

Piano di sviluppo del gioco

- Cicli bisettimanali con numero variabile di storie
- Sviluppo task-oriented in Test Driven Development
- Extreme Programming adattato al contesto

Punti cardine da tenere a mente durante i lavori:

Durante lo sviluppo del progetto, occorre tenere a mente i seguenti punti cardine:

- **Portabilità:** la possibilità di eseguire il codice su qualsiasi sistema;
- **Compatibilità:** evitare conflitti, e minimizzare i requisiti minimi di sistema;
- **Accessibilità:** la meccanica deve essere semplice, e la curva di apprendimento ben regolata;
- **Competitività:** occorre stimolare lo spirito di competizione tra i due giocatori, per incrementare il loro interesse nel gioco.

Questo comporta l'adozione delle seguenti regole auto-imposte:

- ✓ Mai fare a gara a chi finisce il task per primo, meglio procedere con calma ed evitare di perdere molto più tempo in seguito;
- ✓ Evitare di complicarsi la vita, esiste di certo una soluzione più semplice di quella che abbiamo pensato di implementare;
- ✓ MAI aggiungere elementi non richiesti esplicitamente dai task: se mai serviranno, se ne parlerà nelle prossime storie;
- ✓ Comunicare il più possibile, se qualcosa non è chiaro bisogna discuterne tutti i dettagli fino ad eliminare ogni dubbio, anche il più insignificante.

Piano dettagliato di sviluppo:

• Ciclo 1 (19/09/05 – 02/10/05) **COMPLETO**

Storia 1: Il gioco deve disegnare un diamante su uno sfondo nero in finestra o a pieno schermo e riprodurre un suono.

Task list:

- 1.1.1: Aprire una finestra di dimensione fissa (800x600 pixel di client area) - **cdimauro**
- 1.1.2: Inizializzare OpenGL per il disegno 2D e colorare la finestra di nero via OpenGL - **TigerShark**
- 1.1.3: Inizializzare OpenAL - **cisc**
- 1.1.4: Caricare l'immagine del diamante in una texture - **Vifani**
- 1.1.5: Disegnare la texture del diamante applicata a due poligoni che formano un quadrato in coordinate schermo nel centro della finestra - **Vifani**
- 1.1.6: Caricare un suono e riprodurlo - **cisc**
- 1.1.7: Disegnare una texture 64x64 con un diamante - **Antares88**

Storia 2: Un sistema di build automatico che gira sul server, scarica l'ultima versione del codice disponibile dal repository e riporta via mail in caso di errori di compilazione o testing.

Task list:

- 1.2.1: Permettere al diamante di essere posizionato in un punto qualsiasi dello schermo - **Vifani**
- 1.2.2: Spostare il diamante nelle quattro direzioni di X pixel - **Vifani**
- 1.2.3: Intercettare la pressione delle frecce direzionali e spostare il diamante - **cdimauro**
- 1.2.4: Rilevare la "collisione" con i bordi dello finestra e impedire lo spostamento - **DanieleC88**
- 1.2.5: Riprodurre il suono quando il diamante collide con un bordo - **BlueDragon**
- 1.2.6: Permettere al diamante di pulsare durante lo spostamento - **VICIUS**

• **Ciclo 2 (03/10/05 – 16/10/05) COMPLETO**

Storia 1: Il diamante deve essere ridotto a 32x32, e devono essere ignorati i movimenti verso l'alto (diagonali alto-sx ed alto-dx incluse). Inoltre, l'area di gioco deve essere ristretta ad un rettangolo di 256x448 pixel, nelle collisioni con i bordi sinistro e destro non devono essere riprodotti suoni, ed occorre fare in modo che le collisioni con gli stessi arrestino il movimento orizzontale del diamante (per non farlo uscire dall'area), ma non quello verticale di caduta (continuo, presente anche senza la pressione del tasto giù).

