

# DDS (Detailed Design Specification)

## Main.py

### LoginWindow

	이름	역할,설명
attributes	mainlayout	아래의 위젯들이 표시되는 메인 레이아웃
--	gametitle	화면에 표시되는 게임제목
--	gametitlefont	"의 폰트
--	story	게임의 줄거리
--	storylabel	게임의 줄거리를 화면에 표시
--	storyfont	"의 폰트
--	self.nameinput	이름 입력칸
--	inputbutton	입력된 이름으로 로그인하기
method	__init__생성자	로그인창을 생성한다.
--	setUI	로그인창 UI를 보여준다.
--	buttonClicked	로그인창을 닫고,self.nameinput에 입력된 이름을 MainWindow의 인자로 넘겨준다.

### DefeatWindow

	이름	역할,설명
attributes	self.name	화면에 표시될 플레이어 이름
--	self.time	화면에 표시될 게임상 시간
--	self.peakofmoney	화면에 표시될 최대 보유금액
--	mainlayout	다른 레이어들의 메인레이아웃
--	recordlayout	기록관련 위젯이 위치할 레이아웃
--	retrylayout	다시하기관련 위젯이 위치할 레이아웃
--	recordfont	기록관련 위젯의 폰트
--	namelabel	화면에 표시되는 플레이어 이름
--	timelabel	화면에 표시되는 게임상 시간
--	peakofmoneylabel	화면에 표시되는 최대 보유금액
--	bankruptlabel	화면에 표시되는 파산 알림 메세지
--	bankruptfont	"의 폰트
--	retrylabel	플레이어에게 다시할 지 물어보는 메세지
--	retrybutton	다시하기 버튼
method	__init__생성자	패배창을 생성한다.

	이름	역할,설명
--	setUI	패배창 UI를 보여준다.
--	buttonClicked	패배창을 닫고 로그인화면을 띄운다.

AchievementWindow

	이름	역할,설명
attributes	self.achievement	달성한 업적의 종류
--	self.record	플레이어의 업적 달성 기록
--	self.totalrecord	다른 플레이어들의 업적 달성 기록
--	mainlayout	메인 레이아웃
--	labellayout	기타 라벨들의 레이아웃
--	recordlayout	기록관련 위젯들의 레이아웃
--	buttonlayout	계속하기 버튼의 레이아웃
--	achievementlabel	달성한 업적이 무엇인지 표시
--	achievementfont	"의 폰트
--	myrecord	현재 플레이어의 기록 표시
--	font0	"의 폰트
--	recordlabel1	순위표의 제목 표시
--	font1	"의 폰트
--	recordlabel2	순위표의 범주 표시
--	self.records	업적 달성 순위표
--	continuebutton	계속하기 버튼
method	__init__생성자	업적창을 생성한다.
--	setUI	업적창 UI를 보여준다.
--	showRank	인자로 전달 받은 기록을 순위표에 한줄 씩 보여준다.
--	buttonClicked	업적창을 닫는다.

MainWindow

	이름	역할,설명
attributes	self.totaldata	파일에 저장된 이 게임에 모든 사용자 정보
--	self.data	데이터 파일에 저장될 데이터의 모음. 플레이어 이름,게임상 시간, 최대보유금액, 현재보유금액, 남은 빚, 가계부기록, 비트코인정보, 훌쩍게임정보, 구매한 물감,업적달성현황이 사전형식으로 저장된다.
--	self.playername	플레이어 이름
--	self.debt	남은 빚
--	self.money	현재 보유 금액

