# Craps Classes

## To Do

Errors and Messaging

Transfer of line bets to bets/odds Odds need to be by roll number in the case of Field Bets 2 and 12 pays double. This can be done by creating a vector of new bets called assigns. Assigns will be a structure {amount, number, BetType}.

Before this can be completed need to get the BetList figured out. Each assignment will be a new bet in the list of bets

Deal with odds in bets, oddD , oddN. Denominator and Numerator. Need to think about number, as stated above, for field bets 2 and 12 pay double or maybe triple. This is only the case with field bets so that can be added to just the field bet odds maybe.

Deal with BetList

Deal with reporting payments and losses in bets

Read input file for game rules How to bet how many bets

## Bank

This is all to do with the amount of money in the Bank Roll

Parent: None

Private:

Variables:

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| Int | Amount | Amount of money in the Bank Roll |
|  |  |  |
|  |  |  |

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Virtual | Type | Name | Input | Description |
|  | Int | GetAmount | BetType | Return the total in the Bank |
|  |  | Bank | Amount | Constructor Set Amount |
|  |  |  |  |  |

## Bet

This is an individual bet. The bet will have a BetType assigned to the bet. There will also be an amount associated with the bet.

Parent:

Private:

Variables:

BetType type Type of bet object i.e. PassLine

int amount Money bet on this bet

int betTypeId Id of the type of bet this variable contains

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Void | Bet | Int amount | Amount of the bet |
|  |  |  | BetType | Type of bet this will be |
| Virtual | Int | CheckLoser | Int roll | Check bet to see if it is a loser |
| Virtual | Int | CheckChange | Int roll | Check bet to see if it needs to be changed to another BetType |
|  | Int | GetAmount | Void | Return the amount of the bet |

## BetList

Parent:

Private:

Variables:

Std::list <Bet> bets list of bets currently on the table

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Void | BetList |  | Constructor |
|  | Int | GetAmount | BetTypeSt | Get the amount of bets for a BetType |
|  | Int | GetAmount | Void | Get the total of all bets on the table |
|  |  |  |  |  |
|  |  |  |  |  |

BetList Constructor

int CheckBet Loop thru the list of bets and check for results

## BetType

Parent: None

Private:

Variables:

Methods:

Protected:

Variables:

|  |  |  |
| --- | --- | --- |
| vector<int> | winner | Array of winning rolls for this object |
| Vector<int> | loser | Array of losing rolls for this object |
| Vector<int> | change | Array of rolls that need to be changed into new bets |
| Int | oddsD | Denominator for the odds for BetType |
| Int | oddsN | Numerator for the odds for the BetType |

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Virtual | Int | CheckWinner | Int roll | Check bet to see if it is a winner. |
| Virtual | Int | CheckLoser | Int roll | Check bet to see if it is a loser |
| Virtual | Int | CheckChange | Int roll | Check bet to see if it needs to be changed to another BetType |
|  |  |  |  |  |

Public:

Variables:

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | BetType | Void | Constructor |
| Virtual |  | ~BetType |  | Destructor of the object |
|  | Int | Check | Int roll | Check runs the other methods to determine what to do with the bet |
|  |  |  |  |  |

## PassLine

Description

This is the Pass Line bet. It represents a bet placed on the pass line on the come out roll. When the dice are rolled something will happen to this bet. It is either a Winner, a loser or being set to another bet, which would be a pass bet on a number 4,5,6,8,9,10.

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

Int CheckChange(int) Check bet type for a winner. Passed variable is roll number

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | BetType | Void | Constructor |
| Virtual |  | ~BetType |  | Destructor of the object |
|  | Int | Check | Int roll | Check runs the other methods to determine what to do with the bet |
|  |  |  |  |  |

## Pass

Parent: Bettype

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## ComeLine

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

Int CheckChange(int) Check bet type for a winner. Passed variable is roll number

## Come

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## Odds

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## Field

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## OneTime

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## HardWays

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## Place

Parent: BetType

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

## Dice

Parent: None

Private:

Variables:

int sides Number of sides on the Die

Methods:

int Roll Roll the die and return value

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

Dice (int s) Constructor

## Button

Description:

This is the object that mimics the button on the table. It is used to say if a number is currently selected and to denote which number is selected if it is ‘ON”

Parent: None

Private:

Constants:

Int buttonOn = 1 Value for button on

Int buttonOff = 0 Value for button off

Variables:

Int state Current state of the Button

Int number Number that the button represents {4,5,6,8,9,10}

Methods:

Int buttonState Set the state of the button based on the current state and the roll

Protected:

Variables:

Methods:

Public:

Variables:

Methods:

Button(void) Constructor

Int setState(int) Set the state of the button

Int getState(void) Get the current state of the button

Structure to copy

Description:

Parent:

Private:

Variables:

Methods:

Protected:

Variables:

Methods:

Public:

Variables:

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| Vector<int> | Loser | Array of losing rolls for this object |
| Vector<int> | Change | Array of rolls that need to be changed into new bets |
|  |  |  |

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Virtual | Type | Name | Input | Description |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Program Flow

Read SetUp file

New Game

New Shooter

Place a bet

New bets (PassLine, ComeLine)

Assoc Bets (Odds)

Change (Pass, Come)

Roll Dice

Check Bets

Losers

Winners

Report Info

Report Info

Report Info

# Current

What Is currently being worked on

## 21/11/16

## Pass line bet

Working on Winners Losers and Assignment bets

Working on the class bet type / Pass line

<https://stackoverflow.com/questions/37868245/best-way-to-call-child-method-from-parent-in-c-inheritance>

I need to call a version of winner/loser and assign on each bet type but have a place holder in the BetType class be be called from each bet type.