Task list:

- 2.1.1: Ridurre GEM_SIZE fino a 32px - **DanieleC88**
- 2.1.2: Impedire spostamenti verso l'alto degli sprite tramite tastiera (comprese le diagonali) - **TigerShark**
- 2.1.3: Limitare gli spostamenti degli sprite ad un'area dello schermo. Usare come esempio 256x448 pixel - **71104**
- 2.1.4: Riprodurre il suono solo quando il diamante collide con il bordo inferiore - **TigerShark**
- 2.1.5: Aggiungere un movimento continuo verso il basso di qualche pixel al diamante che si deve fermare solo in caso di collisione col bordo inferiore - **cisc**

Storia 2: Introduzione di una griglia (14 righe per 8 colonne) di caselle da 32x32 pixel ciascuna, che andrà a coincidere con l'area di gioco. I movimenti del diamante alla pressione dei tasti direzionali devono avvenire lungo la griglia (nelle direzioni permesse e senza dimenticare i bordi). La velocità di spostamento del diamante all'interno della griglia deve essere pari a circa l'80% di quella attuale (lo spostamento laterale da una casella all'altra deve essere pressochè istantaneo, quello verticale più lento e fluido).

Task list:

- 2.2.1: Creare una texture 32x32 da usare come sfondo per le celle dell'area di gioco - **Jocchan**
- 2.2.2: Disegnare le celle all'interno dell'area di gioco rappresentata da extents (8 colonne x 12 righe = 112 celle) - **71104**
- 2.2.3: Creare una nuova area di gioco per il secondo giocatore e disegnare le celle anche lì - **71104**
- 2.2.4: Modificare lo spostamento laterale delle gemme. Alla pressione dei tasti destra e sinistra lo spostamento nella colonna accanto deve essere istantaneo - **Vifani**
- 2.2.5: Creare una texture 800x600 contenendo il fondale da usare per la finestra - **Jocchan**
- 2.2.6: Disegnare il fondale del task 2.2.5 sotto alle aree di gioco - **71104**
- 2.2.7: Creare un logo semitrasparente da mostrare sopra al fondale - **Jocchan**
- 2.2.8: Disegnare il logo del task 2.2.7 in basso tra le due aree di gioco - **cdimauro**
- 2.2.9: Spezzare la classe playarea in due o più classi - **71104**
- 2.2.10: Auditing del codice e delle classi Test. Aggiungere test per il codice non testato. eliminare test non più necessari e modificare quelli vecchi - **Tutti**

• **Ciclo 3 (17/10/05 – 30/10/05) COMPLETO**

Storia 1: Il sistema di coordinate utilizzato dal gioco è definito con l'origine degli assi in alto a sinistra dello schermo. Occorre tradurre queste coordinate, in modo da renderle compatibili con il formato adottato da OpenGL.

L'hot spot deve essere situato in alto a sinistra. Uno sprite è definito da una porzione rettangolare di texel di dimensioni qualsiasi, posta all'interno di una texture di dimensione in potenza di due.

Task list:

- 3.1.1:** Transizione al nuovo sistema di coordinate in tutto il progetto e i test - **TigerShark**
- 3.1.2:** Aggiungere a Texture e Sprite la capacità di gestire immagini con dimensioni diverse da una potenza di due – **cisc + fek**
- 3.1.3:** Transizione dell'hot spot dal centro all'angolo in alto a sinistra - **71104**

Storia 2: Introduzione di una griglia (14 righe per 8 colonne) di caselle da 32x32 pixel ciascuna, che andrà a coincidere con l'area di gioco. I movimenti del diamante alla pressione dei tasti direzionali devono avvenire lungo la griglia (nelle direzioni permesse e senza dimenticare i bordi). Lo spostamento laterale da una casella all'altra deve essere pressoché istantaneo, quello verticale invece più lento e fluido. Lo sfondo dell'area di gioco è una texture unica da 256x448 pixel.