	이름	역할,설명
--	self.time	게임 상 시간. [주,일,시간,분]으로 돼있다.
--	self.peakofmoney	최대 보유금액
--	self.current_brush_color	현재 보유 물감(색상)
--	self.already108	1억(10**8)원 달성 여부
--	self.already1010	100억(10**10)원 달성 여부
--	self.already1012	1조(10**12)원 달성 여부
--	self.already1016	1경(10**16)원 달성 여부
--	self.already1020	1해(10**20)원 달성 여부
--	self.alreadyac	모든 물감 구입 여부
--	mainlayout	메인레이아웃
--	statuslayout	상태창의 레이아웃
--	tablayout	그림판/상점/돈벌기 창을 전환할 수 있는 탭의 레이아웃.
--	historylayout	가계부의 레이아웃
--	self.paintertab	그림판 탭
--	self.storetab	상점 탭
--	self.getmoneytab	비트코인/홀짝게임 탭
--	self.namelabel	상태창에 표시되는 플레이어 이름
--	self.moneylabel	상태창에 표시되는 현재 보유금액
--	self.timelabel	상태창에 표시되는 게임상시간
--	self.historylabel	가계부 제목과 현재 남은 빚을 표시
--	self.history	가계부
method	__init__생성자	메인 화면을 생성한다.
--	setUI	메인화면 UI를 보여준다.
--	dataLoad	MainWindow가 생성되면, LoginWindow에서 입력된 이름으로 저장된 데이터가 있는지 확인하고, 이를 불러와 self.data에 덮어쓴다.
--	dataSave	self.data를 현재정보로 업데이트하고, 파일에 저장한다. 그리고 그리던 그림은 그림파일로 저장한다.
--	recordLoad	저장된 기록 파일에서 파일을 불러와 반환한다.
--	recordSort	전달받은 기록에 현재 기록을 추가하고 등수로 정렬한뒤, 10개 넘어가면 자른다.
--	recordSaveNReturn	기록을 불러오고, 정렬한뒤, 기록파일에 저장한다.
--	moneyUpdate	입력받은 금액만큼 현재보유금액을 갱신하고, 입력받은 텍스트를 가계부에 추가한다. 그리고 변한 보유금액이 플레이어의 최대보유금액을 초과했는지 확인한다.
--	historyUpdate	입력받은 텍스트를 가계부에 추가한다.
--	peakofmoneyUpdate	현재보유금액이 최대보유금액을 초과했는지 판별하여 최대보유금액을 갱신한다.

	이름	역할,설명
--	timeUpdate	비트코인 그래프 애니메이션으로 부터 호출된다. self.time에 1씩더해 시계처럼 작동하며, 1시간이 지나면, 홀짝게임의 결과를 발표하고, 일주일이 지나면, 빗을 갠는 것을 반복한다.
--	payBack	게임상시간이 1주이링 지날 때마다 남은 빗이 있으면, 일정 금액을 상환한다. 만약 현재 보유금액이 상환액보다 적으면, 구매한 물감을 압류한다. 압류당할 물감도 없으면, 파산한다.
--	foreclosure	보유한 물감을 랜덤으로 삭제 하고, 콤보박스에서 이를 지우며, 해당 색을 상점에서 다시 활성화하고 어떤 물감을 압류당했는지 반환한다.
--	defeat	시간의 흐름을 중지하고, 메인 창을 닫고, 패배화면창을 띄운다.
--	bankrupt	현재보유금액이 없거나, 비트코인의 보유량이 0이거나, 홀짝게임의 총전금이 0이어서 더이상 돈을 벌 방법이 없다면 파산처리한다. 보유금액이 줄어들 때마다 호출돼 현재 상태를 판별한다.
--	showAchievement	업적 달성했는지 판별하고, 업적달성을 했으면, 현재 데이터를 저장하고 달성한 업적, 현재플레이어의 기록, 다른 플레이어의 해당 업적 기록 정보를 갖는 업적창을 띄운다.
--	achievement	업적 달성했는지 판별하는 함수. 달성한 업적의 종류를 반환한다. 만약 업적 달성이 아니면 0을 반환한다.
--	buttonClicked	데이터를 저장할 지 물어보고 yes면 데이터를 저장한다.

## Painter.py

### Painter

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 인스턴스를 정의한다.
--	mainlayout	메인 레이아웃
--	toollayout	도구의 레이아웃
--	canvaslayout	캔버스의 레이아웃
--	self.canvas	캔버스의 인스턴스
--	self.tool	도구의 인스턴스
method	__init__생성자	
--	setUI	그림판 UI를 보여준다.

