture* Texture = NULL;
#SDL_Texture* alertTexture = NULL;
#SDL_Rect alertRect;
#SDL_Rect Rect;
#SDL_Rect moverRect;
```

```
+uint32_t creation_time;
+Animals();
+~Animals();
+bool loadMedia();
+void AnimalRenderer();
+int Sell(char type);
+int itemPrice(char type);
+virtual uint32_t timeElapsed() = 0;
+virtual void walk() = 0;
+virtual char Type() = 0;
+virtual char feedType() = 0;
+virtual bool feed() = 0;
+virtual bool produce() = 0;
+virtual void setProduce() = 0;
+virtual int collectProduce() = 0;
+virtual void setHunger() = 0;
+virtual bool animalSick() = 0;
+virtual void setAlert() = 0;
```

Chicken

```
-bool current_produce;
-bool hunger;
-int produce_amount;
-int y_speed = 2;
-SDL_Rect chickenDown = {243, 395, 42, 48};
-SDL_Rect chickenUp = {243, 141, 41, 45};
```

```
+Chicken(SDL_Renderer* gRenderer);
+~Chicken();
+uint32_t timeElapsed();
+void walk();
+char Type();
+char feedType();
+bool feed();
+bool produce();
+void setProduce();
+int collectProduce();
+void setHunger();
+bool animalSick();
+void setAlert();
```