Task list:

- 3.2.1:** Realizzare la Texture 256x448 trasparente da usare come sfondo per l'area di gioco - **Jocchan**
- 3.2.2:** Creare una classe PlayAreaBackground che disegna la griglia usando la texture del task 3.2.1 - **Gammaglobulino**
- 3.2.3:** Creare una matrice di sprite MxN allineate al background e disegnare i diamanti contenuti – **71104 + fek**
- 3.2.4:** Inserire un diamante nella colonna centrale. Dev'essere possibile inserire un diamante solo dall'ultima riga di ogni colonna. Aggiungere a Texture la possibilità di abilitare l'uso del canale alpha - **Vifani**
- 3.2.5:** Alla pressione dei tasti destra/sinistra il diamante deve spostarsi nella colonna adiacente istantaneamente - **fpitt**
- 3.2.6:** Spostare la gestione della gravità e le collisioni dei diamanti nella classe che gestisce la griglia - **fpitt + cisc**

• **Ciclo 4** (31/10/05 - 13/11/05) **COMPLETO**

Storia 1: Introduzione di una temporizzazione fissata per il movimento della gemma nell'area di gioco.

Occorre fare in modo che, alla rapida pressione delle frecce direzionali sinistra e destra, la gemma si sposti di una e una sola casella. Tenendo premuti i tasti in questione, la gemma continuerà a spostarsi nella direzione corrispondente di una casella per ogni decimo di secondo, a meno di collisioni con i bordi laterali. Come in precedenza, alla pressione contemporanea dei tasti sinistra e destra, la gemma non dovrà spostarsi in nessuna delle due direzioni, e dovrà comunque continuare la sua caduta verticale (a meno di collisione con il bordo inferiore). In secondo luogo, come sfondo della schermata, l'engine del gioco disegnerà una immagine fissa in jpg, di dimensione 800x600 e profondità di colore 24 bit.

Task list:

- 4.1.1:** Disegnare l'immagine Jpg dello sfondo alle coordinate 0x0 di dimensione 800x600 pixel dietro a tutti gli altri elementi presenti nella finestra - **cover**
- 4.1.2:** Creare un timer molto preciso che sia in grado di misurare il trascorrere del tempo con una precisione pari ad almeno 5 millesimi di secondo - **BlueDragon**
- 4.1.3:** Sincronizzare il disegno di ogni frame in ad 1/50 di secondo aggiungendo un loop che attende il termine dei 20 millisecondi previsti per il disegno di un intero frame prima di lasciare il controllo - **71104**
- 4.1.4:** Input deve ignorare la pressione di tasti uguali e consecutivi se l'intervallo tra loro è inferiore ad 1/50 di secondo segnalando solo la pressione del primo - **TigerShark**
- 4.1.5:** Refactoring. Aggiungere a drawable la capacità di specificare il layer su cui disegnare i vari oggetti - **VICIUS**
- 4.1.6:** Refactoring della classe Game. spostare tutta la creazione nel costruttore e creare un gameLoop come si deve - **cionci**

Storia 2: Introduzione del rilevamento del tasto giù, alla cui pressione la gemma deve accelerare il proprio movimento verso il basso causato dalla gravità.

Quando il tasto viene rilasciato, la velocità della gemma tornerà ad essere quella usuale di caduta. Il tutto va implementato senza dimenticare la possibilità di premere contemporaneamente a giù anche i tasti sinistra e destra, considerando tutte le possibili combinazioni (sinistra+destra+giù, sinistra+giù, destra+giù). E' fondamentale che il gioco risponda prontamente ai comandi, distinguendo più pressioni immediatamente successive (anche in caso di situazioni concitate), ed evitando al contempo che ciò comporti spostamenti indesiderati della gemma, come spostamenti di più caselle in caso di una pressione leggermente più lunga del solito.