## PainterTool.py

### PainterTool

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 인스턴스를 정의한다.
--	self.canvas	전달받은 canvas의 인스턴스를 정의한다.
--	self.save_eraser_size	canvas와 연계되는 도구 지우개 굵기 값을 저장한다.
--	self.save_brush_size	canvas와 연계되는 도구 브러쉬 굵기 값을 저장한다.
--	self.save_line_size	canvas와 연계되는 도구 선의 굵기 값을 저장한다.
--	self.save_brush_mode	canvas와 연계되는 브러쉬의 모드 값을 저장한다.
--	self.save_red	색 설정을 위한 RGBA의 Red 값

	이름	역할,설명
--	self.save_blue	색 설정을 위한 RGBA의 Blue 값
--	self.save_green	색 설정을 위한 RGBA의 Green 값
--	self.save_alpha	색 설정을 위한 RGBA의 Alpha 값
--	self.save_brush_color	색 설정을 위해 정한 변수와 QColor를 이용하여 색 정보를 저장한다.
--	self.tools	각 종 그리기 도구가 포함되는 리스트이다.
--	self.tools[0]	붓. 자유로운 곡선그리기가 가능하다.
--	self.tools[1]	직선 그리기
--	self.tools[2]	지우개
--	self.tools[3]	선굵기조절. 붓/직선그리기/지우개의 각각의 크기를 슬라이드로 조절할 수 있다.
--	self.tools[4]	선모드. 실선모드와 점선모드가 있다.
--	self.tools[5]	투명도 조절. 수채화를 구현해놓은 것. 슬라이드로 조절할 수 있다.
--	self.tools[6]	텍스트 입력. 캔버스에 원하는 문구를 입력할 수 있다. 글씨체와 크기까지 조정가능하다.
--	self.tools[7]	저장. 지금까지 그린 그림을 그림파일로 내보낸다.
--	self.tools[8]	전체 지우기. 캔버스에 그려져있는 모든 그림을 지운다.
--	self.cb	보유한 색상들의 콤박스. 원하는 색상을 선택해 선의 색을 바꿀 수 있다.
--	self.label	현재 사용중인 색상이 어떤 것인지 색깔로 보여준다.
method	__init__생성자	PainterTool를 실행한다.
--	setUI	도구의 UI를 보여준다.
--	comboBoxFunction	상점에서 구매한 색으로 변경한다.
--	save	그린 그림을 이미지 파일로 내보낸다.
--	clear	그림을 초기화 한다.
--	ChangedColor	받은 인자의 색으로 색정보를 변경한다.
--	ChangedSize	받은 인자의 굵기로 굵기 값을 변경한다.
--	ChangedRGBA	받은 인자의 값으로 색을 만들 수 있게 RGBA 값을 저장하고 변경한 색정보를 변경한다.
--	MakeColor	저장된 RGBA값을 이용하여 색을 만든다.
--	ChangedValue	색과 굵기와 모드를 모두 변경한다.
--	ChangedFont	폰트의 굵기와 크기를 변경한다.
--	buttonClicked	도구에서 눌린 버튼에 맞게 기능을 실행한다.
--	setThickness	굵기 정하는 창을 띄워주는 함수이다.
--	changingMode	선모드를 변경할 것인지 물어보는 함수이다.
--	ChangedText	텍스트 기능을 변경하기 위해 창을 띄워주는 함수이다.
--	ColorEffect	색효과를 주기 위해 사용자에게 input을 받는 함수이다.

# Canvas.py

## Canvas

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 인스턴스를 정의한다.
--	self.drawingPath	그림을 그리고 있는지 유무 파악
--	self.image	그림을 그릴 이미지 파일
--	self.color_r	색 RGBA의 값 중 Red
--	self.color_g	색 RGBA의 값 중 Green
--	self.color_b	색 RGBA의 값 중 Blue
--	self.color_a	색 RGBA의 값 중 Alpha
--	self.brush_color	브러쉬 색
--	self.brush_size	브러쉬 크기(굵기)
--	self.brush_mode	브러쉬 모드
--	string	텍스트 모드에서 사용할 글자
--	stringFont	글자의 폰트
--	stringFontSize	글자 크기
--	save_drawingType	그림을 그릴 때 브러쉬 모드
--	past_point	저장해 둘 절대 좌표 값(마우스 좌클릭 했을 때)
--	present_point	저장해둘 절대 좌표 값(마우스 좌클릭 뺐을 때)
method	__init__생성자	그림을 그린다.
--	setUI	window를 설정한다.
--	paintEvent	그림 이벤트 함수
--	mousePressEvent	마우스 버튼 누를 때 이벤트 함수
--	mouseMoveEvent	마우스 움직일 때 이벤트 함수
--	mouseReleaseEvent	마우스 버튼 뺐을 때 이벤트 함수