Task list:

- 4.2.1:** Quando il tasto "Giù" è mantenuto premuto il valore della gravità usato da Grid deve essere moltiplicato per un fattore definito nel

file di configurazione del gioco. Quando il tasto viene rilasciato deve ritornare ad essere quello originale - **cionci**

4.2.2: Registrare in una coda all'interno di Input gli eventi "pressione" e "rilascio" dei vari tasti. Ad ogni evento deve essere associato il timestamp in cui l'evento è stato scatenato - **71104 + cdimauro**

4.2.3: Modificare reactToInput di Grid per usare la coda creata nel task 4.2.2 in Input. Il comportamento attuale deve essere mantenuto inalterato - **71104**

4.2.4: Una volta reagito ad un comando, reactToInput deve ricordare uno e un solo ulteriore comando richiesto durante il tempo di 'inattività' e dovrà eseguirlo allo scadere del tempo stesso - **RINVIATO**

• **Ciclo 5 (14/11/05 - 27/11/05) COMPLETO**

Storia 1: Storia di servizio, dedicata al perfezionamento della risposta agli input del giocatore. La risposta ai comandi deve essere istantanea e precisa, senza che alcun comando venga mai saltato, anche nelle situazioni più concitate. Nuovamente, occorrerà assicurarsi che la gemma non si sposti in maniera indesiderata dopo una pressione leggermente più lunga del solito, ed occorrerà correggere in maniera definitiva il bug di posizionamento della stessa alla pressione contemporanea dei tasti sx e dx.

Task list:

5.1.1: Implementare l'algoritmo di gestione degli Input di BlueDragon presente nel branch trunk/ - **71104 + BlueDragon**

Storia 2: Creazione di una nuova gemma quando la precedente tocca il fondo o una gemma sottostante (emissione del bip in entrambi i casi), con rilevamento delle collisioni tra le gemme presenti nell'area di gioco (non sarà possibile spostare la gemma in una casella già occupata da un'altra). Introduzione di un contatore per il punteggio del giocatore, inizializzato a zero, capace di assumere valori validi di 8 cifre e - per ora - mostrato su schermo come semplice testo (possibilmente nelle coordinate della casella Score sinistra).

Task list:

5.2.1: Quando la gemma collide con il bordo inferiore dell'area di gioco ne deve essere creata una nuova nella parte superiore che diventa quella gestita tramite input da tastiera - **cionci**

5.2.2: Modificare Grid in modo che lo spostamento verticale della gemma sia fermato se la casella della riga sottostante è già occupata. (solo pair programming) – **cover + TigerShark**

5.2.3: Impedire lo spostamento laterale della gemma se la casella nella colonna di destinazione è occupata - **DanieleC88**

5.2.4: Aggiungere la possibilità di disegnare un numero intero di massimo 8 cifre usando la classe Sprite e come base una texture - **71104**

5.2.5: Visualizzare nella casella dello score presente alle coordinate 219x421 il numero di diamanti presenti all'interno della griglia. Ogni cifra deve essere distanziata dall'altra 11 pixel - **Vifani**

• **Ciclo 6** (28/11/05 - 11/12/05) **COMPLETO**

Storia 1: Ogni gemma dovrà avere un colore (tra 5 possibili: bianco, rosso, blu, verde, giallo), a cui corrisponderanno una differente png (rispettivamente diamond, ruby, sapphire, emerald, topaz) ed un differente valore in punteggio (per ora rispettivamente 1000, 500, 600, 400, 800), e dovrà riconoscere se nelle vicinanze (diagonali escluse, quindi nelle quattro direzioni principali) ha delle gemme di uguale colore.

Il colore delle prime sei gemme viene scelto casualmente all'inizio del gioco, e viene inserito in una coda. Ad ogni gemma creata, viene estratto il primo valore e viene creato un nuovo valore casuale che si appende in coda.

Task list:

- 6.1.1:** Aggiungere la possibilità di scegliere quale tipo di gemma deve essere creata da GameGemFactory quando viene chiamata la create. I vari tipi di gemme per ora sono: diamond, ruby, sapphire, emerald, topaz - **71104**
- 6.1.2:** La casella dei punti non deve più riportare il numero di gemme all'interno della griglia. Il nuovo valore è la somma dei punti che ogni singola gemma rappresenta. Alle gemme elencate nel task 6.1.1 corrisponderanno i valori: 100, 50, 60, 40, 80 - **TigerShark**
- 6.1.3:** Aggiungere un metodo per modificare la "luminosità" di uno sprite - **Ufo13**
- 6.1.4:** Grid deve aumentare la luminosità delle gemme che hanno almeno un lato in comune con un'altra gemma dello stesso tipo - **71104 + DanieleC88**
- 6.1.5:** Aggiungere a GameGemFactory una coda che deve contenere le prossime 6 Gemme da rilasciare - **cionci**
- 6.1.6:** Quando viene richiesta una nuova gemma GameFemFactory deve restituire quella in testa alla coda ed eliminarla dalla lista. Fatto questo deve essere creato un nuovo valore da appendere in coda - **Ufo13**

Storia 2: Ogni gemma creata dovrà avere un'animazione a ciclo continuo che simuli un riflesso, e che si ripeta ogni 3.5 secondi, valore comunque modificabile in GameConfig. Le animazioni di tutte le gemme su schermo dovranno essere sincronizzate tra loro, ossia iniziare tutte nello stesso momento.

Nella finestra "next" verrà visualizzata una coppia di gemme dei due colori immediatamente successivi nella coda a quella attualmente in caduta. Di queste due gemme, quella in basso sarà la prossima a cadere, e la coppia verrà aggiornata ogni volta che una nuova gemma viene creata.

Task list:

- 6.2.1:** Mostrare nel riquadro "next" dell'area di gioco le prossime due gemme che saranno rilasciate. La prima gemma ad essere rilasciata dovrà essere quella più in basso - **71104**
- 6.2.2:** Aggiungere a sprite la capacità di caricare Texture che contengono i vari frame delle animazioni - **TheBol**
- 6.2.3:** Implementare le animazioni in sprite cambiando frame ogni decimo di secondo. Al termine dell'animazione si deve mostrare il

frame di base per 3500 millisecondi prima di ripartire - **cionci + fek**

6.2.4: Sincronizzare le animazioni delle gemme all'interno della griglia tra di loro. Se una gemma viene creata quando le altre sono a meta animazione deve quindi mostrarsi anche lei a meta animazione - **cionci + fek**

• **Ciclo di Natale (12/12/05 - 06/01/06) COMPLETO**

Storia 1: Storia di servizio. Quando la colonna centrale in cui le gemme entrano nell'area di gioco è piena, il gioco deve smettere di crearle, e deve far comparire al centro dello schermo una png con la scritta Game Over. Occorre anche fare in modo che, quando viene premuto un tasto direzionale (sinistra o destra) mentre l'altro opposto è già premuto, venga eliminata l'incertezza sulla colonna di posizionamento della gemma, che deve restare fissa nella colonna in cui si trovava un istante prima della pressione del tasto contrastante.

Task list:

Xmas.1.1: Interrompere l'inserimento di nuove gemme quando la colonna centrale si riempie dopo di che si deve mostrare la scritta "Game Over" all'interno della griglia - **71104**

Xmas.1.2: Aggiungere un ritardo di 300 millisecondi tra l'istante in cui la gemma si ferma e la creazione di una nuova gemma - **cisc**

Xmas.1.3: Ignorare la pressione del tasto direzionale finché quello con direzione opposta rimane premuto - **cionci**

Xmas.1.4: Portare a 50 millisecondi il repeat delay dopo la prima volta che è stato superato. E riportarlo al suo valore originale quando il tasto viene rilasciato - **Ufo13**