# Store.py

## Store

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 인스턴스를 정의한다.
--	mainlayout	메인레이아웃
--	tablayout	물감 구매버튼들이 나열되는 레이아웃
--	self.redButton	빨간색 물감 구매 버튼
--	self.yellowButton	노란색 물감 구매 버튼

	이름	역할,설명
--	self.blueButton	파란색 물감 구매 버튼
--	self.greenButton	초록색 물감 구매 버튼
--	self.orangeButton	주황색 물감 구매 버튼
--	self.purpleButton	자주색 물감 구매 버튼
--	self.brownButton	갈색 물감 구매 버튼
--	self.cyanButton	밝은 청록색 물감구매 버튼
--	self.skyblueButton	하늘색 물감 구매 버튼
--	self.colorButton_list	버튼의 색깔을 나열한 리스트
--	self.colorButton_dic	각 색깔의 문구와 해당 색깔의 버튼을 매치한 사전
--	self.RGBNumber_dic	각 색깔마다의 RGB값을 튜플 형태로 매치한 사전
--	button_list	버튼들을 나열한 리스트
--	self.color_price	물감 구매 가격
--	self.default_price	물감 구매 가격에 더해지는 가중치
--	self.count	물감 구매 횟수
method	__init__생성자	상점을 생성한다.
--	setUI	상점 UI를 보여준다.
--	buttonClicked	모든 물감버튼에 대한 콜백함수. 버튼을 누르면 물감을 구매할 지 묻는 팝업창을 띄운다. 물감을 구매하면 보유금액과 가계부를 갱신한다.
--	priceChange	물감 구매할 때마다 self.default_price를 연산하여 self.color_price에 더해 물감 가격을 올린다.

## GetMoney.py

### GetMoney

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 인스턴스를 비트코인거래소과 홀짝게임에 전달해 주기 위해 정의한것이다.
--	mainlayout	메인레이아웃
--	tablayout	비트코인거래소와 홀짝게임을 묶은 탭의 레이아웃
--	self.tab1	비트코인 거래소 탭. BitcoinMarket를 생성한다.
--	self.tab2	홀짝게임 탭. OddOrEven을 생성한다.
--	tabs	비트코인 거래소와 홀짝게임을 묶은 탭
method	__init__생성자	GetMoney를 생성한다.
--	setUI	GetMoney의 UI를 보여준다.