Storia 2: Le gemme devono scendere a coppie di due, sebbene la loro caduta sia scollegata. Quella inizialmente in basso è la gemma primaria della coppia, quella inizialmente in alto è la secondaria. Premendo il tasto Z, la secondaria deve cambiare posizione ruotando (ma non su sé stessa, le png devono avere lo stesso aspetto) in senso antiorario intorno alla primaria. Premendo il tasto X, invece, la secondaria deve ruotare in senso orario. Se la casella di destinazione della gemma secondaria è già occupata, e la rotazione è impossibile, la gemma verrà automaticamente fatta ruotare ulteriormente nello stesso senso di rotazione, fino a trovare una casella libera in cui posizionarla. Gli effetti delle rotazioni sono descritti nello schema:

- secondaria sopra la primaria -> la secondaria andrà rispettivamente a sinistra o a destra della primaria
- secondaria a sx della primaria -> andrà rispettivamente sotto o sopra
- secondaria sotto la primaria -> andrà rispettivamente a destra o a sinistra
- secondaria a dx della primaria -> andrà rispettivamente sopra o sotto

Mediante il tasto C, la gemma secondaria dovrà ruotare di due posizioni. Quando ognuna delle due gemme collide con qualche oggetto sottostante, l'altra - se non troverà nulla al di sotto - proseguirà indipendentemente la propria caduta. Il box Next continuerà a mostrare, senza incertezze, la coppia di gemme immediatamente successiva a quella attualmente in caduta (la nuova primaria in basso, la nuova secondaria in alto).

- **Ciclo 8 (09/01/06 - 22/01/06) COMPLETO**

Storia 1: Quando una gemma è a contatto con altre dello stesso colore, e queste formano un quadrilatero nella griglia (quadrato o rettangolo di dimensioni qualsiasi), formeranno un agglomerato. Le gemme nell'agglomerato useranno delle png differenti a seconda che si trovino nei bordi dello stesso (4 angoli e 4 lati) o al suo interno. L'agglomerato è considerabile come un oggetto complesso, il cui valore di punteggio è pari alla somma dei singoli punteggi delle gemme componenti, più un bonus direttamente proporzionale al numero di gemme coinvolte (il valore del bonus è unico per tutti gli agglomerati, ed è variabile da GameConfig). Le gemme presenti in un agglomerato non possono più tornare ad essere singole.

- **Ciclo 9 (23/01/06 - 05/02/06) COMPLETO**

Storia 1: Introduzione del tipo di pezzo "baule" (con numero di colori equivalente a quello delle gemme e punteggio nullo). Quando il baule collide con il fondo o si ferma su una gemma (o agglomerato), tutte le gemme (e/o gli agglomerati) di colore uguale al baule, adiacenti ad esso ed adiacenti tra loro in una delle quattro direzioni principali, spariscono, incrementando il punteggio di un valore pari a quello della somma dei singoli punteggi, e lo spazio lasciato libero viene coperto dalle gemme sovrastanti (gli agglomerati, essendo il legame irreversibile, scenderanno SOLO se non ci saranno gemme al di sotto di tutta la loro larghezza).

Per chiarirci le idee, definiamo questo meccanismo "cancellazione", ed ogni volta che parleremo di cancellazione ci riferiremo a tutto il processo nella sua interezza.

La collisione gemma-baule deve essere rilevata anche in senso inverso: se una gemma, quando si ferma, collide con un baule in una delle quattro direzioni principali, si innesca ugualmente il meccanismo di cancellazione sopra descritto. Questo può innescare delle reazioni a catena (Crush).

Fin quando ci sono cancellazioni in corso, non devono cadere altre gemme.

- **Ciclo 10** (06/02/06 – 19/02/06) **COMPLETO**

Storia 1: Introduzione di una griglia secondaria, in grado di contenere gli stessi oggetti della principale già in uso, e controllata da un secondo giocatore mediante il tastierino numerico (8, 4, 6 e 2 per le 4 direzioni; 7, 9 e 5 per Z, X e C).