## Bitcoin.py

### Bitcoin

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 객체. 보유금액과 시간에 직접적으로 접촉하기 위해 정의됐다.
--	self.x	비트코인 그래프의 x축. 시간(분)
--	self.y	비트코인 그래프의 y축. 각 x에 대한 가격
--	self.economy	비트코인 그래프의 기울기 추세.
--	self.adjustment	비트코인의 과도한 성장을 제한하기 위해 기울기에 곱해지는 가중치.
--	self.gradient	비트코인 그래프의 기울기
--	self.subgradient	비트코인 그래프의 역동적인 움직임을 구현하기 위해 기울기에 더해지는 가중치
--	self.fig	비트코인 그래프가 그려지는 화면을 생성한다.
--	self.ax	비트코인의 축을 그린다.
--	self.canvas	그래프가 그려지는 화면
--	self.line	
--	self.ani	비트코인의 애니메이션 객체. 생성하면 애니메이션을 작동시킨다. interval을 조정하여 애니메이션의 속도를 조절할 수 있다. 1초 == 1000
--	self.itemname	한 비트코인 객체의 이름
--	self.price	비트코인의 현재 가격
--	self.holding	플레이어의 비트코인 보유량
--	self.investmentamount	해당 비트코인에 투자한 금액
--	self.presentvalue	비트코인의 현재 총 가치.보유량 * 현재가격
--	mainlayout	메인레이아웃
--	rightlayout	비트코인 정보가 입력되는 레이아웃
--	leftlayout	그래프가 그려지는 레이아웃
--	self.itemlabel	비트코인 이름의 라벨
--	itemnamefont	"의 폰트.
--	self.pricelabel	비트코인 가격의 라벨
--	self.holdinglabel	비트코인 보유량의 라벨
--	self.presentvaluelabel	비트코인 현재 총가치의 라벨
--	self.gnllabl	gross&loss.평가손익의 라벨
--	buyingbutton	매수버튼
--	sellingbutton	매도버튼
method	__init__생성자	비트코인 하나를 생성한다.
--	setUI	비트코인의UI를 보여준다.
--	buttonClicked	매수/매도버튼에 대한 콜백함수. 누르면 매수/매도 량을 입력할 수 있는 팝업창이 뜬다.
--	changeEconomy	그래프의 기울기 추세를 바꿀지 난수를 생성해 랜덤으로 정한다.



	이름	역할,설명
--	changeGradient	그래프의 기울기를 바꿀지 난수를 생성해 랜덤으로 정한다.
--	updateEconomy	그래프의 기울기 추세를 난수를 생성해 정한다. changeEconomy가 True일 때 호출된다.
--	updateGradient	그래프의 기울기를 재설정한다.changeGradient가 True일때 호출된다. 그래프의 기울기는 가중치 * (정규분포함수 * 기울기 추세)로 계산한다.
--	setAdjustment	가격 폭등을 막기위해 비트코인의 가격의 구간마다 곱해지는 가중치를 조정한다. 가격이 0이 되지 않게 조정도 해준다.
--	setSubgradient	현실처럼 역동적인 비트코인 그래프를 만들기 위해 매 분마다 기울기에 랜덤한 값을 더한다.
--	setYLim	그래프의 가격이 상승함에 따라 조정되는 기울기에 맞게 그래프의 Y축 범위를 조정한다.
--	updateLine	시간이 흐를 때마다 갱신될 그래프의 값을 전달해주는 함수. 그래프의 내용을 갱신과 동시에, 기울기 조정 여부를 정한다. 게임 시간은 여기 있는 timeUpdate에 의해 이루어지며, 그래프는 게임상 시간 60분이 차면 처음부터 그려진다.

#### BitcoinMarket

	이름	역할,설명
attributes	self.status	전달받은 MainWindow의 인스턴스를 각 Bitocin에 전달해주기위해 정의됐다.
--	mainlayout	메인레이아웃
--	scrollarea	비트코인들의 그룹을 스크롤 가능한 공간에 넣은것.
--	bitcoins	비트코인들을 그룹화 한것.
--	self.vbox	bitcoins의 레이아웃.
--	self.bitcoins	생성된 5개의 비트코인이 담겨있는 리스트. 저장할때 데이터에 이 리스트를 저장한다.
method	__init__생성자	비트코인 거래소를 생성한다.
--	setUI	비트코인 거래소의 UI를 보여준다. 이때 현재 플레이어의 이름으로 저장된 비트코인 정보가 있으면 이를 불러오고, 없으면 새로 생성한다.
--	setBitcoinName	소문자 알파벳과 대문자 알파벳에 해당하는 아스키코드 값을 난수로 생성해 랜덤한 이름을 만든다.
--	bitcoinLoad	저장된 비트코인 정보를 받아온다.
--	bitcoinGenerate	비트코인을 생성해, 비트코인 거래소에 추가하고, 전달받은 MainWindow의 인스턴스에 이 코인의 정보를 추가한다.