La coda di gemme da creare deve essere inizialmente uguale per i due giocatori. Ogni volta che una coppia di gemme collide, ed una nuova coppia viene aggiunta in coda, la stessa coppia deve essere aggiunta anche nella coda dell'altro giocatore, in maniera da conservare l'uguaglianza delle due code in ogni situazione possibile.

Il secondo box Score deve inoltre riportare il punteggio dell'avversario, similmente a quello già presente, ed allo stesso modo il secondo box Next deve riportare la prossima coppia in caduta nello schermo dell'avversario, in maniera simile a quanto già accade per il giocatore.

Storia 2: Definizione del gameturn, un intero incrementato di 1 alla fine di ogni loop in Game (ovvero ogni 20 millisecondi). Ogni valore in millisecondi presente nel codice del gioco dovrà essere convertito in gameturn.

Introduzione di un sistema di logging degli eventi, che salvi su file ogni pressione e rilascio dei vari tasti, insieme al relativo gameturn.

- **Ciclo 11** (20/02/06 - 05/03/06) **COMPLETO**

Storia 1: Introduzione di un contatore per le Crush, che venga portato a 1 dopo una prima cancellazione, e continui ad essere incrementato di 1 ogni volta che, quando le gemme sovrastanti cadono per riempire i vuoti, si verifichino altre cancellazioni. Quando il suo valore è maggiore di 1, sulla sinistra dello schermo vengono mostrate delle png indicanti la lunghezza attuale della Crush.

Al termine della Crush, il valore in punteggio delle gemme cancellate durante la Crush stessa, prima di essere sommato al punteggio totale, deve essere moltiplicato per il valore massimo raggiunto dal contatore. Dopo questa operazione, il contatore verrà riportato a zero, e le gemme riprenderanno a cadere.

Storia 2: Introduzione di un contatore contenente il numero totale di gemme (e non bauli) eliminate in una Crush di lunghezza minima pari a 2, ed il cui valore (aggiornato al termine della Crush in questione), se diverso da zero, venga mostrato in un box "Warning" sotto l'area di gioco dell'avversario. Allo stesso modo, il valore dell'equivalente contatore dell'avversario verrà riportato sotto l'area di gioco del giocatore.

Se non ci dovessero essere cancellazioni, o la Crush innescata dovesse avere lunghezza minore di 2, questo contatore verrà riportato a zero, ed il box "Warning" corrispondente verrà reso invisibile.

- **Ciclo 12** (06/03/06 - 19/03/06) **COMPLETO**

Storia 1: Bugfix e refactoring.

Storia 2: Introduzione della *Flashing Gem*, pezzo speciale SENZA colore nè punteggio. Quando questa collide con un'altra gemma (o baule), tutte le gemme e tutti i bauli dello stesso colore di quest'ultima devono essere cancellati dall'area di gioco, senza dare alcun punteggio, nè avere alcun peso nelle eventuali Crush scatenate o già attive.

Il colore delle gemme da cancellare verrà deciso in base al colore degli elementi (in ordine di priorità) al di sotto, a sinistra, a destra o al di sopra della *Flashing Gem*.

Questo pezzo deve quindi avere una percentuale di occorrenza molto bassa, quantificabile intorno al valore di default del 2% e modificabile via GameConfig, insieme alle percentuali di occorrenza delle singole gemme.

- **Ciclo 13** (20/03/06 - 02/04/06) **COMPLETO**

Storia 1: Introduzione delle pietre, un nuovo tipo di pezzo definito da un colore (come per le gemme), da un punteggio nullo e da un numero, scelto con un valore tra 5 e 1, a seconda della riga (dal basso verso l'alto) in cui le stesse verranno depositate, seguendo lo schema in basso.

Tale numero sarà mostrato nella png usata dalla pietra in questione, ed il suo valore diminuirà di 1 ogni volta che una coppia viene droppata dal giocatore. Quando questo arriverà a zero, la pietra in questione si trasformerà in una gemma del colore corrispondente.

Ogni volta che il giocatore dropa una coppia di gemme, e prima che la coppia successiva inizi a cadere, un numero di pietre pari al valore del contatore situato sotto la propria area di gioco, e di colore scelto casualmente, verrà fatto cadere (a velocità accelerata) nella sua schermata, da sinistra verso destra, ed eventualmente su più strati in verticale. Al termine di questa operazione, il contatore verrà riportato a zero, e le gemme torneranno a cadere.

Schema per le pietre		
Righe	->	Valore
13, 12	->	5
11, 10	->	4
9, 8, 7	->	3
6, 5, 4	->	2
3, 2, 1, 0	->	1

- **Ciclo 14** (03/04/06 - 16/04/06) **COMPLETO**

Storia 1: Ogni volta che il giocatore dropa una coppia di gemme, e prima che la coppia successiva inizi a cadere, un numero di pietre pari al valore del contatore situato sotto la propria area di gioco, e di colore scelto

seguendo il pattern definito in basso, verrà fatto cadere (a velocità accelerata) nella sua schermata, da sinistra verso destra, ed eventualmente su più strati in verticale. Al termine di questa operazione, il contatore verrà riportato a zero, e le gemme torneranno a cadere.

Prima della trasformazione in gemma, una pietra potrà essere cancellata solo ed esclusivamente in seguito alla cancellazione di una gemma adiacente. Il suo punteggio sarà nullo, ma conterà ugualmente per il numero di pietre da inviare all'avversario.

Questo numero verrà sottratto da quello delle pietre in arrivo: se il risultato sarà minore di 0, il giocatore ne riceverà un numero minore, pari al risultato appena ottenuto, e per valori maggiori di zero sarà l'avversario a riceverle. Questo processo è definito come "contrattacco" e verrà evidenziato da una png "Counter", mostrata al posto della Warning Box di chi lo effettua.

Pattern: Il pattern è definito da una matrice 8x1, replicabile all'infinito in verticale, e contenente una serie di numeri compresi tra 1 e 5. All'inizio di ogni partita, questi numeri vengono abbinati casualmente a dei colori, e questo definirà il colore delle Stones da inserire nell'area di gioco. Il pattern da seguire attualmente è: 1 2 2 3 3 4 4 5.

FIRST PLAYABLE VERSION

- **Ciclo 15** (01/05/06 - 14/05/06)

Storia 1: Introduzione del menu principale, in cui selezionare la modalità di gioco. Lo sfondo del menu sarà una jpg fissa, e le voci evidenziabili saranno cinque (Story Mode, Versus, Advanced, Custom, Options), grazie ad una png suddivisa in 5 caselle (in verticale), di dimensione 512x60 px ciascuna, ed ognuna contenente la grafica per la corrispondente voce evidenziata. Con le frecce direzionali su e giù, verrà cambiata la casella della png da mostrare, ovviamente nelle coordinate corrispondenti allo sfondo, insieme ad una variabile capace di assumere valori tra 1 e 5 (ognuno corrisponderà ad una diversa modalità). Premendo giù sull'ultima voce si selezionerà la prima, premendo su sulla prima si selezionerà l'ultima. Confermando con invio si accederà al gioco.

Storia 2: Bugfix. Introduzione di un sistema di playback dei log.

- **Ciclo 16** (15/05/06 - 28/05/06)

Advanced Mode: introduzione della dinamite.

- **Ciclo 17** (29/05/06 - 11/06/06)

Advanced Mode: introduzione degli orologi.

- **Ciclo 18** (12/06/06 - 25/06/06)

Connessione ad una griglia remota.

Principali TODOs:

- online via immissione IP
- sistema di scripting
- story mode
- sistema ranghi
- musica dinamica
- desperation moves (Advanced)

Obiettivi:

First Playable (pre-Alpha): 16/04/2006 – ciclo 14

Alpha: 03/09/2006 – ciclo 23

Beta: 01/10/2006 – ciclo 25

Release: 12/11/2006 – ciclo 28