## OddOrEven.py

#### OddOrEven

	이름	역할,설명
attributes	self.status	전달받은 Mainwindow의 인스턴스를 정의한다.
--	self.charge	홀/짝 게임에 베팅하는 금액. 충전금. self.status에 저장돼있는 데이터로 충전금을 초기화한다.
--	self.history	플레이어가 다음 결과를 예측할 수 있게, 전의 결과를 기록해둔 것. self.status에 저장돼있는 데이터로 기록을 초기화한다.
--	mainlayout	홀짝게임의 메인레이아웃
--	historylayout	기록에 대한 레이아웃

	이름	역할,설명
--	buttonlayout	홀/짝버튼에 대한 레이아웃
--	statuslayout	상태메세지에 대한 레이아웃
--	charginglayout	충전금 관련 위젯들에 대한 레이아웃
--	self.historylabel	결과의 기록. 새 결과가 나올때마다 갱신된다. 1일이 지나면 초기화된다.
--	historyfont	기록에 대한 폰트
--	self.oddbutton	홀 버튼. 눌린상태가 돼야 선택한 것이다.
--	self.evenbutton	짝 버튼. "
--	buttonfont	홀/짝 버튼의 폰트
--	status	결과에 따른 상태메세지 문구
--	self.statuslabel	상태메세지 라벨
--	statusfont	"의 폰트
--	self.chargelabel	충전금 라벨
--	chargefont	"의 폰트
--	chargetbutton	금액 충전버튼.
--	withdrawbutton	출금버튼
--	number	홀/짝게임의 결과. 1은 홀, 2는 짝이다.
method	__init__생성자	홀짝게임을 생성한다.
--	setUI	홀짝게임의 UI를 보여준다.
--	statusUpdate	현 홀짝게임의 상태메시지를 갱신한다.
--	chargeUpdate	충전금을 충전/출금하거나 돈을 따거나/잃은 만큼 변화를 반영하고, 충전금 라벨을 갱신한다.
--	getNumber	게임상 시간 1시간마다 호출되며, 1이나 2의 수를 랜덤으로 생성하여 이를 기록 업데이트 함수와 결과확인 함수에 전달한다. 1이 홀이고 2는 짝이다.
--	checkResult	홀/짝 버튼이 눌린 상태와 전달받은 number를 비교해 충전금에 그 결과를 반영하고 상태메세지를 생신한다.
--	historyUpdate	기록을 갱신한다. 게임상 시간 하루가 지나면 초기화된다.
--	buttonClicked	충전/출금, 홀/짝 버튼에 대한 콜백함수. 충전/출금 버튼에서는 사용자의 보유 금액을 변화시키고, 홀/짝 버튼은 한번에 두 버튼을 누를 수 없게 설정한다.

Slide.py

Slide\_Thickness

	이름	역할,설명
attributes	self.tool	전달받은 Tool의 인스턴스를 정의한다.
--	self.currentsize	현재 선의 굵기. tool에서 현재 눌러있는 버튼에 따라 다르게 초기화된다.
--	self.title	슬라이드 창의 제목. tool에서 현재 눌러있는 버튼에 따라 달라진다.
--	self.size	슬라이드 창에 보이는, 슬라이드의 수치이다. ok 버튼을 누르면 선굵기가 해당 수치로 변한다.

	이름	역할,설명
--	self.sld	사용자가 직접 잡고 끌어 수치를 변경할 수 있는 슬라이드.
--	self.okbutton	ok버튼
--	self.cancelbutton	cancel버튼
--	font	슬라이드에 표시되는 수치의 폰트
--	mainlayout	슬라이드의 메인 레이아웃
method	__init__생성자	슬라이드를 생성한다.
--	setUI	슬라이드의 UI를 보여준다.
--	setText	슬라이드의 변화에 따라 슬라이드에 표시되는 값을 변경한다.
--	buttonClicked	ok/cancel에 대한 콜백 함수.

**Slide\_ChangedText**

self.currentsize가 텍스트의 크기를 의미하는 것 말고는 Slide\_Thickness와 동일하다.

**Slide\_ColorEffect**

self.currentsize가 선의 투명도를 의미하는 것 말고는 Slide\_Thickness와 동일하다.

**Button.py**

---

**Button**

수업에 제공된 실습파일에서 크기 조절하는 부분을 삭제한 것이다. 이미 수업에서 다뤘던 부분이니 자세한 설명은 생략한다.